3. 2018 Index of Economic Freedom. [Electronic resource]: [Web site]. – Electronic data. – Access mode: https://www.heritage.org/index/ – Title of the screen

4. Types of social responsibility of business - Journal «Sustainable Business». [Electronic resource]: [Web site]. – Electronic data. – Access mode: http://csrjournal.com/vidy-socialnoj-otvetstvennosti-biznesa - Title of the screen

5. Milton Friedman. Capitalism and Freedom. - Publ. University of Chicago Press, 1962, 166 p.

Berezovsky Volodymyr Volodymyrovych

Kyiv National University of Technologies and Design

Scientific supervisor PhD, associate professor Absaliamova Ya.V.

MODERN FRONT-END DEVELOPMENT STACK

Introduction. Development stack has always been one of the most complicated topics among engineers. In this article we will discuss what kind of environment should be used and what patterns to accomplish in future projects.

Formulation of the problem. Nowadays, there are lots of tools a developer can get use of. It can be a simpler code editor, a library, or even a framework. Unfortunately, there are so many of them, that instead of writing code, we prefer to waste time on choosing the right tool.

The aim of the thesis is to meet the present requirements of web development and analyze the role of frameworks in this field.

The study object is front-end development, from basics (markup, static websites) to advanced features, such as bundling, javascript patterns, usage of different libraries, etc.

The methods of observation, classification, generalization and description are used in this paper.

The scientific novelty of the work is that for the first time the author tried to systematize information for starters, focused on the role of exploring web stack.

Main material. The topic of front-end development is a complicated one, so we have divided our discussion into small parts.

The JavaScript Renaissance. NodeJS was announced in 2009. It was more than just JavaScript on the command line or a server. NodeJS revolutionized a concentration of software development around something that was so desperately needed: an ecosystem, that is focused on the front-end developer. Thanks to Node and package manager (npm), JavaScript saw a renaissance in how applications could be architected as well as how they were developed. The ecosystem thrived, but because it was young it also constantly churned.

Happily, the past few years have allowed certain patterns and conventions to rise to the top. In 2015, the JavaScript community saw the release of a new spec, ES2015, along with an even greater explosion in the ecosystem.

As JavaScript matured, so did our approach to development processes. Building off passion for developing well-designed, maintainable, and mature software applications for our clients, it is understood that success always starts with a strong local development workflow and stack.

Challenges and Opportunities. With so many choices, and such a robust and blossoming ecