

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
Факультет ринкових, інформаційних та інноваційних технологій
Кафедра інформаційно-комп'ютерних технологій
та фундаментальних дисциплін

ДИПЛОМНА БАКАЛАВРСЬКА РОБОТА

на тему:

**«ПРОГРАМНА РЕАЛІЗАЦІЯ АДАПТИВНОГО НАВЧАННЯ
ПО НАПРЯМУ FRONT-END»**

Виконав: студент групи БЧКІ-1-17
спеціальності
123 «Комп'ютерна інженерія»
Харенко О. О.

Керівник: к.ф.-м.н., доцент
Ярмоленко М.В.
Рецензент: к.т.н., доцент
Міценко С.А.

Черкаси 2021

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет *ринкових, інформаційних та інноваційних технологій*

Кафедра *інформаційно-комп'ютерних технологій та фундаментальних дисциплін*

Спеціальність *123 «Комп'ютерна інженерія»*

Освітня програма *«Комп'ютерна інженерія»*

ЗАТВЕРДЖУЮ

Завідувач кафедри ІКТ та ФД

_____ М.В. Ярмоленко
(підпис)

« _____ » _____ 20__ р.

ЗАВДАННЯ

НА ДИПЛОМНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Харенку Олександрю Олександровичу

1. Тема дипломної бакалаврської роботи: Програмна реалізація адаптивного навчання по напрямку Front-end

Керівник роботи к.м.-м.н., доцент Ярмоленко М.В.

затверджені наказом КНУТД від «15» березня 2021 р. № 75-уч.

2. Строк подання студентом дипломної роботи: 01 червня 2021 р.

3. Вихідні дані до дипломної бакалаврської роботи: Наукові джерела та статичні дані щодо додатку, інструменти розробки, покрокова інструкція створення додатку

4. Зміст дипломної бакалаврської роботи (перелік питань, які потрібно розробити): Вступ. Розділ 1. Опис предметної області та дослідження актуальності розробки. Розділ 2. Аналіз програмних аналогів. Розділ 3. Опис програмної реалізації адаптивного навчання по напрямку frontend. Висновки. Список використаних джерел

5. Дата видачі завдання 15.03.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного магістерської роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	26.03.2021	Виконано
2	Розділ 1 Опис предметної області та дослідження актуальності розробки	01.04.2021	Виконано
3	Розділ 2 Аналіз програмних аналогів	23.04.2021	Виконано
4	Розділ 3 Опис програмної реалізації адаптивного навчання по напрямку frontend	14.05.2021	Виконано
5	Висновки	23.05.2021	Виконано
6	Оформлення дипломної роботи (чистовий варіант)	26.05.2021	Виконано
7	Здача дипломної роботи на кафедру для рецензування (за 14 днів до захисту)	31.05.2021	Виконано
8	Перевірка дипломної роботи на наявність ознак плагіату (за 10 днів до захисту)	04.06.2021	Виконано
9	Подання дипломної роботи на затвердження завідувачу кафедри (за 7 днів до захисту)	07.06.2021	Виконано

Студент

(підпис)

Харенко О.О.

(прізвище та ініціали)

**Науковий керівник
роботи**

(підпис)

Ярмоленко М.В.

(прізвище та ініціали)

Рецензент

(підпис)

Міценко С.А.

(прізвище та ініціали)

АНОТАЦІЯ

Даний дипломний проект, представляє собою автономну, інтерактивну онлайн-систему навчання по веб-розробці, а саме по одній з її частин front-end.

Проект буде онлайн-курсом, де кожен бажаючий, не залежно від віку, може навчитися всього, що стосується такої важливої частини веб-розробки, як front-end. Все, що потрібно від користувача даного проекту, це тільки терпіння і бажання вчитися й пізнавати нове.

Обсяг бакалаврської роботи складає 62 сторінок, з них 8 додатки. При написанні бакалаврської роботи була використана література та Інтернет джерела.

Ключові слова: Веб-технології, Front end, Front-end розробник.

ANNOTATION

This diploma project is an autonomous, interactive online learning system for web development, namely one of its parts front-end.

The project will be an online course where everyone, regardless of age, can learn everything related to such an important part of web development as front-end. All that is required from the user of this project is only patience and desire to learn and learn something new.

The volume of the bachelor's thesis is 62 pages, including 8 appendices. Literature and Internet sources were used in writing the bachelor's thesis.

Key words: Web technologies, Front end, Front-end developer.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ РОЗРОБКИ	10
1.1 Опис принципів тестування	10
1.2 Переваги вивчення напрямку Front-end.....	21
1.3 Постановка задачі.....	36
РОЗДІЛ 2 АНАЛІЗ ПРОГРАМНИХ АНАЛОГІВ	38
2.1 Опис платформи Coursera	38
2.2 Опис відкритого онлайн-курсу Udemy	40
2.3 Опис веб-платформи для навчання freeCodeCamp.....	43
2.4 Опис онлайн-колекції завдань Edabit.....	46
РОЗДІЛ 3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ АДАПТИВНОГО НАВЧАННЯ ПО НАПРЯМУ FRONT-END	49
3.1 Опис середовища розробки.....	49
3.2 Опис додатку «FrontEndBox»	60
ВИСНОВКИ	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66

ВСТУП

Веб-технології – це різні інструменти та техніки, які використовуються в процесі зв'язку між різними типами пристроїв через Інтернет. Щоб краще зрозуміти цей термін, розіб'ємо його на дві частини: „веб” та „технологія”. У цьому випадку Інтернет стосується Всесвітньої павутини, більш відомої як WWW (World Wide Web). Вперше він з'явився у 1989 році, коли відомий вчений та інженер Тім Бернерс-Лі створив ефективний механізм розподілу ресурсів між науковцями усього світу.

Поняття мережі можна легко пояснити на прикладі. Припустимо, ви шукаєте зображення милої лами. Ви вводите "www.google.com" у свій веб-переглядач. Ви вводите «мила лама» у рядок пошуку та натискаєте на вкладку «Зображення». Google надає вам безліч знімків з лами, зібраних з мільйонів різних джерел.

Всі веб-сторінки, документи та будь-які інші ресурси ідентифікуються та розміщуються за допомогою їх URL-адрес. Вони колективно утворюють те, що ми називаємо Всесвітньою павутиною.

Оскільки у нас є обмежений простір для вивчення широкого спектру доступних веб-технологій, ми вибрали кілька, які найчастіше використовуються як початківцями, так і експертами в цій галузі.

Веб-технології діляться на наступні розділи:

1. Основи, які охоплюватимуть веб-браузери та деякі основи розробки веб-додатків
2. Мови програмування та фреймворки, які використовуються при розробці веб-сайтів
3. Бази даних, які використовуються на сервері для зберігання даних, необхідних або зібраних веб-сайтами
4. Деякі протоколи, тобто правила спілкування в Інтернеті
5. Графічні, аудіовізуальні та інші мультимедійні елементи

6. Деякі формати даних, які зазвичай використовуються для передачі даних через Інтернет

7. Інші різні веб-технології

Веб-браузери, які просто називають браузерами, дають нам змогу переглядати всі ресурси, що є частиною Всесвітньої мережі. Вони засновані на архітектурі клієнт-сервер. У цьому випадку клієнтом є браузер. Можна сприймати сервер, як комбінацію програмного та апаратного забезпечення, яке приймає запити клієнта, а потім надсилає запитуваний ресурс клієнту.

Щоразу, коли вводиться URL-адреса в адресний рядок веб-переглядача, він передає запит серверу, а потім отримує та відображає те, що користувач просив.

Деякі популярні веб-браузери – Opera, Mozilla Firefox, Google Chrome та Safari.

Технології відіграють важливу роль у нашому повсякденному житті. Вони оточують нас і потрібні нам для виконання простих завдань і спрощення нашого життя. Усе програмне забезпечення, яке ми використовуємо, виготовляється розробниками програмного забезпечення (програмістами).

Однією з таких важливих технологій є веб-розробка. В нашу епоху інтернет-розвитку, веб-технології стали вкрай потрібними для людей. В процесі веб-розробки, створюють веб-сторінки та програми для користування в Інтернеті. Веб-розробка не обов'язково фокусується на дизайні веб-сайту; швидше, це стосується в основному частини програмування та кодування, що є основною складовою функціонування веб-сайту.

Це стосується розробки та обслуговування веб-сторінок, включаючи такі поняття, як веб-дизайн, веб-програмування, веб-публікація та управління базами даних. Крім того, він включає різні типи засобів та методів створення веб-контенту, такі як текстові редактори для ручного редагування веб-сайтів, Dreamweaver для розробки веб-сторінки, використання веб-сайту для ведення блогу, для оновлення блогів тощо.

Починаючи від базових і простих веб-сайтів, закінчуючи складними веб-додатками та платформами соціальних медіа, а також від численних веб-сторінок,

що дають змогу здійснювати покупки в Інтернеті, і закінчуючи навіть системами управління контентом (CMS). Усі онлайн-інструменти та веб-сайти, якими ми регулярно користуємося, є частиною веб-розробки. Крім того, всі ці інструменти та веб-сайти створені веб-розробниками.

Веб-розробка загалом стосується завдань, пов'язаних з розробкою веб-сайтів для розміщення в Інтернеті. Процес веб-розробки включає, серед інших завдань, веб-дизайн, розробку веб-контенту, сценарії на стороні клієнта/сервера та налаштування безпеки мережі.

У більш широкому розумінні веб-розробка охоплює всі дії, оновлення та операції, необхідні для побудови, обслуговування та управління веб-сайтом, щоб забезпечити його ефективність, зручність роботи та швидкість оптимальними.

Також, веб-розробка включає в себе всі ті стратегічні дії, необхідні для забезпечення належного ранжування за результатами пошукової системи. Зазвичай ці завдання стосуються іншої спеціалізації, а саме оптимізації пошукових систем (SEO).

Веб-розробка – це кодування або програмування, що забезпечує функціональність веб-сайту відповідно до вимог власника. В основному мова йде про недизайнерський аспект побудови веб-сайтів, який включає кодування та розмітку.

Веб-розробка варіюється від створення простих текстових сторінок до складних веб-додатків, соціальних мереж та електронних бізнес-додатків.

1. Ієрархія веб-розробки така:
2. Кодування на стороні клієнта;
3. Кодування на стороні сервера;
4. Технологія баз даних.

Більшість веб-розробників використовують для розробки веб-сайтів мову розмітки гіпертексту (HTML), каскадні таблиці стилів (CSS) та мову програмування JavaScript.

РОЗДІЛ 1

ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

ТА ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ РОЗРОБКИ

1.1 Опис принципів тестування

Тест - це освітнє оцінювання, призначене для вимірювання знань, навичок, здібностей, фізичної підготовки або кваліфікації учасника тестування з багатьох тем. Тест може проводитись усно, на папері, на комп'ютері або в заздалегідь визначеній області, що вимагає від учасника тестування продемонструвати набір навичок.

Тести різняться за стилем, строгістю та вимогами. Немає загального консенсусу або незмінного стандарту для тестових форматів та складності. Часто формат і складність тесту залежать від освітньої філософії викладача, тематики, розміру класу, політики навчального закладу та вимог акредитації або керівних органів.

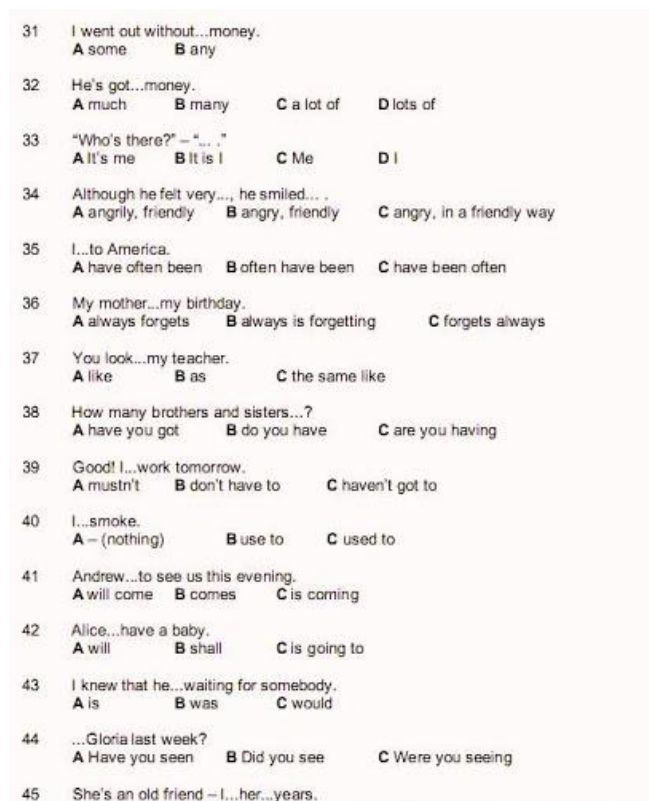


Рисунок 1.1 – Уривок шкільного тесту з англійської мови

Тест може проводитися офіційно або неофіційно. Прикладом неформального тесту є тест на читання, який батько проводить дитині. Офіційним тестом може бути підсумковий іспит, який проводить викладач у класі, або тест IQ, який проводить психолог у клініці.

Тестовий бал може бути інтерпретований з урахуванням норми чи критерію, а іноді і того, і іншого. Норму можна встановити самостійно або шляхом статистичного аналізу великої кількості учасників.

Тести використовуються в більшості освітніх систем. Тести можуть варіюватися від коротких, неофіційних запитань, обраних вчителем, до основних тестів, до яких студенти та викладачі готуються місяцями.

Деякі країни, такі як Великобританія та Франція, вимагають від усіх учнів середньої школи складання стандартизованого тесту з окремих предметів, таких як Загальний сертифікат середньої освіти (GCSE) (в Англії), як вимоги до закінчення школи. Ці тести використовуються насамперед для оцінки знань студента з певних предметів, таких як математика, природознавство чи література. Навпаки, старшокласники інших країн, таких як США, можуть не складати стандартизованого тесту для випуску. Більше того, студенти в цих країнах зазвичай проходять стандартизовані тести лише для того, щоб претендувати на посаду в університетській програмі, і, як правило, їм надається можливість проходити різні стандартизовані тести, такі як ACT або SAT, які використовуються насамперед для вимірювання навичок міркування студента. Учні середньої школи в США також можуть пройти тести підвищення кваліфікації з певних предметів, щоб отримати кредит на університетському рівні. Залежно від політики виробника тестів або країни, проведення стандартизованих тестів може відбуватися у великому залі, класі або тестовому центрі. Під час тестування може також бути присутнім відповідальний за правдиве складання тесту, який надає інструкції, відповідає на запитання або запобігає обману.

Оцінки за стандартизованим тестом можуть також використовуватися університетами для визначення того, чи слід приймати кандидата до однієї зі

своїх академічних чи професійних програм. Наприклад, університети Великобританії приймають абітурієнтів до своїх бакалаврських програм, заснованих переважно або виключно на оцінках здобувачів за довузівськими кваліфікаціями, такими як рівні GCE A або CambridgePre-U. На відміну від них, університети США використовують тестовий бал абітурієнта на SAT або ACT, як лише один із багатьох критеріїв прийому, щоб визначити, чи повинен абітурієнт бути прийнятим до однієї зі своїх програм бакалавр. Інші критерії в цьому випадку можуть включати оцінки здобувача середньої школи, позакласні роботи, особисті заяви та рекомендаційні листи.

Певні країни іноді використовують стандартизовані тести для управління якістю своїх навчальних закладів. Наприклад, закон США про заборону залишати дитину на інший рік, вимагає, щоб окремі штати розробляли оцінювання для учнів певних класів. На практиці ці оцінювання зазвичай виглядають у формі стандартизованих тестів. Потім тестові результати учнів конкретних класів навчального закладу використовуються для визначення статусу цього навчального закладу, тобто чи слід дозволити йому продовжувати працювати таким же чином або отримувати фінансування.

Нарешті, стандартизовані тести іноді використовують для порівняння знань студентів з різних установ чи країн. Наприклад, Організація економічного співробітництва та розвитку (ОЕСР) використовує Програму міжнародного оцінювання студентів (PISA) для оцінки певних навичок та знань студентів з різних країн-учасниць.

Стандартизовані тести іноді використовуються певними керівними органами, щоб визначити, чи дозволяється учаснику тестування займатися певною професією, використовувати конкретну назву посади чи претендувати на компетентність у певному наборі навичок. Наприклад, тестування учасника, який має намір стати юристом, зазвичай вимагається керівним органом, таким як урядовий орган з питань ліцензування адвокатури, для складання іспиту з адвокатської справи.

У деяких країнах також застосовуються стандартизовані тести для регулювання імміграції. Наприклад, іммігранти до Австралії, згідно із законом, зобов'язані здати іспит на громадянство в рамках процесу натуралізації країни.

Тести іноді використовують як інструмент для вибору учасників, які мають потенціал для успіху в змаганнях, таких як спортивні змагання. Наприклад фігуристи, які бажають взяти участь у змаганнях з фігурного катання, повинні пройти офіційні випробування, щоб пройти кваліфікацію.

Комп'ютерне тестування стосується проведення оцінювання за допомогою комп'ютерів, як альтернатива ручці та паперу. Таке тестування можна провести в Інтернеті або за допомогою автоматизованої системи. Це економічний та ефективний засіб для одночасного проведення масштабних онлайн-іспитів. Це дозволяє навчальним закладам швидко оцінювати результати роботи кандидатів незалежно від їх географічного розташування. Деякі його особливості – це онлайн-навігація, випадковий порядок запитань та одночасне управління студентами.

Оцінки використовувались у процесі відбору дуже давно. До цифрової ери оцінювачі використовували традиційні тести на основі підходів. Інтернет-технології стають невід'ємною частиною життя, а обчислювальні пристрої стають обов'язковою необхідністю. Комп'ютерний тест має деякі потенційні переваги порівняно з традиційними методами оцінки. Комп'ютерне тестування знаходить своє застосування в різних секторах, таких як освіта, урядовий відбір та корпоративний сектор. Усі зацікавлені сторони, залучені до веб-формату іспитів, такі як оцінювачі, учасники, адміністратори, селектори тощо, отримують вигоду від цього способу тестування.

Потреба у проведенні комп'ютерних іспитів справедливо зростає, враховуючи поточну ситуацію. Існує серйозне занепокоєння щодо безперервності освіти, оскільки університети та школи закриваються на невизначений час, що спричиняє затримку результатів іспитів. Комп'ютерні іспити забезпечують автоматичну оцінку та зворотний зв'язок за розділами. Вони забезпечують прості у використанні та надійні результати за допомогою передових технологічних

інновацій. Наприклад, відео або слайд-шоу для оцінки студентів-медиків під час хірургічного втручання або віртуальні симуляції для оцінки соціальних навичок революціонізували спосіб проведення іспитів. Корпоративний сектор також наголошує на використанні комп'ютерних тестів, щоб визначити найкращого кандидата на доступні посади. Це спрощує перший рівень відбору, коли бракує претендентів.



Рисунок 1.2 – Тестування за допомогою комп'ютера

Якщо говорити простими словами, комп'ютерне оцінювання - це практика присвоєння учням тестів на комп'ютері замість паперу. Комп'ютерне оцінювання вже широко використовуються в багатьох середовищах електронного навчання.

Цей тип тестування також використовується в школах, часто для проведення стандартизованих тестів, а також у коледжах і університетах та в програмах навчання працівників.

Комп'ютерне оцінювання – це не нова ідея – насправді вона існує в тій чи іншій формі з 1959 року, коли була побудована перша система комп'ютерного навчання. Однак, комп'ютерні оцінки не отримали особливого визнання кілька десятиліть тому, вони стали справді впливовими лише за останні кілька років, оскільки Інтернет-технології значно вдосконалилися з тих часів.

Оскільки комп'ютерне оцінювання настільки змінилася з моменту її створення, не завжди ясно, що люди мають на увазі, коли вживають цей термін. Найширше, визначення комп'ютерного оцінювання таке: «використання цифрових інструментів для діяльності, пов'язаної з оцінюванням». Будь-який тип тестування, який передбачає використання цих технологій, на відміну від паперу та олівців, можна розглядати під егідою комп'ютерного оцінювання.

Веб-іспити можуть бути використані для академічного оцінювання або під час масового найму, де оціночні тести відіграють важливу роль у процесі скринінгу. Керівництво також використовує комп'ютерне тестування для сприяння працівникам у їх кар'єрному просуванні. Розуміючи сильні та слабкі сторони кандидатів, команда навчання та розвитку може розробити адекватні навчальні програми з розвитку робочої сили. Добре інформована робоча сила підтримує бізнес стабільним та конкурентоспроможним.

Комп'ютери та технології вплинули майже на всі галузі у всьому світі. І, однією конкретною сферою, яка зазнала суттєвих змін, є галузь освіти.

Швидкість, з якою розвивався процес викладання, оцінювання та навчання, є достатньою, щоб підтвердити вплив. Книги перейшли в електронний формат, фізичні кабінети перетворились на онлайн-класи, а паперові тести на онлайн-тести.

Дуже довго, до появи комп'ютерів, тести на папері були єдиним методом оцінки, доступним викладачам та роботодавцям. Педагоги та викладачі записували набір запитань на папері, а потім роздавали ці роботи учням, чекали їх завершення, перш ніж оцінювати тести, писати відгуки та готувати звіти. Лише коли з'явилися системи оцінювання, засновані на технологіях, ідея оцінки ручкою та папером стала зайвою і поступово поступилася місцем кращій, ефективнішій системі оцінювання.

Такі привабливі функції, як доступність, масштабованість та безпека, роблять комп'ютерні онлайн-оцінювання найбільш бажаними, і це є основною причиною їх широкого прийняття. Тепер машина може робити все, що раніше робили викладачі, дозволяючи їм зосередитись на більш важливіших завданнях.

Типи оцінювань тестування:

1. Формативне оцінювання - це неформальні та офіційні тести, які проводяться в процесі навчання. Метою формативного оцінювання є моніторинг навчання студентів, щоб забезпечити постійний зворотний зв'язок, який може бути використаний викладачами для вдосконалення свого викладання, а студентами – для вдосконалення свого навчання.
2. Підсумкове оцінювання – оцінюють компетентність в кінці певної частини навчального процесу, з метою визначення, чи кандидат засвоїв знання чи навички до необхідного стандарту. Підсумкові оцінки можуть охоплювати кількадекільні інструктаж, роботу цілого курсу у таких випадках, як випускні іспити або навіть багаторічне навчання, у разі випускних іспитів у середній школі.
3. Нормове оцінювання – те, що відповідає нормам, порівнюють успішність студента з національною чи іншою групою "норми". Лише певний відсоток учасників тестування отримає найкращі та гірші результати. Посилання на норми зазвичай називають оцінкою по кривій, коли групою порівняння є учні в одному класі. Тести, повідомляють, чи відповідали учасники тесту краще чи гірше, ніж середній гіпотетичний студент, що визначається шляхом порівняння балів із результатами статистично відібраної групи учасників тесту, як правило, того самого віку чи рівня, які вже пройшли іспит.
4. Оцінювання по критерію – призначене для вимірювання успішності учнів на основі встановленого набору критеріїв або стандартів навчання.
5. Оцінювання на основі результатів – вимагає від студентів вирішення реальних проблем. Наприклад, студент може продемонструвати навички випічки, випікаючи торт, і оцінюючи результат за зовнішнім виглядом, смаком та текстурою.
6. Стандартизоване оцінювання – це всі тести, які вводяться та оцінюються послідовно, незалежно від того, це швидка вікторина, створена місцевим учителем, або ретельно досліджений тест, який дають мільйонам людей.

Стандартизовані тести часто використовуються в освіті, професійній сертифікації, психології, військовій галузі та багатьох інших сферах.

7. Нестандартизоване оцінювання – є гнучкими за обсягом та форматом та мінливими за складністю. Наприклад, учитель може обійти клас і задати кожному студентові різне запитання. Деякі питання неминуче будуть складнішими за інші, і вчитель може бути більш суворим до відповідей кращих учнів. Нестандартизований тест може бути використаний для визначення рівня кваліфікації студентів, мотивації студентів до навчання, надання зворотного зв'язку студентам та модифікації навчальної програми, щоб зробити її більш підходящою для студентів з низьким або високим рівнем кваліфікації.
8. Конкурсне оцінювання – це тести, де кандидати класифікуються відповідно до їх оцінок, а потім обираються найкращі по рейтингу. Якщо іспит відкритий для декількох позицій, то перші декілька кандидатів у званнях складають, інші відхиляються. Вони використовуються як вступні іспити для вступу до університетів. Конкурсні іспити вважаються рівноправним способом відбору гідних претендентів без ризику поширення впливу, упередженості чи інших проблем.



Рисунок 1.3 – Один з способів заповнення аркушу відповідей

Один тест може мати кілька якостей. Наприклад, адвокатський іспит для юристів-початківців може бути нормованим, стандартизованим, підсумковим оцінюванням. Це означає, що пройдуть лише ті, хто має вищі бали, що всі вони склали один і той же тест за однакових обставин і оцінювались однаковими стандартами балів. Тест повинен визначити, чи достатньо знають випускники юридичних шкіл, щоб працювати по своїй професії.

Види тестування:

1. Письмове тестування – це тести, які проводяться на папері або на комп'ютері. Учасник тесту, який складає письмовий тест, може відповісти на конкретні завдання, написавши або набравши текст у заданому просторі тесту або в окремій формі чи документі. Вибір розробником тесту, який стиль чи формат використовувати при розробці, є, як правило, довільним, оскільки немає єдиного інваріантного стандарту для тестування. Як би там не було, певні стилі та формат тестування стали більш широко використовуватися, ніж інші.
2. Тест з широким вибором – має елементи, відформатовані як запитання з декількома варіантами, кандидату буде дано кілька наборів відповідей на кожне питання, і кандидат повинен вибрати, яка відповідь або група відповідей є правильними. Як навчальний інструмент, завдання з декількома варіантами перевіряють багато рівнів знань учасника відразу і здатність інтегрувати інформацію. Тим не менше, існують труднощі, пов'язані з використанням питань із множинним вибором. Завдання з множинним вибором не дозволяють учасникам тесту продемонструвати знання, що перевищують запропонований вибір, і навіть можуть заохочувати здогадки чи припущення, через наявність принаймні однієї правильної відповіді. Нарешті, завдання з множинним вибором не перевіряють ставлення учасника тесту до навчання, оскільки правильні відповіді можна легко підробити.
3. Тест з альтернативною відповіддю – пропонує кандидатам бінарний вибір – твердження є істинним, або хибним. Цей метод має деякі проблеми в своїй

структурі, оскільки залежно від кількості питань, значна кількість кандидатів може отримати 100% лише шляхом здогадок, а в середньому повинна отримати 50%.

4. Тест відповідності – це вид тестування, який надає визначений термін і вимагає від учасника ідентифікаційних характеристик правильному терміну.
5. Тест завершення – елемент заповнення порожнього місця у визначенні, вимагає від учасника тесту згадати правильний термін. Існує два типи тестів заповнення порожнього місця. У спрощеній версії передбачено банк можливих слів, які заповняють пропуски. Для деяких іспитів усі слова в банку слів використовуються рівно один раз. Якщо вчитель хотів створити тест середньої складності, він би запропонував тест із банком слів, але деякі слова можуть використовуватися більше одного разу. Найскладнішим різновидом такого тесту є тест із заповненням порожнього місця, в якому взагалі не передбачено жодного банку слів. Це, як правило, вимагає вищого рівня розуміння та пам'яті, ніж тест із множинним вибором.
6. Есе – вид тестування, який, як правило, вимагає від учасника тестування розгорнутої відповіді, щоб задовольнити вимоги тесту. Як інструмент оцінювання, статті есе можуть перевірити складний рівень навчання. Такий вид тесту ускладнює учасникам тестування вгадування правильної відповіді.

З точки зору розробника тесту, існує велика мінливість щодо часу та зусиль, необхідних для підготовки тесту. Подібним чином, з точки зору учасника тестування, існує також велика варіативність щодо часу, необхідного для отримання бажаної оцінки з будь-якого тесту. Коли розробник конструює тест, кількість часу та зусиль залежить від значущості самого тесту, кваліфікації учасника тесту, формату тесту, розміру групи учасників, терміну тестування та досвіду розробника тесту.

Багато розробників тестів одночасно були студентами, і тому вони можуть змінювати або відверто брати запитання з попередніх тестів. У деяких країнах,

книговидавці часто надають навчальні пакети, що включають банк тестів, викладачам університетів, які приймають видані книги для своїх курсів. Ці банки тестів можуть містити до чотирьох тисяч зразків тестових запитань, які були перевірені і випробувані. Викладач, який вирішить використовувати цей банк тестів, повинен буде лише вибрати фіксовану кількість тестових питань для побудови тесту.

Як правило, нестандартизовані тести, які є короткими і не складають значної частини загальної оцінки курсу або балів, не вимагають від учасника тестування багато часу на підготовку до тесту. І навпаки, нестандартизовані тести, які є довгими, рідкісними і складають основну частину загальної оцінки курсу або балів, як правило, вимагають від учасника тестування витратити велику кількість часу на підготовку до тесту. Для підготовки до нестандартного тестування учасники тесту можуть покладатися на свої довідники, конспекти занять чи лекцій, Інтернет та минулий досвід. Учасники тестів можуть також використовувати різні навчальні засоби для підготовки, такі як картки та мнемотехніка. Майбутні учасники тестування, можуть навіть найняти викладачів, які навчатимуть їх у процесі навчання, щоб вони могли збільшити ймовірність отримання бажаної оцінки. У таких країнах як Великобританія, попит на приватне навчання значно зріс за останні роки. Учасники тестування можуть покладатися на копії тесту з попередніх років або семестрів навчання. Ці минулі тести можуть бути надані другом або групою, яка має копії попередніх тестів, викладачами та їх установами, або самим постачальником тесту.

На відміну від нестандартизованого тесту, час, необхідний учасникам тестування для підготовки до стандартизованих тестів, є менш мінливим і, як правило, значним. Це пояснюється тим, що стандартизовані тести, зазвичай, однакові за обсягом, форматом та складністю, і часто мають важливі наслідки щодо майбутнього учасника тестування, наприклад, право відвідувати певну університетську програму або займатися бажаною професією. Нерідкі випадки, коли учасники тестування готуються до стандартизованих тестів, спираючись на комерційно доступні книги, що забезпечують глибоке висвітлення

стандартизованого тесту або збірки попередніх тестів. У багатьох країнах учасники тестування навіть записуються до центрів підготовки, які надають обширні або додаткові інструкції учасникам тестування, щоб допомогти їм краще підготуватися.

Обман – це процес використання несанкціонованих засобів або методів з метою отримання бажаного балу тесту або оцінки. Це може варіюватися від принесення та використання приміток під час закритого огляду книги, копіювання відповіді іншого учасника тесту або вибору відповідей під час індивідуального тестування, до відправки довіреній особі для проходження тесту.

Для боротьби з шахрайством застосовано кілька загальних методів. Вони включають використання декількох інгібіторів під час періоду тестування для моніторингу учасників. Розробники тестів можуть сконструювати кілька варіантів одного і того ж тесту для одночасного введення різним учасникам тестування. У деяких випадках інструктори можуть не проводити власні тести, а залишатимуть завдання іншим інструкторам чи інспекторам, що може означати, що ті, хто контролює проведення тестування, не знатимуть кандидатів особисто, а отже зменшується вірогідність упередженості і зацікавленості.

1.2 Переваги вивчення напряму Front-end

Front-end веб-розробка, також відома як розробка на стороні клієнта – це створення веб-сайту або веб-програми, щоб користувач міг бачити та взаємодіяти з ними безпосередньо. Проблема, пов'язана з розробкою інтерфейсу, полягає в тому, що інструменти та методи, використані для створення інтерфейсу веб-сайту, постійно змінюються, і тому розробник повинен постійно бути в курсі того, як розвивається галузь.

У світі веб-розробки існують інтерфейсні та фонові роботи. На задній панелі програмісти виконують приховану роботу та процеси, які працюють за лаштунками, результат яких кінцевий користувач не бачить.

Front-end розробка, навпаки, стосується частин веб-сайту або програми, які кінцевий користувач насправді бачить і взаємодіє з ними. Як розробник

інтерфейсу, ви будете працювати над тим, щоб зробити веб-сайти красивими, функціональними та доступними. Front-end розробники також дуже залучені до взаємодії з користувачем, і нерідко тісно співпрацюють з дизайнерами та замовниками, щоб допомогти створити найкращі рішення для клієнтів.

Інтерфейс веб-сайту – це все, що ви бачите та те з чим можете взаємодіяти за допомогою браузера. Отже, створення цієї візуальної частини називається front-end розробкою. Можна навіть сказати, що дизайнери, що створюють користувальницькі інтерфейси, також є розробниками інтерфейсу, оскільки вони працюють у співпраці над тією ж частиною проекту.

Для створення інтерфейсу інженери використовують комбінацію HTML (для базової структури сторінки та вмісту), CSS (для візуального редагування) та JavaScript (для створення веб-сайтів інтерактивними). Той самий набір інструментів використовується для створення прогресивних веб-програм – мобільних додатків, які виглядають і відчують себе рідними в галузі мобільної розробки, але створюються із використанням інтерфейсних технологій.

Гіпертекстова мова розмітки (Hyper Text Markup Language), яку частіше називають HTML, – це місце, де розпочинається веб-сайт. Це найважливіша складова всіх вступних уроків, які охоплюють основи роботи в Інтернеті. Це мова клієнта, яка використовується для кодування інтерфейсу веб-сайту. Це допомагає розробникам визначити структуру веб-сторінки, і робить це за допомогою елементів, які ідентифікуються тегами. HTML визначає базову структуру веб-сайту – основу, на якій базується все інше. Він формує блоки, які визначають макет сторінки, формат та критичні компоненти.

Мова розмітки HyperText або HTML – це стандартна мова розмітки для документів, призначених для відображення у веб-браузері. Йому можуть допомогти такі технології, як каскадні таблиці стилів (CSS) та мови сценаріїв, такі як JavaScript.

Веб-браузери отримують документи HTML з веб-сервера або з локальної пам'яті та надають документи на мультимедійні веб-сторінки. HTML описує

структуру веб-сторінки семантично та включає ознаки зовнішнього вигляду документа.

Елементи HTML - це будівельні блоки HTML-сторінок. За допомогою HTML-конструкцій зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у відтворену сторінку. HTML забезпечує засіб для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи.

Елементи HTML розмежовані тегами, записаними за допомогою кутових дужок. Такі теги, безпосередньо вносять вміст на сторінку. Інші теги, такі як `<p>`, оточують та надають інформацію про текст документа і можуть включати інші теги як піделементи. Браузери не відображають теги HTML, але використовують їх для інтерпретації вмісту сторінки.

HTML може вбудовувати програми, написані мовою програмування, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає зовнішній вигляд та макет вмісту. Консорціум всесвітньої павутини (W3C), колишній супровідник HTML і поточний підтримувач стандартів CSS, заохочує використання CSS над явним презентаційним HTML з 1997 року.

```

141 <div class="wrapper">
142
143   <div class="row">
144     <div class="col-sm-6">
145       <h1>All-In-One SEO Software & Tools</h1>
146       <div id="greenhomepagebar" class="hidden-xs"></div>
147       <ul class="list-unstyled hidden-xs">
148         <li><span class="glyphicon glyphicon-ok"></span> Website Crawling and Site Auditor</li>
149         <li><span class="glyphicon glyphicon-ok"></span> Google Rank Tracker with Daily Updates</li>
150         <li><span class="glyphicon glyphicon-ok"></span> Backlink Checker and Link Building Tools</li>
151         <li><span class="glyphicon glyphicon-ok"></span> Continuous Monitoring and Whitelabel Reporting</li>
152       </ul>
153
154       <div class="registerplease margin-top">
155         <div class="row">
156           <div class="col-md-6 center margin-top">
157             <a href="https://www.seobility.net/en/register/step1" class="btn btn-success registerbutton">Sign Up Free</a>
158           </div>
159         </div>
160       </div>
161     </div>
162
163   </div>
164
165
166   <div class="col-sm-6 hidden-xs">
167     
168   </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 <div class="wrapper" id="homepagefeaturebox">
174   <div class="row featureshowcaseheadline">
175     <div class="col-md-12 center">
176       <h2>Online SEO Software for better website optimization</h2>
177       <p class="hidden-xs">Seobility comes with all SEO analyses and tools you need</p>

```

Рисунок 1.4 – Приклад коду на HTML

Хоча теоретично можливо розробляти веб-сайт лише у форматі HTML, це буде просто сайт без функціональної сторони, якщо він не розширений за допомогою CSS та JavaScript. Крім того, навіть такі прості модифікації стилю, як зміна кольору кнопки, вимагають великої кількості коду, який виконується не лише за допомогою HTML. Кожен елемент, укладений у дужки <>, є елементом структури основи сайту. Однак DOCTYPE – це лише декларація, яка повідомляє браузеру, що те, що він збирається завантажити, – це документ HTML. Елемент <body> визначає весь вміст, який утворює тіло документа. <h1> та <p> позначають початок заголовка та абзацу відповідно. </body>, </h1> та </p> закривають теги. Вони вказують, де закінчуються текст, заголовок та абзац.

```
h1 {
  font-family: courier, courier-new, serif;
  font-size: 20pt;
  color: blue;
  border-bottom: 2px solid blue;
}
p {
  font-family: arial, verdana, sans-serif;
  font-size: 12pt;
  color: #6B6BD7;
}
.red_txt {
  color: red;
}
```

Рисунок 1.5 – Приклад CSS-коду

CSS – одна з найбільш фундаментальних технологій дизайну веб-сайтів. Щоб ще більше прикрасити веб-сторінки, можна інтегрувати CSS (каскадні таблиці стилів) у свій HTML-код. Процес застосування певних стилів до певних елементів HTML-коду може перетворитися на повторюване завдання. Для економії часу та енергії CSS можна використовувати для впорядкування процесу стилізації сторінок на всьому сайті. CSS використовується для стилізації вмісту

веб-сайту. За кожним елементом або тегом сайту, закріплений той чи інший набір CSS-інструкцій. Кожна із цих інструкцій, відповідає за зовнішні зміни того самого елемента, за яким вона закріплена. Усі стилістичні зміни кольорів, форм, зовнішнього вигляду – саме за це відповідає CSS.

Каскадні таблиці стилів (CSS) - це мова таблиці стилів, що використовується для опису презентації документа, написаного мовою розмітки, такою як HTML. CSS – це стильова технологія Всесвітньої павутини, поряд з HTML та JavaScript.

CSS призначений для розділення презентації та вмісту, включаючи макет, кольори та шрифти. Це розділення може покращити доступність вмісту, забезпечити більшу гнучкість і контроль у специфікації характеристик сайту, дозволити декільком веб-сторінкам спільне форматування, вказавши відповідний CSS в окремому файлі .css, що зменшує складність та повторюваність структурного вмісту, а також дозволяє файл .css, який буде кешовано, щоб покращити швидкість завантаження сторінки між сторінками, що ділять файл, та його форматування.

Поділ форматування та вмісту також робить можливим представлення однієї і тієї ж сторінки розмітки в різних стилях для різних методів візуалізації, таких як екран, друк, голосом (через мовний браузер або зчитувач екрана) та на основі Брайля тактильні пристосування. CSS також має правила альтернативного форматування, якщо доступ до вмісту здійснюється на мобільному пристрої.

Мова програмування JavaScript використовується для того, щоб подбати про

```
10
11   xhr.open('POST', loginUrl, true);
12   xhr.setRequestHeader('Content-Type', 'application/json; charset=UTF-8');
13   xhr.addEventListener('load', function() {
14     var responseObject = JSON.parse(this.response);
15     console.log(responseObject);
16     if (responseObject.token) {
17       tokenElement.innerHTML = responseObject.token;
18     } else {
19       tokenElement.innerHTML = "No token received";
20     }
21   });
22
23   var sendObject = JSON.stringify({name: user, password: password});
24
25   console.log('going to send', sendObject);
```

Рисунок 1.6 – Приклад коду на JavaScript

інтерактивність багатьох унікальних елементів веб-сайту. З його допомогою можна створювати ефекти, які змінюють зовнішній вигляд піктограм та випадаючих меню, додають анімацію, ігри та інші інтерактивні елементи. Вперше JavaScript було створено в 1995 році. Сьогодні він улюблений розробниками по всьому світу. Зараз це життєво важливий компонент веб-розробки, який допомагає створювати динамічні елементи веб-сторінок. Введення JavaScript привело до революційної хвилі інтерактивних веб-сторінок. Сьогодні включення графічних елементів, інтеграція баз даних та загальна динаміка веб-сайту можливі завдяки появі JavaScript. Хоча JavaScript спочатку був представлений як мова інтерфейсу розробки, зараз він стає популярним вибором і для внутрішньої розробки.

JavaScript часто скорочується як JS, є мовою програмування, яка відповідає специфікації ECMAScript. JavaScript є високорівневим та багатопарадигмою. Він має синтаксис фігурних дужок, динамічне введення тексту, орієнтацію на об'єкти на основі прототипу та функції першого класу.

Поряд з HTML і CSS, JavaScript є однією з основних технологій Всесвітньої мережі. Понад 97% веб-сайтів використовують його на стороні клієнта для розробки інтерактивної поведінки веб-сторінок, часто включаючи сторонні бібліотеки. Усі основні веб-браузери мають спеціальний механізм JavaScript для виконання коду на пристрої користувача.

Як мова багатопарадигми, JavaScript підтримує керовані події, функціональні та імперативні стилі програмування. Він має інтерфейси прикладного програмування (API) для роботи з текстом, датами, регулярними виразами, стандартними структурами даних та об'єктною моделлю документа (DOM).

Стандарт ECMAScript не включає жодного введення / виводу (вводу / виводу), наприклад, мережевих чи графічних засобів. На практиці веб-браузер або інша система виконання забезпечує API для вводу-виводу.

Спочатку двигуни JavaScript використовувались лише у веб-браузерах, але зараз вони є основними компонентами інших програмних систем, зокрема серверів та різноманітних додатків.

Незважаючи на те, що між JavaScript та Java є подібність, включаючи назву мови, синтаксис та відповідні стандартні бібліотеки, ці дві мови сильно відрізняються за дизайном і механікою роботи.

Веб-розробники, як правило, поділяються на розробників зовнішнього інтерфейсу (front-end developer), розробників внутрішньої частини (back-end developer) та розробників, які володіють навиками і тих і інших (fullstack developer). Frontend стосується всіх тих частин веб-сайту, які користувач може бачити на своєму екрані та взаємодіяти.

Backend стосується прямо протилежного цьому. Він включає приховані механізми, які роблять функцію веб-сторінки. Типовий користувач, як правило, не знає, що відбувається у серверній системі.

Fullstack-розробники, здатні виконувати одночасно і інтерфейсні, і серверні завдання.

Веб-протоколи – це деякі заздалегідь визначені правила, яких повинен дотримуватися кожен, хто спілкується через Інтернет.

Протокол передачі гіпертексту (Hyper Text Transfer Protocol), більш відомий як HTTP – це веб-протокол, який визначає два поняття:

1. Як клієнтські запити передаються на сервери.
2. Як сервери реагують на запити клієнтів.

У запиті повинні бути зазначені наступні чотири пункти:

1. URL-адреса ресурсу, який бажає клієнт.
2. Метод, такий як GET (який використовується для отримання даних із сервера) або POST (який використовується для надсилання даних, які потрібно оновити на сервері).
3. Список заголовків (заголовки запитів можуть містити різну інформацію про клієнта або ресурс, який вони запитували, наприклад,

ім'я браузера клієнта, операційну систему, розмір повідомлення, час і дату запиту, інформацію про контроль доступу / дані тощо).

4. Тіло, яке містить інформацію, яку клієнт хоче надіслати на сервер.

Після обробки запиту сервер надсилає відповідь з іншого кінця. Ця відповідь визначає три елементи:

1. Код стану у формі тризначного номера (Код стану, який ви могли бачити під час щоденних сеансів веб-перегляду, – 404, що вказує на те, що запитаний ресурс не вдалося знайти).
2. Список заголовків (заголовки відповідей за структурою нагадують заголовки запитів і можуть містити таку інформацію, як розмір повідомлення, тип вмісту тощо).
3. Тіло, яке містить інформацію, яку клієнт запитував від сервера.

Всесвітня павутина зростатиме з дивовижними темпами, включаючи більшу кількість людей, пристроїв та ресурсів, доступних в Інтернеті. Тому очікується розширення веб-технологій. Отже, технічні фахівці будуть прагнути до того, щоб забезпечити доступний веб-вміст для користувачів скрізь. Для захисту приватного веб-вмісту від зловмисних кібератак доведеться вжити більш жорстких заходів безпеки.

WWW розпочався як Web 1.0 зі статичними веб-сторінками, написаними простим старим HTML. Потім він перейшов до Web 2.0 з появою JavaScript та інших технологій, що дозволило користувачам взаємодіяти з веб-контентом та створювати власні.

Незабаром Інтернет піде на крок далі і вступить у свою третю фазу, тобто Web 3.0, яка буде набагато децентралізованішою, ніж сьогодні. Сучасні технології, такі як AI та IoT, незабаром домінуватимуть в Інтернеті. Все буде набагато автоматизованіше як з точки зору розробки, так і споживання нового контенту.

Завдання проекту веб-сайту полягає у забезпеченні того, щоб користувачі, відкриваючи сайт, бачили інформацію у форматі, який легко читати. Це ускладнюється тим, що користувачі зараз використовують велику кількість

пристроїв із різними розмірами екрана та роздільною здатністю, що змушує дизайнера враховувати ці аспекти при розробці сайту. Їм потрібно переконатися, що їх сайт правильно відображається в різних браузерах (крос-браузерність), різних операційних системах (крос-платформа) та різних пристроях (крос-пристрій), що вимагає ретельного планування з боку розробника.

Front-end розробник розробляє веб-сайти та додатки з використанням веб-технологій (тобто HTML, CSS, DOM та JavaScript), які працюють на Open Web Platform або виступають в якості компіляції для середовищ, що не пов'язані з веб-платформою (тобто React Native).

В сучасному світі, програмування займає, без сумнівів, одну з провідних позицій в становленні науково-технічного прогресу. Це і не дивно, оскільки майже жодний технічний пристрій не може обійтися без програмного забезпечення, виготовленням якого і займаються програмісти.

Але, які ж основні переваги вивчення саме напряму Front-end?

1. Front-end розробники користуються попитом. Незалежно від того, чи хочете ви працювати в організації, або як самозайнятий фрілансер, front-end розробники користуються попитом серед роботодавців. Насправді Creative Group назвала front-end розробку однією з 10 найкращих робочих місць у галузі творчості та маркетингу 2017 року.
2. Це прибуткова кар'єра. Front-end розробники отримують гідну платню. Згідно з статистикою по заробітній платі компанії Creative Group, front-end розробник із трирічним досвідом роботи, може заробляти від 80 000 до 119 500 доларів.
3. Гнучкість. Як розробник інтерфейсу ви можете працювати з будь-якого місця, включаючи будинок. Це робить front-end одним з найбільш гнучких напрямів, які ви можете вибрати. Набір навичок, який ви набуваєте, дозволяє вам працювати в різних областях, таких як редагування зображень, UX / UI, мобільна розробка тощо.

4. Творчість. Front-end розробка – це все для творчості та творчого вирішення проблем. Як веб-розробник інтерфейсу, ви матимете навички створювати власні веб-сайти, щоб забезпечити кращий досвід користувачів.
5. Багато варіантів кар'єри. Присутність в Інтернеті стала головним фактором у залученні клієнтів / користувачів, що є постійним викликом для компаній. Якщо ви вирішите працювати в корпораціях, некомерційних організаціях або стати самозайнятим фрілансером, ви можете створити успішну кар'єру.
6. Професія, яка постійно розвивається. Front-end розробка – це захоплююча кар'єра, оскільки вона завжди розвивається і постійно змінюється. Це означає, що у вас завжди буде можливість взаємодіяти з новими інструментами та освоювати нові навички. Це сфера, яка надає багато можливостей для практичного навчання.

Саме слово «фронт» говорить про те, що фронт-енд розробник знаходиться на передовій лінії з кінцевим користувачем, і відповідає за зовнішній функціонал додатка, з яким контактує користувач. Програмним кодом на сервері займається інший фахівець – бек-енд розробник. До відома: існує універсальний фахівець, який працює одночасно на фронт-енд і бек-енд. Він називається фул-стак розробник (з англ. «Fullstackdeveloper»).

JavaScript – головна мова фронт-енд розробника в web, на якому він здійснює програмування користувальницького інтерфейсу. Основне завдання front-ендрозробника – верстка сайтів: написання HTML-коду і CSS-стилів для подання інформації в зрозумілому всім браузерам вигляді і зовнішньому відповідності сайту макету дизайнера. Часто буває, що макети не відображають всіх деталей загальної ідеї. В даному випадку фронт-енд розробник включає свої дизайнерські здібності, щоб врахувати всі технічні деталі і нюанси. Є багато додаткових інструментів, які автоматизують рутинну частину роботи: Sass / SCSS, jQuery, LESS, AngularJS, Bootstrap, Prototype, Ember.js, Backbone, React.js, GruntGulp. А поява ChromeDevTools і Firebug дозволило підвищити продуктивність праці.

Фронт-енд розробник в web відповідає за те, щоб сторінки сайту виглядали у всіх браузерах однаково і відповідали макету дизайнера. З появою нових засобів розробки і більш сучасних браузерів це завдання вирішується простіше. Але інтерфейси сучасних сайтів або інтернет-сервісів стають складнішими, а різноманіття пристроїв, з яких користувачі виходять в мережу, постійно розширюється – ноутбуки, планшети, смартфони. Тому фронт-енд розробнику дуже важливо, як і будь-якому програмісту, постійно перебувати в курсі всіх нововведень і фішок, щоб успішно адаптувати сайти під нові пристрої.

Frontend – це публічна частина web-додатків (вебсайтів), з якої користувач може взаємодіяти і контактувати напряму. У Frontend входить відображення функціональних завдань, призначеного для користувача інтерфейсу, що виконуються на стороні клієнта, а також обробка запитів користувачів. По суті, фронтенд– це все те, що бачить користувач при відкритті web-сторінки.

У свою чергу, web-додаток – клієнт-серверний додаток, в якому клієнтом виступає в основному браузер, а сервером –web-сервер. Логіка web-додатки розподілена між сервером і клієнтом, зберігання даних здійснюється переважно на сервері, обмін інформацією відбувається по мережі. Простіше кажучи, це те, що бачить користувач і які дії виконує кожен раз, коли підключається до мережі інтернет і відкриває будь-який браузер.

Frontend-розробка – це робота зі створення публічної частини web-додатки, з якої безпосередньо контактує користувач, і функціоналу, який зазвичай виконується на стороні клієнта. Тобто, фронтенд розробник працює над тим, щоб на сайті кожна кнопка, іконка, текст і вікно не тільки стояли на своєму місці, не перекривали один одного і виглядали цілісно (це веб-верстка), але і щоб вони виконували своє пряме призначення – виробляли якісь дію (наприклад, щоб кнопка "купити" відкривала кошик, а "play" запускала відтворення фільму або музики).

З метою створення затребуваного і доступного продукту (веб-додатки) фронтенд-розробнику необхідно взаємодіяти з іншими програмістами, дизайнерами, маркетологами, аналітиками та іншими фахівцями.

Функціонал фронт-енд розробника в web виглядає наступним чином:

1. створення HTML-сторінки сайту на основі дизайн-макетів;
2. верстка сайту і шаблонів для CMS;
3. прив'язка до призначеного для користувача інтерфейсу скриптів, які забезпечують візуалізацію і анімацію сторінок сайту;
4. забезпечення необхідного рівня користувальницького інтерфейсу (UI –User Interface) і досвіду взаємодії (UX –Uzer Experience).

Зараз є певний перекис в сторону фронтенд-розробників, які виробляють wow-ефект на клієнтів. В Україні зараз активно шукаються React / Vue / Angular розробники. І нерідко зарплата таких фронтенд-фахівців з досвідом 1-2 роки із зарплати порівнянна з середньостатистичним Java-бекендером з досвідом 6-8 років.

Властивості якими повинен володіти front-end розробник:

1. Аналітичний розум
2. Відповідальність
3. Увага до дрібниць
4. Творчі здібності
5. Здатність дотримуватися дедлайни
6. Вміння працювати в команді
7. Уміння об'єктивно оцінювати свої можливості для вирішення поставлених завдань
8. Спрямованість на результат
9. Постійне бажання освоювати нові технології та інструменти

В Україні є дуже багато передових ІТ компаній, які відомі по всьому світу:

1. АМС Bridge. До рейтингу включена вперше. Українська компанія з офісами у Дніпрі, Сумах, Хмельницькому та Чернівцях;
2. Сіklum. Входить до ТОП-5 найбільших ІТ-компаній України за версією DOU (2500+ спеціалістів). Дебютувала у списку ІАОР два року тому. І у 2016, і у 2017 роках укладачі рейтингу особливо відмічали інноваційність

- компанії у наданні послуг, позитивні відгуки клієнтів, проекти з корпоративної соціальної відповідальності;
3. ELEKS. В Україні компанія налічує понад 1100 фахівців (9 місце за версією DOU). Вперше потрапила до рейтингу у 2015 році, опинившись одразу у кількох підкатегоріях кращих компаній, що розвиваються (Rising Star). Впродовж останніх років укладачі рейтингу відзначають інноваційність у наданні послуг, клієнтські відгуки, міжнародні нагороди, здобуті компанією;
 4. Inforpulse. До рейтингу IAOP входить вперше. Компанію засновано у 1991 році у Києві. Сьогодні займає шосту сходинку у ТОП-50 IT-компаній України за версією DOU (1600+ спеціалістів);
 5. Innovacs. Також є дебютантом рейтингу. Є американською компанією зі штаб-квартирою у Києві, заснована українцем. 36-те місце у рейтингу DOU (440+ спеціалістів в Україні);
 6. Intellias. Новачок рейтингу. Входить до ТОП-50 IT-компаній України за версією DOU (800+ спеціалістів, 16-та позиція у списку);
 7. Miratech. 780+ спеціалістів в Україні (17 місце у рейтингу DOU). Компанія є одним із вітчизняних ветеранів рейтингу IAOP, входячи до списку кращих світових постачальників IT-послуг вже 6 років поспіль. Серед сильних сторін, завдяки яким компанію включають до рейтингу, - позитивні відгуки, інноваційність, наявність міжнародних нагород та сертифікатів;
 8. N-iX. Відносний новачок рейтингу – дебют компанії у переліку найкращих відбувся у 2017 році. Важливо зазначити, що в 2016 році IAOP окремо, поза рейтингом, відмітили позитивну динаміку росту доходів та кількості спеціалістів компанії. На сьогодні в компанії працює 745 спеціалістів в Україні та вона посідає 18 стрічку рейтингу DOU ТОП-50;
 9. Program-Ace. Вперше харківська компанія з'явилася у The Global Outsourcing 100 торік, привернувши увагу укладачів позитивними відгуками клієнтів та проектами у сфері корпоративної соціальної відповідальності;

10.Sigma Software. У рейтингу з 2016 року. Вперше увійшла до списку завдяки позитивним клієнтським відгукам, у тому ж році окремо були виділені кількість спеціалістів, темпи зростання та обсяги виручки у категорії Rising Star. В минулому році IAOP, окрім відгуків клієнтів компанії, відмітили і її проекти у сфері корпоративної соціальної відповідальності;

При описі професійних обов'язків Frontend-розробника часто виникає плутанина, що багато в чому обумовлено політикою самих роботодавців, які хочуть отримувати відразу все, та ще й як можна дешевше. Тому у вакансіях для «верстальників» нерідко можна зустріти вимоги, як для «фронтенда-програмістів». Ну а серед описів вакансій для «фронтенда» вказуються в якості обов'язкових навички backend-розробників, що фактично робить з здобувача повноцінного web- або fullstack-програміста.

Цікава ситуація склалася з найменуваннями самих вакансій. Справа в тому, що frontend- і react-, javascript- і angular-, а також web-розробники – все тим чи іншим чином пов'язані з фронтенда.

Frontend-розробнику доводиться ставати хорошим верстальником. Чи не рівня Senior-фахівця, але він повинен справлятися з версткою макета будь-якої складності. До ключових навичок верстальника відносяться:

1. Кросбраузерності адаптивна верстка отриманих від web-дизайнера макетів в PSD-форматі – зручне відображення веб-сайту в існуючих браузерах на екранах будь-яких пристроїв;
2. Кросбраузерності адаптивна верстка e-mail листів для розсилок – кожен поштовий клієнт читає код по-своєму, а потрібно, щоб лист відображалося скрізь однаково;
3. Валідна верстка – відповідність стандартам W3C;
4. Семантична верстка – осмислене розташування фрагментів коду на сторінці, правильне використання тегів, зрозумілі назви класів і ідентифікаторів;

5. Оптимізована під пошукові системи верстка – швидкість завантаження, структурований код, акценти для пошукових систем, теги title, description і alt + title для зображень;
6. Верстка під CMS (готові движки сайтів) – розробка цілісних шаблонів і їх «натяжка» на CMS.

Інструменти і технології верстальників, які повинен знати кожен розробник:

1. SMACSS, а також BEM і ще OOCSS – це прийняті в усьому світі стандарти розробки структурної частини CSS;
2. AJAX / jQuery – потрібні для розробки будь-яких динамічних елементів на веб-сторінках, а ще для створення форм онлайн-заявок;
3. CSS-препроцесори – дають можливість розробляти CSS набагато швидше;
4. Photoshop – потрібен для поділу PSD-макетів на складові елементи;
5. SVG / Canvas - для обробки застосовуються зображень;
6. HTML / CSS – потрібно максимальний рівень знань, т. К. Це ключові для верстальників інструменти. Найбільше часу варто приділити знайомству з Grid CSS, а також Flexbox;
7. Media Queries – використовується, щоб виконувати і перевіряти кроссплатформенну і кросбраузерну види верстки;
8. Шаблонизатор – щоб підставляти дані в динамічному режимі;
9. WordPress і Drupal, OpenCart і Joomla, MODx і Bitrix тощо. – поширені движки сайтів (CMS).

Розробка і програмування веб-додатків, в сучасну епоху інтернет-технологій, займає значне місце в розробці програмного забезпечення. В наш час, мережа Інтернет, завдяки новітнім технологіям, стала набагато доступнішою і потреба в локальних додатках значно впала. А отже зріс попит на такі професії, як Front-end та Back-end розробники. Якщо раніше, в епоху раннього Інтернету, вся робота веб-розробника полягала, виключно в розробці односторінкових веб-сайтів, то зараз технології пішли набагато вперед і сучасні веб-додатки, написані на мові JavaScript і її фреймворках, можуть скласти серйозну конкуренцію класичним локальним додаткам, написаним на таких мовах, як Java, C++, C# і т.д.

1.3 Постановка задачі

Даний дипломний проект, представляє собою автономну, інтерактивну онлайн-систему навчання по веб-розробці, а саме по одній з її частин front-end.

В наш час передових технологій, коли навіть працювати можна не виходячи з дому, дистанційне онлайн-навчання вже давно стало невід'ємною частиною освітнього процесу. Це чудовий спосіб швидко отримати потрібні знання та навички, при цьому зберігаючи соціальну дистанцію, що до вподоби багатьом людям, які відчують себе некомфортно в великих групах людей. До того ж, такі проекти дають змогу людям, які працюють, отримувати нові навички або навіть освоювати нову професію, залишаючи багато часу і для основної роботи. Тому в сучасному світі, є велика потреба саме в таких онлайн-школах. Одною із таких онлайн-шкіл і є представлена дипломна робота.

Проект буде, таким собі, онлайн-курсом, де кожен бажаючий, не залежно від віку, може навчитися всього, що стосується такої важливої частини веб-розробки, як front-end. Все, що потрібно від користувача даного проекту, це тільки терпіння і бажання вчитися й пізнавати нове.

Система надає вибір. Якщо користувач новачок в світі ІТ , і йому ніколи не доводилося працювати з веб-розробкою і програмуванням, як таким, він може прямо про це вказати при реєстрації, і тоді система запропонує йому повний курс по front-end. Але якщо користувач уже має деякі уявлення і просто хоче вдосконалити або вивчити певну тему, система запропонує йому каталог доступних тем і він зможе вибрати те, що йому дійсно потрібно, не витрачаючи час на проходження розділів front-end, які йому вже знайомі.

Якщо ж користувач, має деякий обсяг знань з фронтенду, але не може самостійно оцінити їх рівень, система дасть змогу пройти невеличкий тест, який буде включати в себе 12 запитань по front-end.

Всього існуватиме 3 рівні знань:

1. Початківець(1-4 правильні відповіді). Користувач майже не володіє ніякими знаннями. Пропонується почати повний курс з початку;

2. Учень(5-8 правильних відповідей). Користувач володіє певним рівнем знань, але він далеко не досконалий;
3. Майстер(9-12 правильних відповідей). Користувач має доволі великий обсяг знань з даної теми і скоріше за все, навіть, має досвід роботи в front-end.

Після вибору свого навчального плану, користувач, може починати вивчення тем. Спочатку пропонується ознайомитися з теорією, після кожного підрозділу теми, можна наочно побачити, як працює технологія фронтенду, про яку розповідається в уроці, це дає змогу зрозуміти, по якому саме принципу вона працює. Також, в кінці кожної теоретичної частини уроку, буде декілька посилань на розширену документацію з інших джерел, якщо користувач має бажання більш глибоко зрозуміти принцип роботи даної технології або якщо теоретична частина, представлена на уроці, з якихось причин здалась йому недостатньо зрозумілою або інформативною.

На вибір також дається мова, на якій буде викладено матеріал і інтерфейс додатку.

Всього запропоновано дві мови, англійська та українська. Англійська мова дуже тісно зв'язана за програмуванням, оскільки майже все програмне забезпечення і вся термінологія викладена англійською мовою.

Студент буде запропоновано змінити мову на самому початку, але якщо виникне потреба змінити мову ще раз, це можна буде зробити в меню користувача.

Після проходження всього навчального курсу, користувачу пропонується пройти фінальний тест, знову ж, виключно з власної волі, за допомогою якого, система ще раз оцінить рівень знань користувача і дасть йому змогу наочно побачити прогрес в його знаннях. Людина сама повинна мати бажання вчитися, в іншому випадку будь-які курси безсилі.

РОЗДІЛ 2

АНАЛІЗ ПРОГРАМНИХ АНАЛОГІВ

2.1 Опис платформи Coursera

Coursera[6] – це американський провайдер масових відкритих онлайн-курсів, заснований у 2012 році викладачами інформатики Стенфордського університету Ендрю Нг та Дафні Коллером, який пропонує масові відкриті онлайн-курси, спеціалізації, ступені, професійні та майстер-курси.



Рисунок 2.1 – Емблема онлайн-курсів Coursera

Coursera співпрацює з університетами та іншими організаціями, пропонуючи онлайн-курси, сертифікації та ступені з різних предметів.

Coursera була заснована з баченням забезпечити навчальний досвід, що змінює життя для студентів у всьому світі. Сьогодні Coursera – це глобальна платформа онлайн-навчання, яка пропонує будь-кому та де завгодно доступ до онлайн-курсів та ступенів провідних університетів та компаній.

Курси Coursera тривають приблизно 4-12 тижнів, при цьому проводяться від однієї до двох годин відеолекцій на тиждень. Ці курси пропонують вікторини, щотижневі вправи, оцінки та тести, а іноді і фінальний проект або іспит для завершення курсу. Курси також доступні за запитом, і в цьому випадку користувачі можуть пройти курс повільно з усіма доступними матеріалами одночасно. Станом на травень 2015 року Coursera запропонував 104 спеціальних курси, а також пропонує короткі керовані, 2-3-годинні проекти, які можна виконати. А це можна зробити вдома.

Станом на 2017 рік, Coursera пропонує повний ступінь магістра. Спочатку вони починали з магістра в галузі інновацій та підприємництва (OMIE) від HEC Paris та магістра бухгалтерського обліку (iMSA) з Університету Іллінойсу, але перейшли до магістра в галузі комп'ютерних наук та магістра ділового адміністрування (iMBA). Також як частина MBA, є деякі курси, які пропонуються окремо, і вони будуть включені в програму MBA під час реєстрації на курси цифрового маркетингу.

У 2020 році, у відповідь на пандемію COVID-19, Coursera розпочали курс в Інституті Jamil при Імперському коледжі Лондона. Курс під назвою "Наукові питання: поговоримо про COVID-19" був найпопулярнішим, він був запущений на Coursera у 2020 році, і в цьому році було зареєстровано понад 130 000 студентів. Курс був представлений керівником досліджень Інституту Jamil професором Хелен Уорд та заступником директора, доктором Катаріною Хок із спеціальними модулями, у співпраці з іншими дослідниками Imperial Empire.

У липні 2016 року компанія випустила корпоративний продукт під назвою Coursera для бізнесу. TechCrunch зазначає, що компанія "відкрилася для отримання додаткового доходу від прибуткового корпоративного ринку електронного навчання, який, як повідомляється, коштує лише 12 мільярдів доларів". Серед бізнес-клієнтів Coursera – L'Oréal, Boston Consulting Group та Axis Bank.

У жовтні 2016 року Coursera запустила щомісячну модель передплати на спеціалізацію разом із безкоштовною 1-тижневою пробною версією. Компанія заявила, що ціна підписки варіюватиметься "залежно від предметної області".

У березні 2018 року Coursera запустила шість повністю онлайн-курсів, включаючи ступінь бакалавра та магістра в різних галузях.

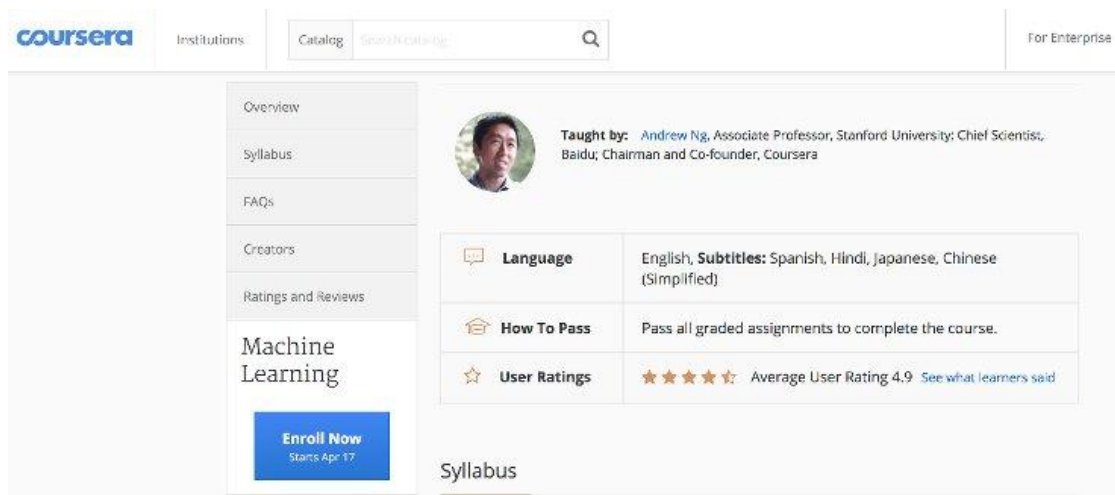


Рисунок 2.2 – Внутрішній інтерфейс Coursera

Coursera пропонує безліч курсів з самих різних напрямків. Одним з самих популярних і провідних напрямків, є Frontend developing. Курс дає можливість нуля, дізнатися, як впроваджувати сучасні веб-сторінки з HTML та CSS, дає розуміння, як кодувати сторінки таким чином, щоб його компоненти переставляли та змінювали розмір автоматично залежно від розміру екрана користувача. Ще одне, але, безумовно, не менш важливе, курс найпоширенішої, популярної та неймовірно потужної мови програмування, Javascript.

2.2 Опис відкритого онлайн-курсу Udeemy

UdeemyInc. - це американський відкритий онлайн-курс, призначений для дорослих та студентів. Він був заснований у травні 2010 року Ереном Балі, Гаганом Біані та Октаєм Кагларом.

Можна самостійно вибрати той напрямок, який користувач вважає потрібним, або просто відповісти на кілька швидких запитань, які включають в себе теми, які цікавлять студента, і запитання, направлені на те, щоб зрозуміти

наскільки він знайомий з конкретними предметами, щоб отримати персоналізовані рекомендації курсу.

Курси Udemy [7] проводяться за допомогою відео, аудіо чи тексту. Особливості курсу залежать від платформи, але можуть включати вікторини, субтитри, закладки, огляди та оголошення інструкторів.



Рисунок 2.3 – Емблема онлайн-курсів Udemy

Станом на лютий 2021 рік платформа налічує понад 40 мільйонів студентів, 155 000 курсів та 70 000 викладачів, які викладають курси на понад 65 мовах. Існує понад 480 мільйонів курсів. Студенти та викладачі із понад 180 країн, а дві третини студентів не з США.

Студенти відвідують курси здебільшого як засіб вдосконалення навичок, пов'язаних із роботою. Udemy доклали особливих зусиль для залучення корпоративних тренерів, які хочуть створити курси для працівників своєї компанії. Станом на 2021 рік веб-сайт має понад 155 000 курсів.

Популярні категорії курсів на Udemy включають:

1. Розвиток
2. Бізнес
3. IT та програмне забезпечення
4. Дизайн
5. Маркетинг

6. Особистісний розвиток

7. Фотографія

8. Музика

Штаб-квартира Udemy розташована в Сан-Франциско, штат Каліфорнія, офіси розташовані в Денвері, штат Колорадо; Дублін, Ірландія; Анкара, Туреччина; Сан-Паулу, Бразилія та Гуруграм, Індія.

Udemy – це платформа, яка дозволяє викладачам створювати онлайн-курси на бажані теми. За допомогою засобів розробки курсів Udemy вони можуть завантажувати відео, презентації PowerPoint, PDF-файли, аудіо, ZIP-файли в режимі реального часу для створення курсів. Вчителі також можуть взаємодіяти з користувачами через онлайн-форуми.

Курси пропонуються в широкому діапазоні категорій, включаючи бізнес та підприємництво, науку, мистецтво, здоров'я та фітнес, мову, музику та технології. Udemy також пропонує Udemy for Business, надаючи бізнесу доступ до цільового набору з понад 7000 навчальних курсів на теми, починаючи від тактики цифрового маркетингу і закінчуючи продуктивністю офісу, дизайном, управлінням, програмуванням тощо. Завдяки Udemy for Business організації також можуть створювати власні навчальні портали для корпоративного навчання.

Курси в Udemy можуть бути платними або безкоштовними, залежно від викладача. У 2015 році 10 найкращих вчителів отримали загальний дохід понад 17 мільйонів доларів.

У квітні 2013 року Udemy запропонував додаток для Apple iOS, який дозволяє студентам брати уроки безпосередньо з iPhone. Версія для Android була запущена в січні 2014 року. Станом на січень 2014 року додаток iOS було завантажено більше 1 мільйона разів, і 20 відсотків користувачів Udemy отримують доступ до своїх курсів через мобільний телефон. У липні 2016 року

Udemy розширив свою платформу iOS, включивши Apple TV. 11 січня 2020 року мобільний додаток Udemy став найпопулярнішим додатком для Android в Індії.

Udemy згадувався в The New York Times, The China Post, Fast Company, BBC, YPN та TechCrunch.

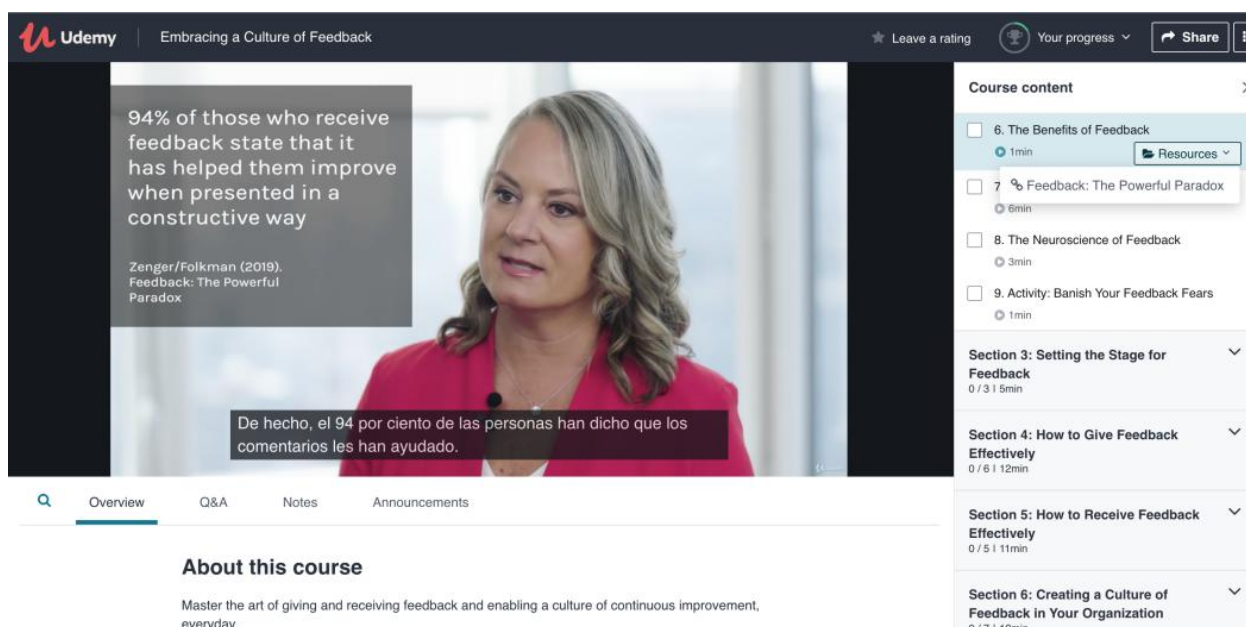


Рисунок 2.4 - Внутрішній інтерфейс Udemy

Це один з самих передових і популярних курсів, для вивчення веб-розробки – HTML, CSS, JavaScript, Node та багато іншого. Він зосереджується не лише на інтерфейсній розробці, а й на внутрішній системі та інших аспектах веб-розробки.

2.3 Опис веб-платформи для навчання freeCodeCamp

freeCodeCamp [8] – це некомерційна організація, яка складається з інтерактивної веб-платформи для навчання, форуму спільноти в Інтернеті, чатів, інтернет-публікацій та місцевих організацій, які мають намір робити веб-розробку для навчання доступно кожному. Починаючи з підручників, які знайомлять учнів з HTML, CSS та JavaScript, студенти переходять до проектних завдань, які вони виконують індивідуально або в парах.

Після виконання всіх проектних завдань студенти працюють з іншими некомерційними організаціями для створення веб-додатків, надаючи студентам практичний досвід розробки.

freeCodeCamp був запущений у жовтні 2014 року та є частиною FreeCodeCampInc. Засновник Квінсі Ларсон – розробник програмного забезпечення, який взявся за програмування в аспірантурі та створив freeCodeCamp як спосіб впорядкувати прогрес студентів від початківців до підготовлених.



Рисунок 2.5 – Емблема онлайн-курсів freeCodeCamp

Оригінальна навчальна програма була зосереджена на MongoDB, Express.js, AngularJS та Node.js, і, за оцінками, на її виконання потрібно 800 годин. Багато уроків були посиланнями на безкоштовні матеріали на інших платформах, таких як Codecademy, Stanford або Code School. Курс був розділений на "Waypoints" (швидкі, інтерактивні навчальні посібники), "Bonfires" (виклики алгоритмів), "Ziplines" (проекти інтерфейсу) та "Basejumps" (проекти повного стеку). Після завершення фронт-енд проектів та повного стеку студент отримував відповідні сертифікати.

Навчальна програма була оновлена в січні 2016 року, щоб менше покладатися на зовнішні матеріали, видаляти нетрадиційні назви розділів та

переключати фокус з AngularJS на React.js, як обрану бібліотеку інтерфейсу. До роботи курсу було внесено ряд доповнень, включаючи D3.js та Sass, які збільшили загальний орієнтовний час до 2080 годин та додали ще два сертифікати, візуалізацію даних та back-end.

Самостійний навчальний план передбачає 1400 годин інтерактивних завдань, кодування та проектів веб-розробки, плюс 800 годин участі у проектах з відкритим кодом для некомерційних організацій, і постійно розширюється новими завданнями та проектами. Це означає приблизно один рік кодування. Навчальна програма розділена на адаптивний веб-дизайн, алгоритми та структури даних JavaScript, бібліотеки інтерфейсів, візуалізацію даних, API та мікросервіси, а також інформаційну безпеку та забезпечення якості. Учасники отримують сертифікат після заповнення кожного розділу.

Навчальний план наголошує на парному програмуванні, призначеному для виховання культури співпраці та спільного навчання, за допомогою якого, можна подолати сумніви студента в практичності своїх навичок.

Мови та технології, які зараз викладає freeCodeCamp, включають HTML5, CSS3, JavaScript, jQuery, Bootstrap, Sass, React.js, Node.js, Python, Express.js, MongoDB та Git.

Коли студенти freeCodeCamp отримують усі сертифікати навчальної програми, вони отримують можливість, працювати з некомерційними організаціями. Прикладами є некомерційна компанія Корепнік та People Saving Animals.

У 2016 році freeCodeCamp оголосив про свою ініціативу Mail for Good, яка поширює та відкриває джерела їхньої некомерційної діяльності на всі

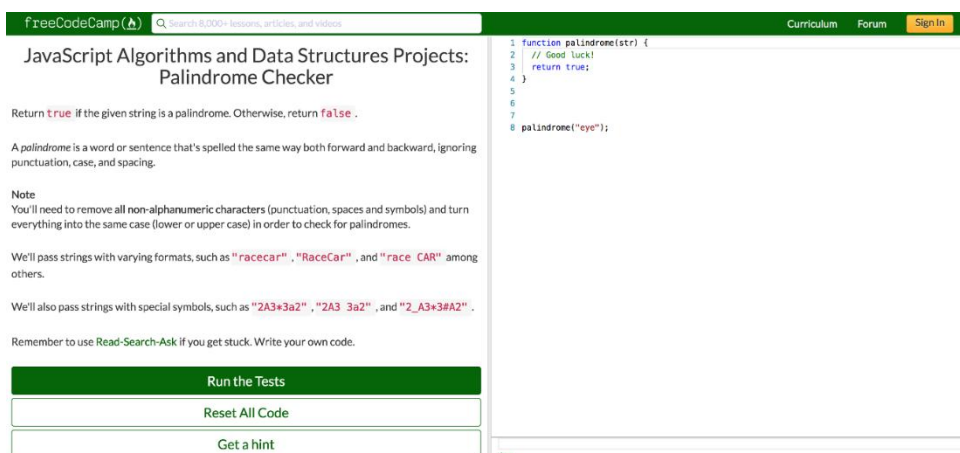


Рисунок 2.6 – Внутрішній інтерфейс freeCodeCamp

некомерційні організації. Протягом десяти місяців після запуску ініціатива створила сім інструментів з відкритим кодом. Mail for Good – це один із проектів, який допомагає організаціям надсилати масові повідомлення електронною поштою за низькою ціною, що є дешевшою альтернативою таким послугам, як MailChimp.

Платформа freeCodeCamp має приблизно 350 000 відвідувачів на місяць, зі студентами з більш ніж 160 країн.

freeCodeCamp має міжнародні спільноти, де студенти можуть взаємодіяти особисто. Деякі групи були представлені в місцевих новинах, посилаючись на freeCodeCamp, як на вступ до програмування з перспективою на передбачувані вакансії на робочих місцях, пов'язаних з програмуванням у наступному десятилітті.

Надто є можливість навчатися в дорозі, слухаючи підкаст freeCodeCamp або дивлячись на YouTube канал freeCodeCamp. Також є мобільний додаток, розроблений спеціально для навчання програмуванню – Grasshopper. Він безкоштовний і розроблений автором програми freeCodeCamp та її командою. Доступний на iOS та Android.

freeCodeCamp – це приголомшливий ресурс для тих, хто навчається самостійно. Він збирає часто розпорошену інформацію про програмування в чіткі та корисні уроки. Якщо ви самостійно навчаєтесь, freeCodeCamp– чудове місце для початку навчання.

2.4Опис онлайн-колекції завдань Edabit

Edabit [9] – це онлайн-колекція завдань для 8 мов програмування, включаючи C ++, JavaScript та Python. Вони надають інженерам програмного забезпечення невеликі завдання, які допомагають швидко розвивати свої навички. На рівні від найпростішого до експертного, Edabit має на вибір понад 10 000 завдань кодування. Користувачі створюють профіль та беруть участь у іграх, щоб розблокувати досягнення та підвищити рівень. Заробляючи бали за правильні рішення, користувачі Edabit можуть відстежувати навички, якими вони оволоділи.

Кожен виклик включає посилання на ресурси, рішення та коментарі інших кодерів. Користувачі також можуть створювати власні колекції, які об'єднують різні завдання з програмування. Ті, хто зосереджується на розвитку певної навички, можуть ознайомитись із вже існуючими колекціями, що охоплюють подібні теми. Edabit, є відкритим кодом, тому всі бажаючі можуть також додати нові задачі до бази завдань.

Інженери-програмісти, які прагнуть розвинути нові навички в алгоритмах, логіці, структурах даних тощо, можуть отримати вигоду від завдань на Edabit. Також можна використовувати платформу для вивчення таких мов програмування, як Ruby та PHP.



Рисунок 2.7 – Емблема онлайн-курсів Edabit

Edabit дозволяє користувачам фільтрувати завдання за такими темами, як масиви та цикли даних, що може бути хорошим варіантом для тих, хто хоче розглянути конкретні навички перед своїми технічними співбесідами. Оскільки у них є питання різного рівня кваліфікації, люди, які не мають досвіду програмної інженерії, також можуть практикувати основи на Edabit.

Edabit можна використовувати безкоштовно. Щоб зареєструватися, люди повинні створити профіль, який вони можуть зробити за допомогою своїх облікових записів Google або Facebook.

З огляду на нескінченну пропозицію місць для вивчення та практикування задач по JavaScript, може бути важко знайти правильне, яке відповідає особистому стилю навчання. Edabit – це простий сайт з безліччю задач по JavaScript, які можна практикувати щодня.

Існують сотні уроків по JavaScript, які дозволяють вирішити задачі будь-якої складності – від дуже простих, аж до експертних. Простий інтерфейс чудово справляється з тим, щоб завжди триматися на шляху, дозволяючи зосередитись виключно на завданнях. Існує календар, який відстежує, скільки днів поспіль користувач програмував, і можна перейти на рівень залежно від кількості очок, які було зароблено.

Двома найпотужнішими функціями для навчання на Edabit.com є вкладка "Ресурси" та можливість бачити рішення інших користувачів після того, як було подано власне рішення. Якщо натиснути вкладку "ресурси", з'явиться список посилань, які можуть бути корисними для вирішення цієї конкретної задачі. Особливо для початківців ця функція може перевіряти завдання та намагатися вирішити проблему самостійно з Google, коли користувач застряг на якійсь задачі.

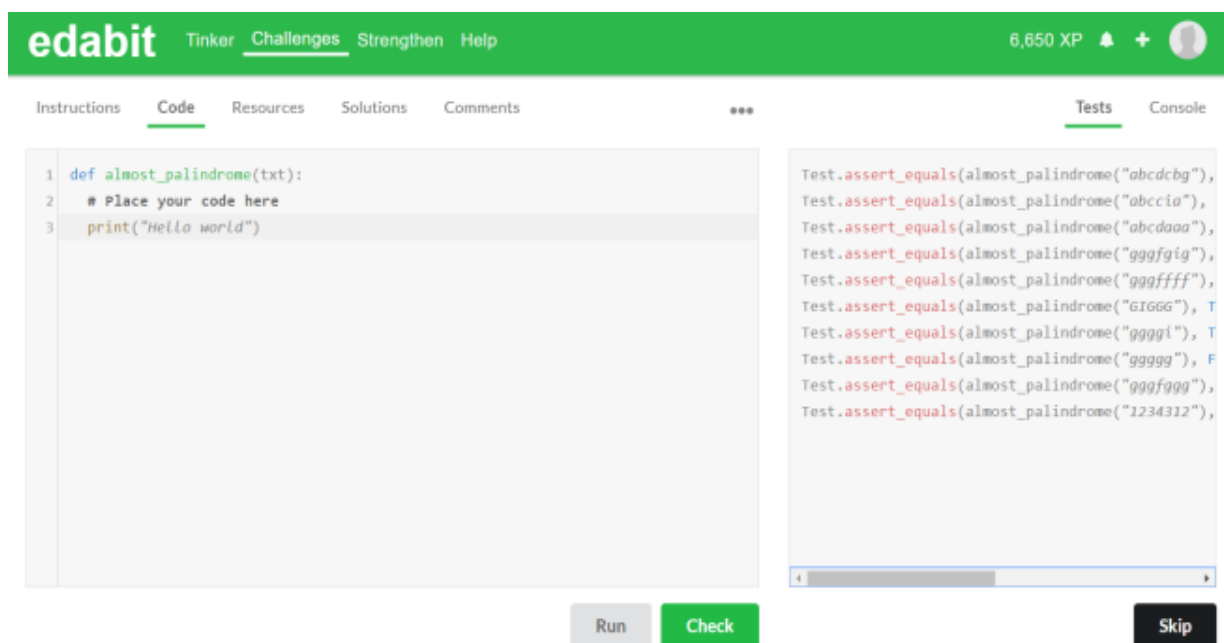


Рисунок 2.8 – Внутрішній інтерфейс Edabit

Завжди приємно вирішити проблему самостійно, але майже завжди є кращий спосіб її вирішення.

РОЗДІЛ 3

ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ АДАПТИВНОГО НАВЧАННЯ ПО НАПРЯМУ FRONT-END

3.1 Опис середовища розробки

Для створення дипломного проекту було використано засіб для створення та редагування сучасних веб-застосунків і програм Visual Studio Code, версія 1.53.0, веб-переглядач Mozilla Firefox, версія 84.0.2 та операційна система Ubuntu, версія 20.04 на базі Linux.

VisualStudioCode

Visual Studio Code – це безкоштовний редактор вихідного коду, розроблений для Microsoft Windows, Linux та MacOS. Особливості включають підтримку налагодження, роз'яснення синтаксису, інтелектуальне заповнення, оновлення коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, налаштування та встановлювати розширення з додатковими функціями.

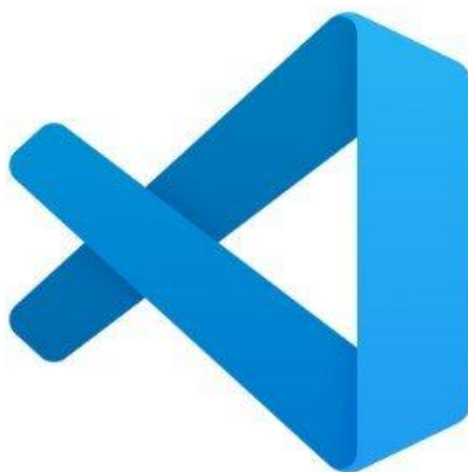


Рисунок 3.1 – Емблема Visual Studio Code

Корпорація Майкрософт випустила вихідний код Visual Studio Code за ліцензією MIT, який знаходиться у сховищі microsoft / vscode на GitHub, але зібрані версії безкоштовні.

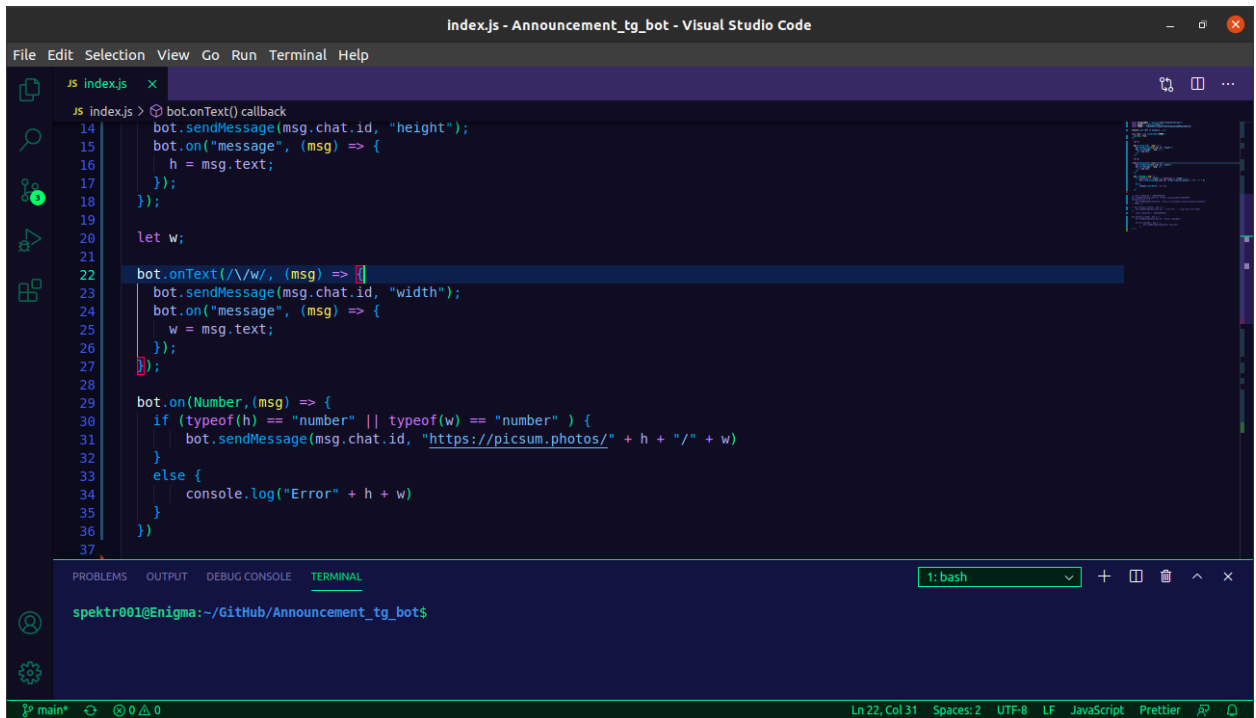


Рисунок 3.2 – Зовнішній вигляд Visual Studio Code 1.53.0

Згідно з опитуванням розробників Stack Overflow 2019, Visual Studio Code був названий найпопулярнішим інструментом IDE, і його використовували 50,7% з 87 317 респондентів. Visual Studio Code був представлений 29 квітня 2015 року на конференції Microsoft Build 2015, а версія для попереднього перегляду була випущена незабаром після цього.

Visual Studio Code був випущений за ліцензією MIT 18 листопада 2015 року, а його вихідний код – на GitHub. Також оголошено про підтримку розширення. 14 квітня 2016 р. Visual Studio Code був випущений в Інтернет для загального огляду.

Visual Studio Code поєднує простоту редактора вихідного коду з потужними інструментами розробника, такими як IntelliSense, інструмент для заповнення та налагодження коду.

Перш за все, це редактор, який не заважає. Чудовий цикл редагування-побудова-налагодження, дає змогу менше звертати увагу на дрібниці при розробці, і дає більше часу на реалізацію персональних ідей.

Visual Studio Code, в основі своїй, має блискавичний редактор вихідного коду, ідеально підходить для повсякденного використання. Оскільки VS Code підтримує сотні мов, він допомагає миттєво збільшити продуктивність, виділяючи синтаксис, узгоджуючи дужки, автоматичні розриви абзаців, вибір кадру та часткове редагування. Інтуїтивні комбінації клавіш, простота налаштування та загальнодоступних комбінацій клавіш полегшують керування кодом.

Для серйозного кодування часто використовуються інструменти, які більше розуміють код, ніж текстові блоки. Visual Studio Code включає підтримку завершення коду IntelliSense, розуміння вдосконаленого семантичного коду, навігації та рефакторингу коду.

Чим складніше кодування, тим складніше його налагодити. Visual Studio Code включає інтерактивний налагоджувач, тому можна переглядати вихідний код, перевіряти змінні, переглядати стеки викликів та виконувати команди на консолі.

Visual Studio Code інтегрується з інструментами збірки та сценаріями, щоб прискорити щоденні робочі процеси та виконувати загальні завдання. VS Code має підтримку Git, тому ви можете переглядати очікувані зміни, не виходячи з редактора, включаючи перегляд змін, що очікують на розгляд.

Visual Studio Code дає можливість налаштувати кожну функцію на свій смак і встановити будь-яку кількість сторонніх розширень. VS Code – це проект з відкритим кодом, тому кожен бажаючий може зробити свій внесок у зростаючу та жваву спільноту на GitHub.

VS Code включає збагачену вбудовану підтримку розробки Node.js за допомогою JavaScript і TypeScript, що базується на тих самих базових технологіях, що керують Visual Studio. VS Code також включає чудові інструменти для веб-технологій, такі як JSX / React, HTML, CSS, SCSS, Less та JSON.

Архітектурно, Visual Studio Code, є найкращим поєднанням мовних та веб-технологій. Використовуючи Electron, VS Code поєднує в собі веб-технології, такі як JavaScript та Node.js, зі швидкістю та гнучкістю власних додатків. Visual Studio

Code використовує нову, швидшу версію того самого виробничого редактора на основі HTML, який підтримує хмарний редактор Monaco, Internet Explorer F12 Tools та інші проекти. Крім того, Visual Studio Code використовує архітектуру служби інструментів для інтеграції з багатьма технологіями, що працюють на Visual Studio, включаючи Roslyn для .NET, TypeScript, механізм налагодження Visual Studio тощо.

VS Code включає універсальний шаблон розширення, який дозволяє розробникам створювати і використовувати додатки та змінювати власний досвід редагування-побудови-налагодження.

Visual Studio Code в основному є редактором коду. Як і багато інших редакторів, VS code має загальний інтерфейс користувача та макет дослідника ліворуч, відкриває всі доступні файли, папки редактор праворуч, показуючи вміст файлів, які ви відкрили .

Visual Studio Code має простий і зрозумілий макет, який максимізує доступний для редактора простір, залишаючи достатньо місця для перегляду та доступу до всього контексту вашої папки або проекту. Інтерфейс користувача розділений на п'ять розділів:

1. Редактор – основна частина редагування файлів. Він дає змогу відкрити будь-яку кількість редакторів вертикально або горизонтально.
2. Бічна панель – містить різноманітні подання, такі як провідник, які допоможуть працювати над проектом.
3. Рядок стану – інформація про відкриті проекти та файли, які редагуються.
4. Панель активності – розташована в лівій частині екрану, вона дозволяє перемикатися між переглядачами та надає додаткові метрики, пов'язані з контекстом, наприклад, кількість змін, коли ввімкнено Git.
5. Панелі – дають змогу відображати або налагоджувати різні панелі під областю редактора, інформацію, помилки, попередження або вбудовані термінали. Можна перемістити панель праворуч, щоб отримати вертикальний простір.

Кожного разу, при запуску Visual Studio Code, він відкривається в тому ж стані, в якому він був, коли востаннє його закривали. Папка, макет та відкриті файли зберігаються.

Мінімапа (контур коду) дає високоякісний огляд вихідного коду, що корисно для швидкої навігації та розуміння коду. Мінімапа файлу відображається праворуч від редактора. Можна натиснути або перетягнути затінену область, щоб швидко перейти до різних розділів файлу.

Visual Studio Code відображає заголовки з вкладками в розділі заголовків редактора. Вкладки дозволяють швидко перемикатись між елементами, і дають можливість перетягувати вкладки, щоб змінити їх порядок.

Якщо в рядку є більше відкритих елементів, ніж може поміститися в області заголовка, можна скористатися командою «Показати відкриті редактори».

Таким чином, можна сміливо сказати, що Visual Studio Code є одним з найбільш оптимальних редакторів коду. Без сумнівів, є й інші редактори коду, зі схожим функціоналом, можливостями, але саме VS Code займає позицію самого передового і швидкісного редактору зі списку редакторів.

Mozilla Firefox

Mozilla Firefox або просто Firefox – це безкоштовний веб-браузер із відкритим кодом, розроблений Mozilla Foundation та його дочірньою компанією Mozilla Corporation. Firefox використовує механізм компонування Gecko для візуалізації веб-сторінок, що реалізує поточні та очікувані веб-стандарти. У 2017 році Firefox почав впроваджувати нову технологію під кодовою назвою Quantum для сприяння паралелізму та більш інтуїтивно зрозумілому інтерфейсу користувача. Firefox офіційно доступний для Windows 7 або новішої версії, macOS та Linux. Його неофіційні порти доступні для різних Unix та Unix-подібних операційних систем, включаючи FreeBSD, OpenBSD, NetBSD, illumos, та Solaris Unix. Firefox також доступний для Android та iOS. Однак версія iOS використовує механізм компонування WebKit замість Gecko через вимоги до платформи, як і у всіх інших веб-браузерах iOS. Оптимізована версія Firefox

також доступна на Amazon Fire TV, як один з двох основних браузерів, доступних у базовому браузері Amazon.

Firefox був створений у 2002 році під кодовою назвою "Phoenix" членами спільноти Mozilla, які бажали користуватися окремим браузером, а не пакетом програм Mozilla Application Suite. Протягом своєї бета-фази Firefox виявився популярним серед своїх тестувальників і отримав високу оцінку за його швидкість, безпеку та доповнення в порівнянні з домінуючим на той час Internet Explorer 6. Firefox був випущений 9 листопада 2004 р. Firefox є духовним наступником Netscape Navigator, оскільки спільнота Mozilla була створена Netscape в 1998 році до їх придбання AOL.



Рисунок 3.3 – Емблема Mozilla Firefox

Firefox включає більшість функцій, з якими знайомі користувачі інших браузерів. Перша ітерація браузера також включала кілька нових функцій, яких інші браузери не мали на той час, наприклад, панель інструментів закладок та перегляд з вкладками, які дозволяють користувачеві швидко переключатися між кількома веб-сайтами. З тих пір інші браузери перейняли багато з цих функцій, хоча значна база розробників Firefox продовжує створювати нові плагіни, що пропонують підвищену зручність та функціональність. Оскільки елементи

керування сценаріями, такі як Java та ActiveX, можуть бути легко скасовані під час встановлення, існує ймовірність кращого захисту.

Функції включають перегляд з вкладками, перевірку правопису, покроковий пошук, додавання закладок в реальному часі, розумні закладки, менеджер завантажень, приватний перегляд, перегляд з урахуванням місцезнаходження (також відомий як "геолокація") на основі служби Google, та інтегровану систему пошуку, який використовує Google за замовчуванням. Крім того, Firefox забезпечує середовище для веб-розробників, в якому вони можуть використовувати вбудовані інструменти, такі як консоль помилок або інспектор DOM, а також розширення, такі як Firebug, і нещодавно випущена функція інтеграції з Rocket. Firefox Hello – це реалізація WebRTC, додана в жовтні 2014 року, яка дозволяє користувачам Firefox та інших сумісних систем здійснювати відеодзвінки з додатковою функцією спільного використання екрана та файлів, відправляючи посилання один одному. Видалення Firefox Hello планувалося у вересні 2016 року.

Функції можна додавати за допомогою доповнень, створених сторонніми розробниками. Додатки в основному кодуються за допомогою HTML, CSS, JavaScript, з API, відомим як WebExtensions, який призначений для сумісності з системами розширень Google Chrome і Microsoft Edge. Раніше Firefox підтримував доповнення за допомогою API XUL та XPCOM, що дозволяло їм безпосередньо

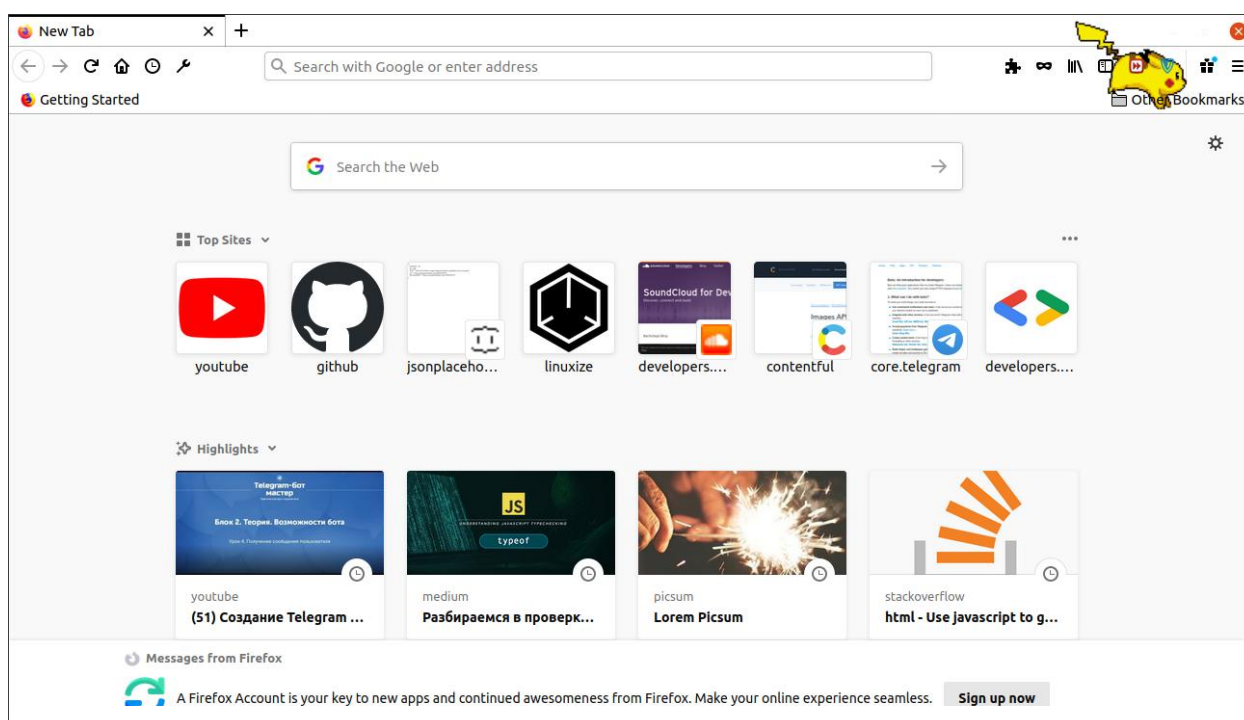


Рисунок 3.4 – Зовнішній вигляд Mozilla Firefox 84.0.2

отримувати доступ до більшої частини внутрішньої функціональності браузера та маніпулювати ними. Оскільки сумісність не була включена в багатопроцесорну архітектуру, надбудови XUL вважалися застарілими доповненнями і більше не підтримуються у Firefox 57 та новіших версіях.

Firefox реалізує багато веб-стандартів, включаючи HTML5, XML, XHTML, MathML, SVG 2 (частково), CSS (з розширеннями), ECMAScript (JavaScript), DOM, XSLT, зображення XPath та APNG (анімований PNG) з альфа-прозорістю. Firefox також реалізує пропозиції щодо стандартів, створені WHATWG, такі як зберігання на стороні клієнта, та елемент canvas. Ці стандарти реалізовані за допомогою механізму компонування Gecko та механізму JavaScript SpiderMonkey. Firefox 4 був першим найбільшим випуском для підтримки HTML5 та CSS3.

Ubuntu

Ubuntu – це дистрибутив Linux, заснований на Debian, який складається переважно з вільного програмного забезпечення з відкритим кодом. Ubuntu офіційно випущений у трьох виданнях: Desktop, Server, і Core. Усі випуски можуть працювати на комп'ютері як окремо, так і на віртуальній машині. Ubuntu – це популярна операційна система для хмарних обчислень із підтримкою OpenStack. Типовим робочим столом Ubuntu є GNOME, починаючи з версії 17.10. Перший офіційний випуск Ubuntu – версія 4.10 під кодовою назвою "WartyWarthog" – був запущений у жовтні 2004 року та викликав значний світовий інтерес, коли тисячі ентузіастів та експертів вільного програмного забезпечення приєдналися до спільноти Ubuntu.

Десктопна версія Ubuntu на сьогоднішній день є найпоширенішою у світі платформою робочих станцій Linux, що забезпечує роботу інженерів по всьому світу. Ubuntu Core встановлює стандарт для крихітних транзакційних операційних систем та для високозахисених підключених пристроїв. Ubuntu Server є довідковою операційною системою для проекту OpenStack і надзвичайно популярною гостьовою ОС на AWS, Azure та GoogleCloud. Ubuntu попередньо інстальовано на комп'ютерах від Dell, HP, Asus, Lenovo та інших світових

постачальників.

Ubuntu випускається кожні шість місяців, з довгостроковою підтримкою (LTS) кожні два роки. Станом на 22 жовтня 2020 року останнім випуском довгострокової підтримки є 20.04 ("Focal Fossa"), який підтримується до 2025 року як загальна підтримка та до 2030 року як платний варіант. Останній стандартний випуск – 20.10 ("Groovy Gorilla"), який підтримується дев'ять місяців.

Ubuntu названо на честь філософії ubuntu народів Нгуні, яка, за словами Canonical, означає "людяність до інших" з конотацією "Я такий, яким я є завдяки тому, хто ми всі".



Рисунок 3.5 – Емблема Ubuntu

Інсталяція Ubuntu за замовчуванням містить широкий спектр програмного забезпечення, що включає LibreOffice, Firefox, Thunderbird, Transmission та кілька легких ігор, таких як судоку та шахи. Багато додаткових програмних пакетів доступні за допомогою вбудованого програмного забезпечення Ubuntu (раніше Ubuntu Software Center), а також будь-яких інших інструментів управління пакетами на основі APT. Багато додаткових програмних пакетів, які більше не встановлюються за замовчуванням, таких як Evolution, GIMP, Pidgin та Synaptic, як і раніше доступні у сховищах та встановлюються основним інструментом або будь-яким іншим інструментом управління пакетами на основі APT. Також

доступні пакети оснащення для перехресного розподілу та flatpaks, які дозволяють встановлювати програмне забезпечення, таке як деяке програмне забезпечення корпорації Майкрософт, у більшості основних операційних систем Linux (таких як будь-яка підтримувана в даний час версія Ubuntu та Fedora). Файловим менеджером за замовчуванням є GNOME Files, який раніше називався Nautilus.

Все прикладне програмне забезпечення, встановлене за замовчуванням, є безкоштовним. Крім того, Ubuntu перерозподіляє деякі драйвери обладнання, які доступні лише у двійковому форматі, але такі пакети чітко позначені в обмеженому компоненті.

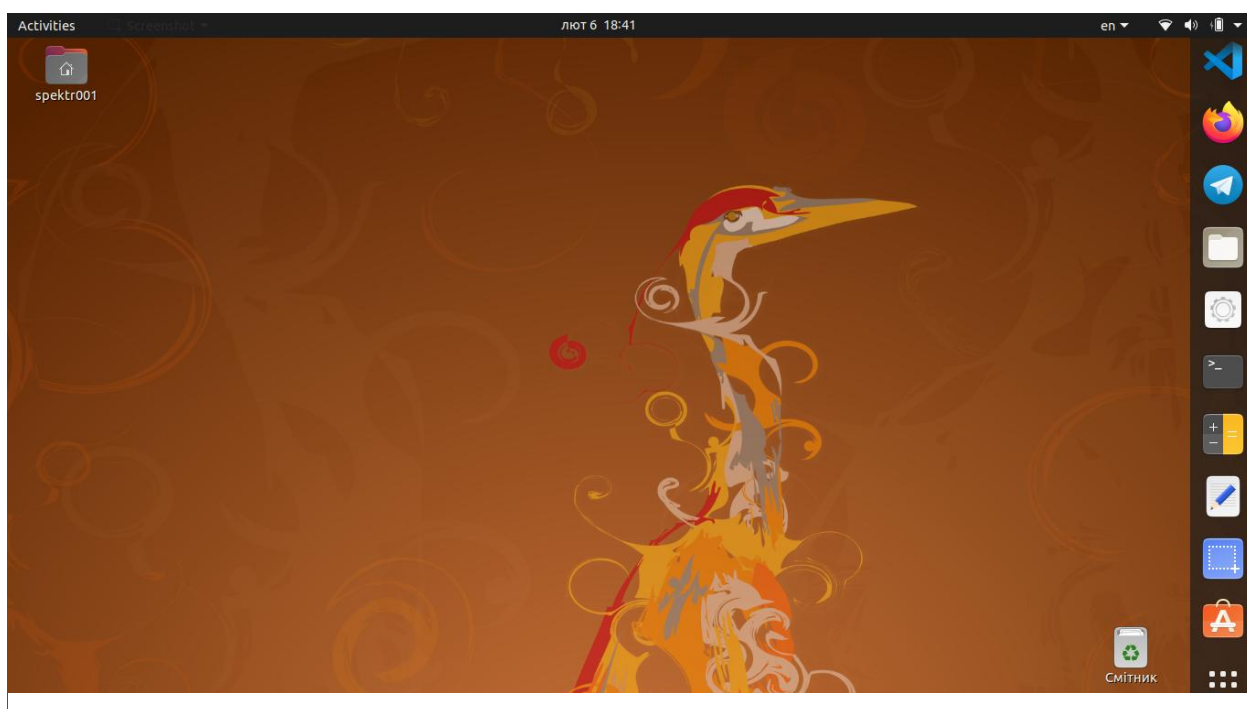


Рисунок 3.6 – Зовнішній вигляд Ubuntu 20.04

Користувацькі програми працюють із низькими привілеями і не можуть пошкодити операційну систему або файли інших користувачів. Для підвищення безпеки інструмент `sudo` використовується для призначення тимчасових привілеїв на виконання адміністративних завдань, що дозволяє кореневому обліковому запису залишатися заблокованим та допомагає запобігти випадковому недосвідченому внесенню катастрофічних системних змін або відкриттю дірок у безпеці.

Більшість мережевих портів за замовчуванням закриті для запобігання злому. Вбудований брандмауер дозволяє кінцевим користувачам, які встановлюють мережеві сервери, контролювати доступ. Для його налаштування доступний графічний інтерфейс (графічний інтерфейс для нескладного брандмауера). Ubuntu компілює свої пакети, використовуючи такі функції GCC, як PIE та захист від переповнення буфера, щоб зміцнити своє програмне забезпечення. Ці додаткові функції значно підвищують рівень безпеки за рахунок продуктивності 1% у 32-розрядної та 0,01% у 64-розрядної версіях.

Ubuntu також підтримує повне шифрування диска, а також шифрування домашнього та приватного каталогів.

Системні вимоги різняться залежно від продуктів Ubuntu. Для настільного випуску Ubuntu 20.04 LTS рекомендується ПК із принаймні 2 ГГцдвоядерним процесором, 4 ГБ оперативної пам'яті та 25 ГБ вільного місця на диску. Для менш потужних комп'ютерів існують інші дистрибутиви Ubuntu, такі як Lubuntu та Xubuntu. Ubuntu підтримує архітектуру ARM. Він також доступний на Power ISA.

Живий образ – типовий спосіб для користувачів оцінити та згодом встановити Ubuntu. Їх можна завантажити у вигляді образу диска (.iso), а потім записати на DVD і завантажити. Інші методи включають запуск оновлюваної версії через UNetbootin або Startup Disk Creator (попередньо встановлений інструмент в Ubuntu, доступний на машинах, на яких вже запущена ОС) безпосередньо з USB-накопичувача (створення відповідно живого DVD або живого USB-носія). Запуск Ubuntu таким чином відбувається повільніше, ніж запуск з жорсткого диска, але не змінює роботу комп'ютера, якщо користувач не дасть спеціальної вказівки. Якщо користувач вирішив завантажити живий образ, а не запустити інсталятор під час завантаження, все ще існує можливість використовувати інсталятор під назвою Ubiquity для встановлення Ubuntu після завантаження в реальне середовище. Образи дисків усіх поточних та попередніх версій доступні для завантаження на веб-сайті Ubuntu. Для створення спеціальних копій DVD-дисків Ubuntu Live (або компакт-дисків) доступні різні сторонні

програми, такі як Reconstructor. Доступні "мінімальні компакт-диски" (для використання на сервері), які вміщуються на компакт-диску.

Крім того, встановлення флеш-накопичувачів USB можна використовувати для завантаження Ubuntu та Kubuntu таким чином, що дозволяє постійно зберігати налаштування користувача та переносимість встановленої USB-системи між фізичними машинами (однак, BIOS комп'ютерів повинен підтримувати завантаження з USB). У новіших версіях UbuntuLive USB, можна використовувати для встановлення Ubuntu на USB-накопичувач (з CD-диском або DVD-диском у реальному часі або без нього).

Десктопне видання також можна встановити за допомогою образу Netboot (він же netboottarball), який використовує debian-інсталятор і дозволяє певні спеціалізовані налаштування Ubuntu: налаштування автоматизованих розгортань, оновлення зі старих налаштувань без доступу до мережі, розділення LVM або RAID, встановлення на системи з менш ніж 256 МБ оперативної пам'яті (хоча системи з низьким обсягом пам'яті можуть не мати можливості розумно запустити повне робоче середовище).

3.2 Опис додатку «FrontEndBox»

Додаток «FrontEndBox» – це не просто навчальна веб-платформа, яка навчає всіх бажаючих основам Front-end розробки. Основна ідея даного додатку, заключається саме в упорядкуванні та відборі знань з усієї мережі Інтернет і їх правильна подача для всіх хто бажає вивчити всі тонкощі веб-розробки, але навіть не має уяви з чого почати.

В сучасному світі, в Інтернеті є багато відкритих джерел, де можна без будь-яких зусиль отримати знання майже в усіх відомих галузях та покращити свої теоретичні пізнання майже з будь-якої професії. Але основна проблема заключається в тому, що є хибні, застарілі або неправильно складені статті, уроки, відео, які можуть не допомогти, а зашкодити новачку. Основна задумка цього проекту, саме і заключається в тому, щоб послужити початківцям в галузі Frontend, таким собі «компасом», відфільтрувати усю не потрібну інформацію та

залишити тільки те, що дійсно принесе користь новачкам, а не те, що просто забере час і нічому не навчить або більше того надасть хибні знання які вже не актуальні.

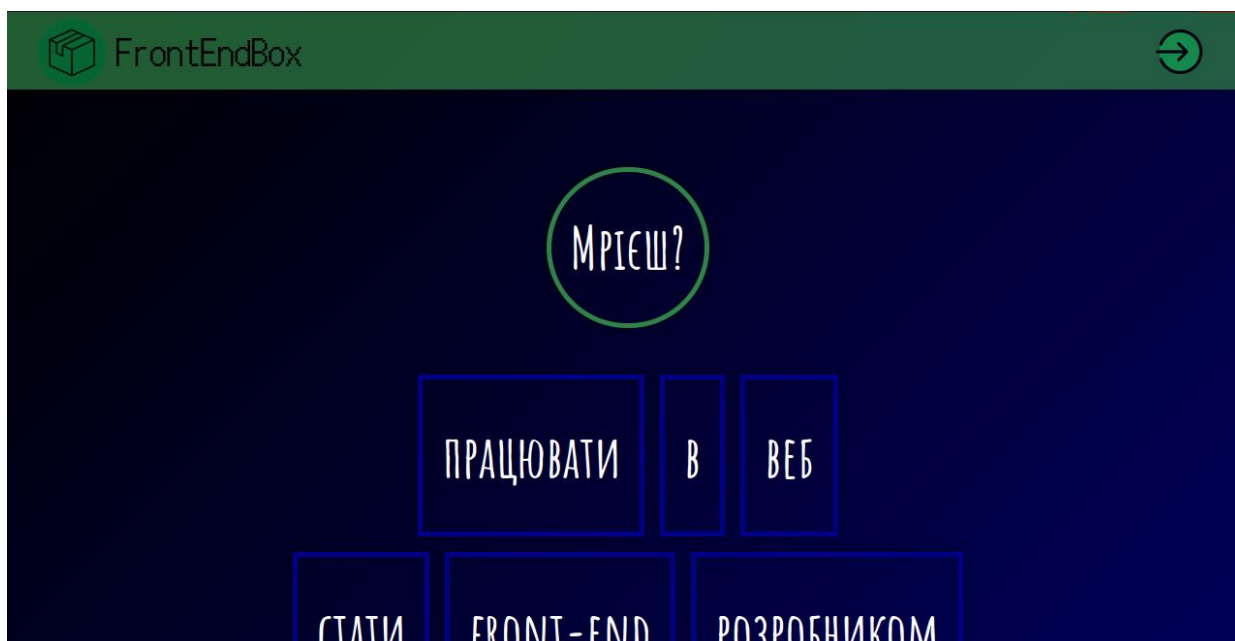


Рисунок 3.7 – Зовнішній інтерфейс додатку FrontEndbox. Сайт-візитівка

Перше, що повинен пройти користувач, це невелика рекламна веб-сторінка додатку, де коротко буде прорекламовано додаток. Після ознайомлення з сайтом, у користувача є змога зареєструватися, але це не зовсім та реєстрація в класичному розумінні, все, що потребується від користувача, це тільки ім'я. Використовувати захист паролем не має абсолютно ніякого сенсу, оскільки додаток не зберігатиме в собі абсолютно ніякої особистої інформації. Тому намагатися зламати подібний додаток, теж немає ніякого сенсу.

В першій частині додатку, користувачу пропонується вибрати, якою саме мову він хотів би бачити матеріал в даному додатку. На вибір є українська і англійська мови. Вся індустрія веб-розробки і програмування в цілому побудоване саме на знанні англійської мови, оскільки майже весь користувацький інтерфейс програм для розробки і написання коду, документація написані саме на англійській мові, тому користувач додатку, за бажанням зможе

паралельно з вивченням основ front-end розробки, підвищити свої знання англійської мови. Якщо ж користувач непевний в своїх знаннях саме англійської мови, тоді є змога вибрати українську мову.

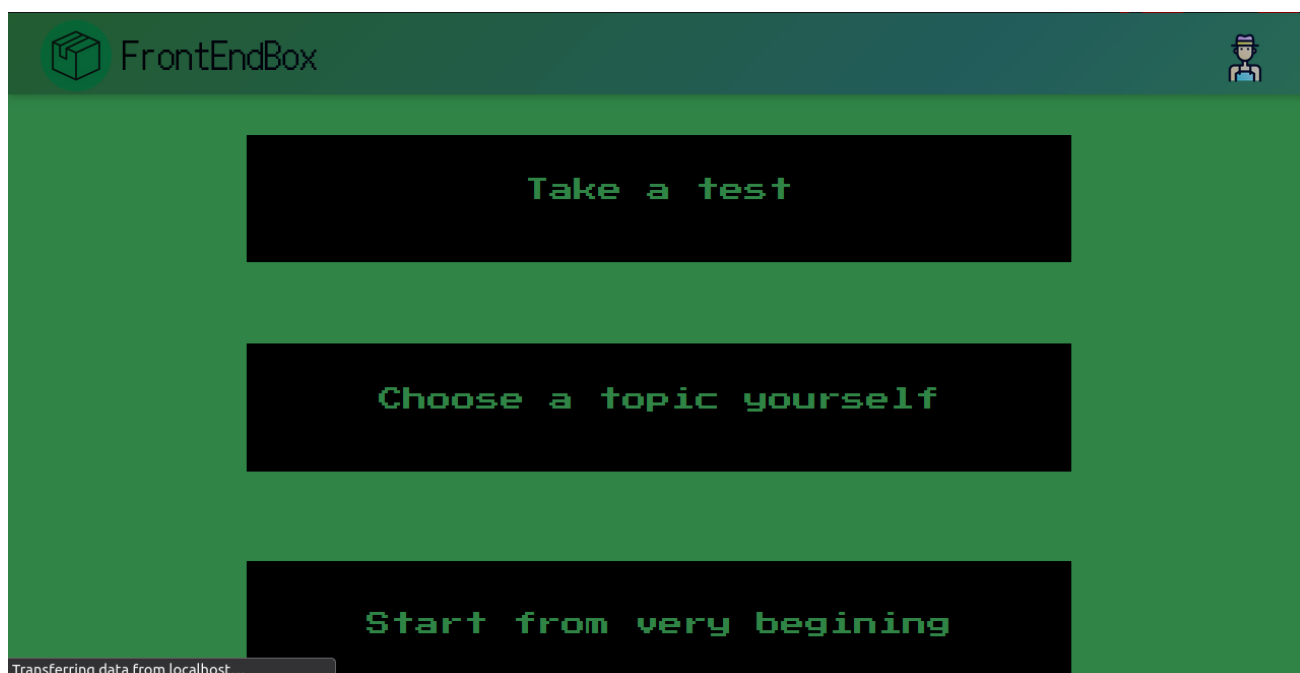


Рисунок 3.8 – Внутрішній інтерфейс додатку FrontEndbox. Сторінка вибору

Другим етапом, є вибір шляху навчання.

На вибір 3 способи:

1. Перший – пройти тестування, для того, щоб система змогла оцінити рівень знань студента для того, щоб було легше зрозуміти, на якому зараз рівні знаходиться користувач.
2. Другий спосіб – вибрати самостійно. У разі, якщо користувач вибирає саме цей пункт, з'являється бокове меню, в якому вказані всі теми та уроки і користувач самостійно обирає те, що вважає за потрібне.
3. Третій спосіб – почати з самого початку. Після вибору цього способу, відкривається початкова сторінка і користувач починає навчання.

Після всіх етапів вибору способу, студент може починати повноцінне навчання, проходячи урок за уроком в заданому порядку.

Стосовно саме питання взаємодії користувача з системою, у верхній частині додатку розміщено кнопку, при натисканні на яку випадає користувацьке меню, за допомогою якого, студент зможе змінити свій аватар, змінити тему додатку на темну або на світлу, змінити своє ім'я або ж змінити мову додатку.

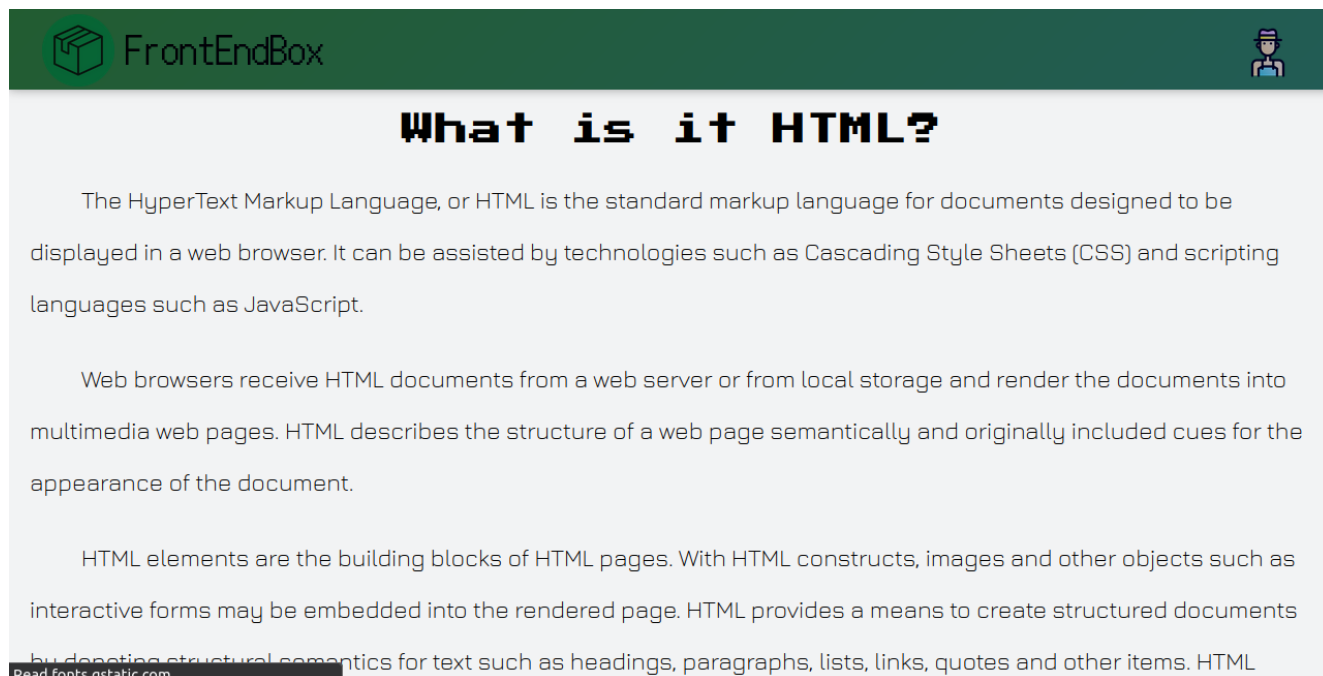


Рисунок 3.9 – Внутрішній інтерфейс додатку FrontEndbox. Сторінка уроку

Окрім навчальних матеріалів, в інтерфейс додатку, додано кілька речей, які зможуть прикрасити сам процес навчання. Одне із таких, це система досягнень. Кожного разу, коли студент доходить до певного етапу, або виконує якесь завдання, випадає невелике віконце з повідомленням про те, що було відкрито нове досягнення. Або змога змінити аватарку. Ці речі ніяким чином не покращають якість знань або підвищують швидкість навчання, але це може грати роль невеличкого стимулу або загального підвищення настрою, що може позитивно відобразитися на успіхах у навчанні.

ВИСНОВКИ

В сучасному світі інтернет-технологій, веб-розробка, являється одним з провідних методів розробки програмного забезпечення. Веб-розробка займає лідируючі позиції в списку технологій розробок, за рахунок універсальності для всіх можливих девайсів, починаючи від сучасних fullhd телевізорів і закінчуючи портативними смартфонами.

В свою чергу веб-розробка, як технологія, складається з двох складових, сторони інтерфейса (frontend) та серверної сторони (backend). Кожна із цих частин веб-розробки представляє свою цінність і не може утворювати повноцінну веб-технологію поодиноці.

Frontend, як частина веб-розробки, яка відповідає за те, що побачить користувач перед собою, заслуговує особливої уваги, оскільки кожен front-end розробник повинен бути не просто програмістом який пише код, а і, свого роду, митцем, здібним побачити проєкт правильно, поставити себе на місце користувача і дати собі відповідь на запитання «Чи приємно і комфортно користувачу буде працювати з сайтом? ».

З усього вищесказаного, можна сміливо зробити висновок, що інтернет-технології зараз знаходяться в етапі розквіту і основні етапи ще попереду. Оскільки сучасна технологічна обстановка в світі гостро потребує якісної веб-розробки, це означає, що попит на спеціалістів такого напрямлення, як frontend не спадатиме, а отже не спадатиме попит і на якісні онлайн-курси з цього напрямку.

В наш час швидкісного і якісного інтернету та зайнятих людей, які б хотіли пізнати всі ази front-end розробки, але банально не мають вільного часу або бажання для того, щоб записуватися на курси або, тим більше, вступати до вишу, онлайн-курси, на кшталт курсу опис якого представлено в даній роботі, мають дуже велике значення і дають їм шанс опанувати професію мрії у найкоротші строки.

Відомо, що стандарти та тенденції веб-розробки frontend змінюються швидше, ніж їх можна впровадити. Тому розробникам дуже важливо бути в курсі останніх мов програмування, фреймворків та технологій, щоб залишатися

актуальними та конкурентоспроможними у галузі, що постійно розвивається. Подібні онлайн-курси, дають змогу надолужити пропущене, посилити свої слабкі сторони, заповнити пробіли і стати професіоналом своєї справи.

Без сумніву, онлайн-курси це теперішнє і майбутнє освітнього процесу, особливо для людей які хочуть освоїти щось нове, але мають на це не так багато часу. Основний плюс подібних курсів закладається в тому, що здобувати знання і навички людина повинна сама, для цього людині потрібні бажання і розуміння того, що це дійсно потрібно, а це, в свою чергу, і є ключом до успішного здобуття і засвоєння будь-яких знань і навичок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гото Келлі, Котлер Емілі. Веб-редизайн, 2-е видання. СПб. : «Символ-Плюс», 2006. С. 416.
2. Марко Беллиньясо. Розробка Web-додатків в середовищі ASP.NET 2.0: завдання - проєкт - рішення = ASP.NET 2.0 Website Programming: Problem - Design - Solution. М. : «Діалектика», 2007. С. 640
3. Оліщук Андрій Володимирович. Розробка Web-додатків на PHP 5. Професійна робота. М. : «Вільямс», 2006. С. 352.
4. "What is Web Development? – Definition from Techopedia". URL: <https://www.techopedia.com/> (Дата звернення 07.04.2021).
5. Campbell, Jennifer (2017). Web Design: Introductory. Cengage Learning. p. 27.
6. URL: <https://www.coursera.org/>.
7. URL: <https://www.udemy.com/>.
8. URL: <https://www.freecodecamp.org/>.
9. URL: <https://edabit.com/>.

ДОДАТОК А

«ЗАТВЕРДЖУЮ»

Завідувач кафедри ІКТ та ФД
к.ф.-м.н., доцент Ярмоленко М.В.

_____” _____ 2021 р.

Програмна реалізація адаптивного навчання по напрямку Front-end

Специфікація

482. КНУТД. 211281- 01

Листів 2

Розробник

Харенко О.О.

Керівник

Ярмоленко М.В.

ДОДАТОК Б

Програмна реалізація адаптивного
навчання по напрямку Front end

Інструкція користувачеві
482. КНУТД. 211281-01 34 01
Листів 8

Розробник

Харенко О.О.

2021

ЗМІСТ

<u>1 ПРИЗНАЧЕННЯ ПРОГРАМИ</u>	<u>3</u>
<u>2 РОБОТА З ПРОГРАМОЮ</u>	<u>3</u>

1 ПРИЗНАЧЕННЯ ПРОГРАМИ

Даний дипломний проект, представляє собою автономну, інтерактивну онлайн-систему навчання по веб-розробці, а саме по одній з її частин front-end.

В наш час передових технологій, коли навіть працювати можна не виходячи з дому, дистанційне онлайн-навчання вже давно стало невід’ємною частиною освітнього процесу. Це чудовий спосіб швидко отримати потрібні знання та навички, при цьому зберігаючи соціальну дистанцію, що до вподоби багатьом людям, які відчувають себе некомфортно в великих групах людей. До того ж, такі проекти дають змогу людям, які працюють, отримувати нові навички або навіть освоювати нову професію, залишаючи багато часу і для основної роботи.

2 РОБОТА З ПРОГРАМОЮ

1. Зайти на сайт за допомогою посилання



Рисунок Б.1 – Стартова сторінка

2. Натиснути на кнопку реєстрації у правому верхньому куті

3. Після вибору кнопки реєстрації, ввести ім'я і підтвердити реєстрацію

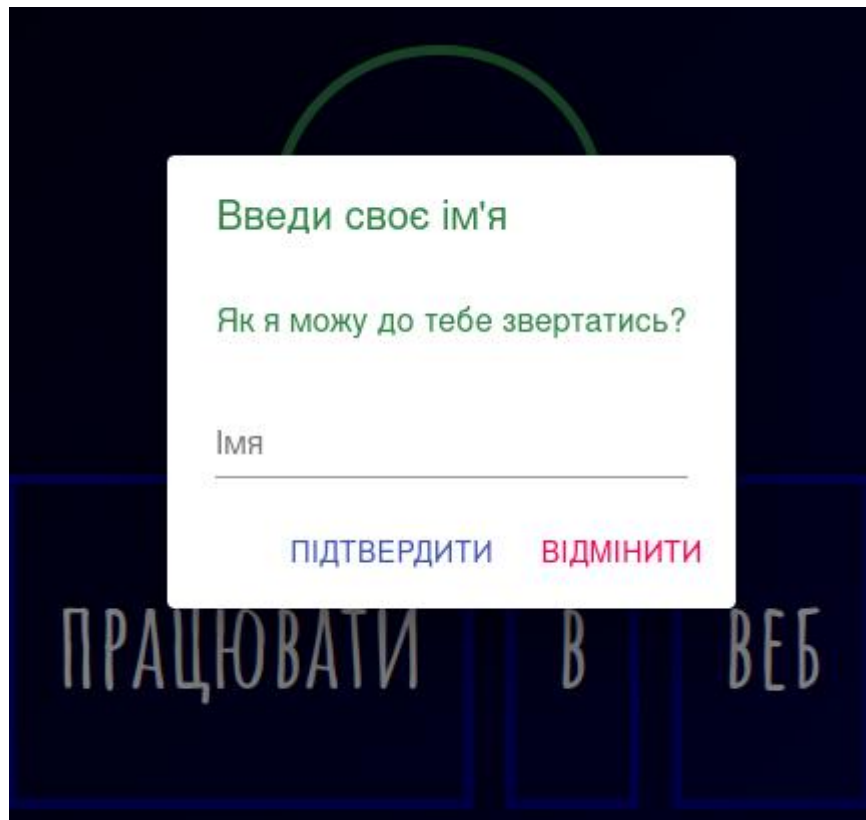


Рисунок Б.2 – Модальне вікно реєстрації

4. Після процесу реєстрації необхідно вибрати мову додатку в діалоговому вікні, на вибір українська і англійська

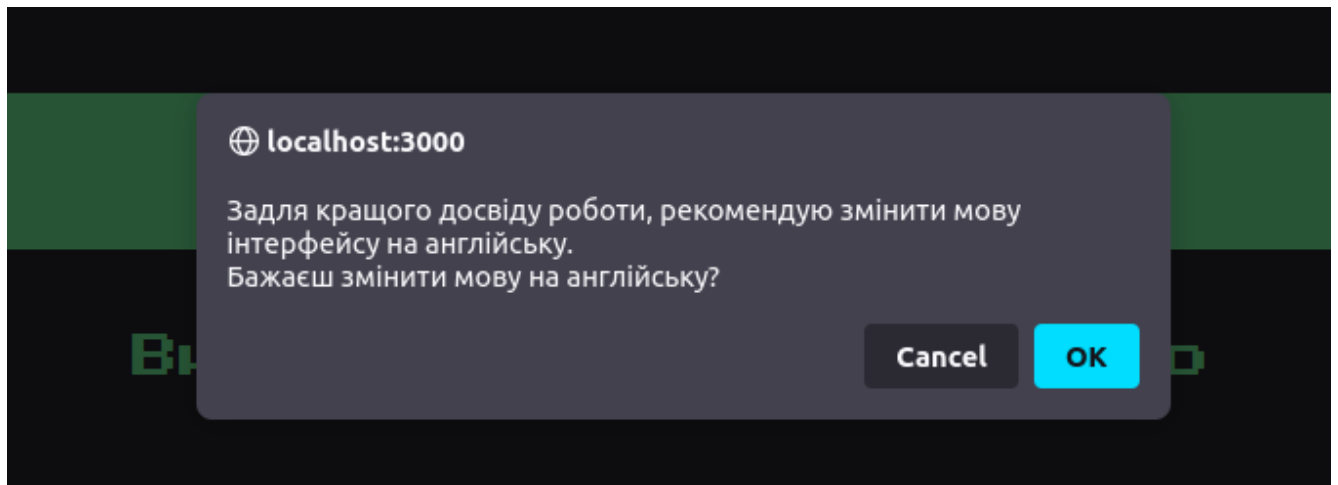


Рисунок Б.3 – Діалогове вікно вибору мови додатку

5. Після вибору мови додатку, потрібно вибрати спосіб початку навчання:

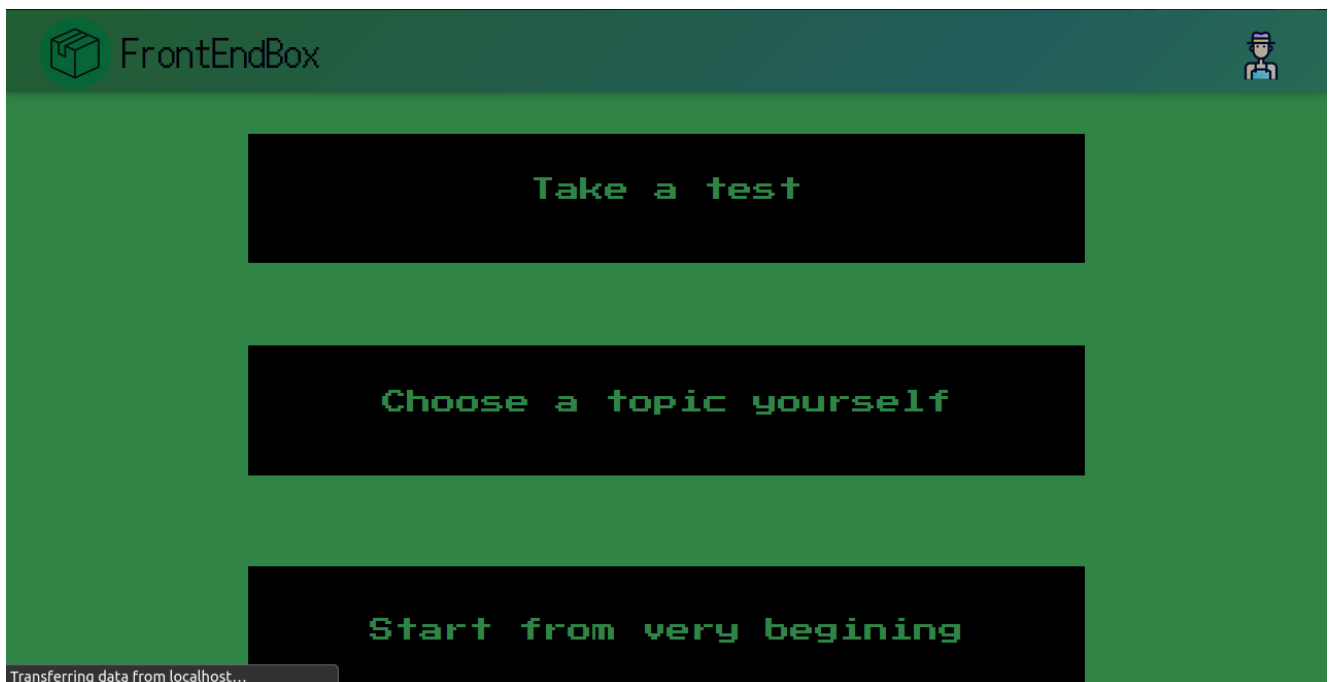


Рисунок Б.4 – Сторінка вибору способу початку навчання

1. Пройти кваліфікаційний тест (верхня кнопка);

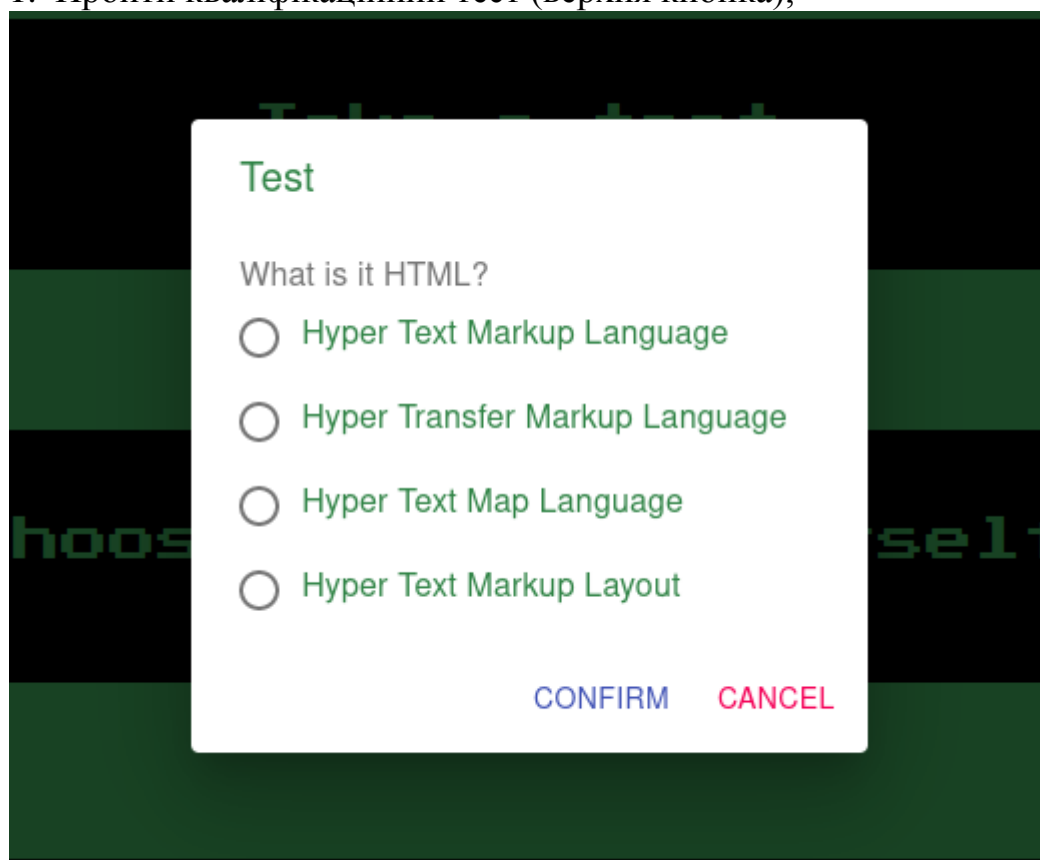


Рисунок Б.5 – Вікно тестування

2. Вибрати тему самостійно (середня кнопка);

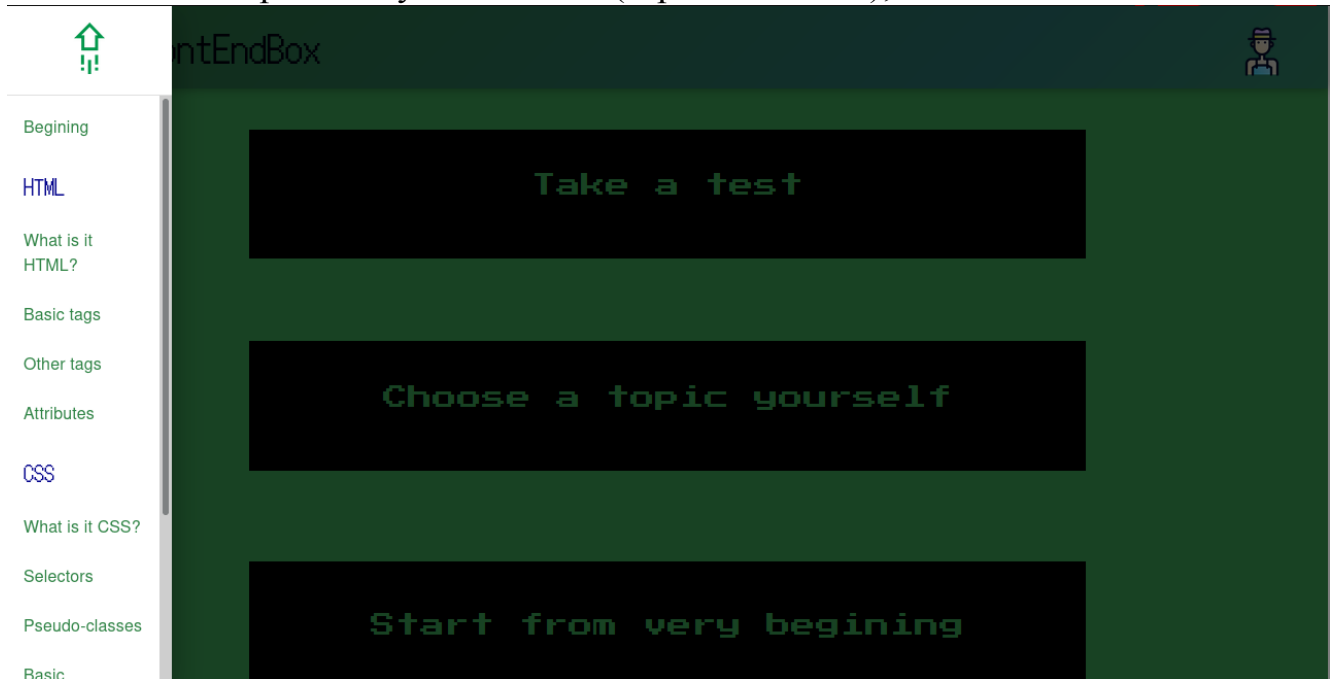


Рисунок Б.6 – Вікно вибору теми

3. Почати з самого початку (нижня кнопка).

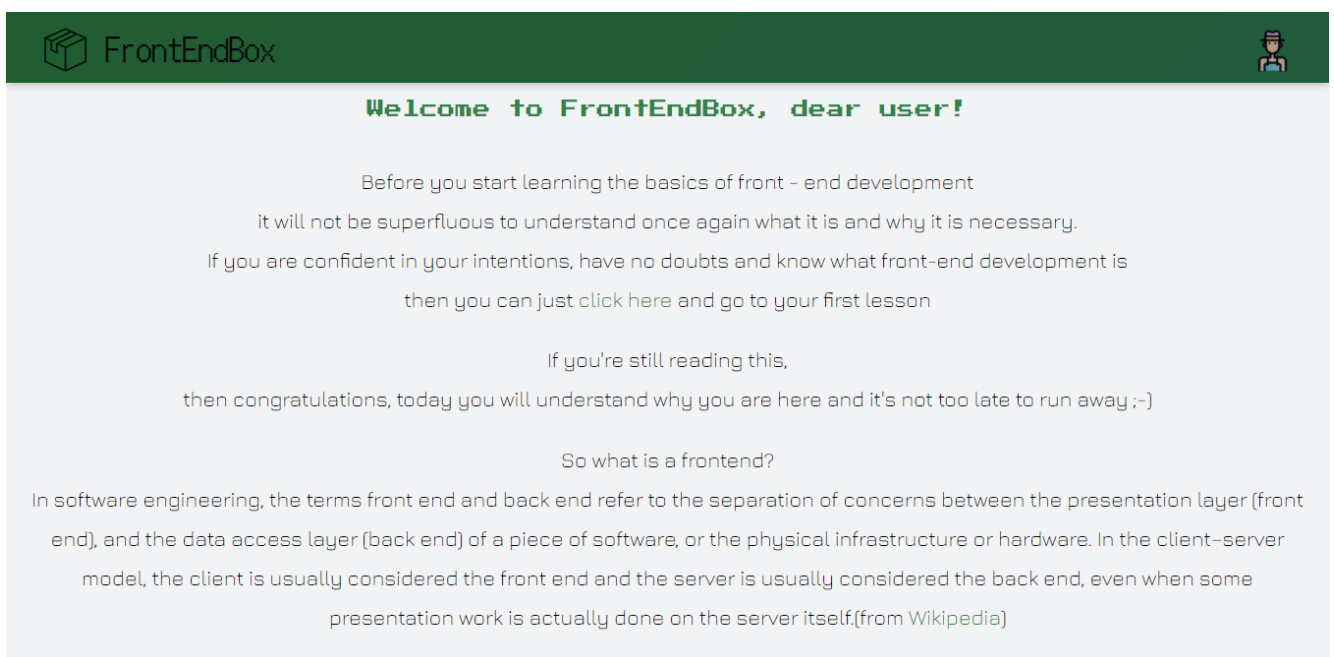


Рисунок Б.7 – Вступна сторінка

5. Після вибору способу початку навчання, необхідно ознайомитися зі змістом уроку



What is it HTML?

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML

Read fonts.gstatic.com

Рисунок Б.8 – Сторінка уроку

6. Ознайомившись зі змістом уроку, в кінці сторінки розташовані посилання на статтю та відео з додатковим матеріалом і кнопка для переходу на наступний урок

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Conclusion

Essentially, HTML is the basis of any website or web application. It combines the external part of the site (CSS) and the logical part (JavaScript). But we'll talk about them later.

If you want to know more

[Article](#)
[YouTube](#)

[Next lesson](#)

Рисунок Б.9 – Нижня частина сторінки

7. Для взаємодії з додатком, існує користувацьке меню за допомогою якого можна змінити тему додатку, змінити мову, ім'я, аватар і вийти з додатку.

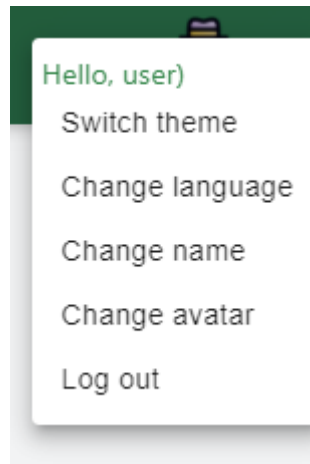


Рисунок Б.10 – Користувацьке меню

ДОДАТОК В

Програмна реалізація адаптивного
навчання по напрямку Front end

Лістинг програми

482. КНУТД. 211279-01 34 01

Листів 10

Розробник

Харенко О.О.

2021

ЗМІСТ

<u>1 ПРИЗНАЧЕННЯ ПРОГРАМИ</u>	<u>3</u>
<u>2 РОБОТА З ПРОГРАМОЮ</u>	<u>3</u>

```
import React from 'react';  
import './App.scss';  
import RouteMap from './components/RouteMap';
```

```
function App() {  
  return (  
    <RouteMap />  
  );  
}
```

```
export default App;  
import React from 'react';  
import { Header } from './Header/Header';  
import { Main } from './Main/Main';  
import { Footer } from './Footer/Footer';
```

```
export function IntroducePage() {  
  return (  
    <>  
      <Header />  
      <Main />  
      <Footer />  
    </>  
  );  
}
```

```

    );
}
import React, { useState, useEffect } from 'react';
import { WOW } from 'wowjs'
import { animateScroll as scroll } from 'react-scroll'
import clsObj from './header.module.scss'
import { LogInBtn } from './LogInBtn';
import { UserBtn } from './UserBtn';
import { Navbar } from '../../dashboard/Modals/Navbar'
import { Link } from 'react-router-dom';

```

```

export function Header() {

```

```

    const [open, setOpen] = useState(false)

```

```

    useEffect(() => {

```

```

        const wow = new WOW({ live: false })

```

```

        wow.init()

```

```

    }, [])

```

```

    const handleClick = () => {

```

```

        if (localStorage.name === "") {

```



```

        scroll.scrollToTop()
    } else {
        setOpen(true)
    }
}

```

```
const updateState = (newValue) => { setOpen(newValue) }
```

```

return (
    <>
        <header className={clsObj.header__Block}>
            <div className={clsObj.headerLogo__Block}>
                <a className={clsObj.logoLink__btn + " wow
bounceInLeft"} onClick={handleClick}>
                    <img alt="Logo"
className={clsObj.logoLink__img} />
                </a>
                {localStorage.name !== "" && localStorage.name
!== undefined
                ? <Link to="/dashboard"><span
className={clsObj.headerTitle__txt}>FrontEndBox</span></
Link>

```

```

      : <span
className={clsObj.headerTitle__txt}>FrontEndBox</span>

    </div>
    {localStorage.name !== "" && localStorage.name !==
undefined ? <UserBtn /> : <LogInBtn /> }
  </header>

  <Navbar navState={open} updateState={updateState}
/>
  </>
);
}
import React, { useEffect } from 'react';
import clsObj from './main.module.scss'
import { MotivationBlock } from './MotivationBlock';

export function Main() {

  useEffect(() => {
    const emptyItem = ""
    localStorage.setItem("name", emptyItem)
    localStorage.setItem("lang", emptyItem)

```

```

localStorage.setItem("achiv1", emptyItem)
localStorage.setItem("achiv2", emptyItem)
localStorage.setItem("achiv3", emptyItem)
localStorage.setItem("achiv4", emptyItem)
localStorage.setItem("theme", "light")
localStorage.setItem("avatar",
"/static/media/user.7ffa5473.png")
}, [])

return (
  <main className={clsObj.main__Block}>
    <MotivationBlock animation=" wow bounceInUp"
text="Мпiew?" blockType="title" />
    <div className={clsObj.motivRow__Block}>
      <MotivationBlock animation=" wow
bounceInDown" text="працюєamu" />
      <MotivationBlock animation=" wow bounceInUp"
text="є" animtime="2s" />
      <MotivationBlock animation=" wow
bounceInDown" text="веб" />
    </div>
    <div className={clsObj.motivRow__Block}>

```

```

    <MotivationBlock animation=" wow
bounceInDown" text="стату" />
    <MotivationBlock animation=" wow bounceInUp"
text="front-end" />
    <MotivationBlock animation=" wow
bounceInDown" text="розробником" />
</div>
<div className={clsObj.motivRow__Block}>
    <MotivationBlock animation=" wow
bounceInDown" text="вдосконалити" />
    <MotivationBlock animation=" wow bounceInUp"
text="свої" />
    <MotivationBlock animation=" wow
bounceInDown" text="знання" />
</div>
<MotivationBlock animation=" wow bounceInUp"
text="Бажаєш?" blockType="title" />
<div className={clsObj.motivRow__Block}>
    <MotivationBlock animation=" wow
bounceInDown" text="навчитися" />
    <MotivationBlock animation=" wow bounceInUp"
text="правильно" />

```

```

        <MotivationBlock animation=" wow
bounceInDown" text="програмувати" />
    </div>
    <div className={clsObj.motivRow__Block}>
        <MotivationBlock animation=" wow
bounceInDown" text="зрозуміти" />
        <MotivationBlock animation=" wow bounceInUp"
text="принцип" />
        <MotivationBlock animation=" wow
bounceInDown" text="технологій" />
    </div>
    <div className={clsObj.motivRow__Block}>
        <MotivationBlock animation=" wow
bounceInDown" text="стати" />
        <MotivationBlock animation=" wow bounceInUp"
text="майстром" />
        <MotivationBlock animation=" wow
bounceInDown" text="справу" />
    </div>
    <MotivationBlock animation=" wow bounceInUp"
text="Але..." blockType="title" />
    <div className={clsObj.motivRow__Block}>

```

```

        <MotivationBlock animation=" wow
bounceInDown" text="відчуваєш" />
        <MotivationBlock animation=" wow bounceInUp"
text="себе" />
        <MotivationBlock animation=" wow
bounceInDown" text="невпевнені" />
    </div>
    <div className={clsObj.motivRow__Block}>
        <MotivationBlock animation=" wow
bounceInDown" text="не знаєш" />
        <MotivationBlock animation=" wow bounceInUp"
text="чи це" />
        <MotivationBlock animation=" wow
bounceInDown" text="мвоє" />
    </div>
    <div className={clsObj.motivRow__Block}>
        <MotivationBlock animation=" wow
bounceInDown" text="не маєш" />
        <MotivationBlock animation=" wow bounceInUp"
text="багато часу" />
        <MotivationBlock animation=" wow
bounceInDown" text="на навчання" />
    </div>

```

```

    <MotivationBlock animation=" wow bounceInUp"
text="Є came me" blockType="title" />
    <div className={clsObj.motivRow__Block}>
        <MotivationBlock animation=" wow
bounceInDown" text="що" />
        <MotivationBlock animation=" wow bounceInUp"
text="мобі" />
        <MotivationBlock animation=" wow
bounceInDown" text="потрібно" />
    </div>
    <MotivationBlock animation=" wow
bounceInDown" text="FrontEndBox" textType="latin" />
    <div className={clsObj.motivRow__Block}>
        <MotivationBlock animation=" wow
bounceInDown" text="came me" />
        <MotivationBlock animation=" wow bounceInUp"
text="що му" />
        <MotivationBlock animation=" wow
bounceInDown" text="шукати" />
    </div>
    <MotivationBlock animation=" wow bounceInUp"
text="Що це make?" blockType="title" />

```

```

        <MotivationBlock animation=" wow
bounceInDown" text="FrontEndBox" textType="latin" />
        <div className={clsObj.motivRow__Block}>
            <MotivationBlock animation=" wow
bounceInDown" text="це абсолютно" />
            <MotivationBlock animation=" wow bounceInUp"
text="автоматизована" />
            <MotivationBlock animation=" wow
bounceInDown" text="система навчання" />
        </div>
        <MotivationBlock animation=" wow
bounceInDown" text="ніяких викладачів" />
        <MotivationBlock animation=" wow bounceInUp"
text="тільки ти," />
        <MotivationBlock animation=" wow
bounceInDown" text="твоє бажання вчитися і" />
        <MotivationBlock animation=" wow flipInX"
text="FrontEndBox" textType="latin" />
    </main>
    );
}

```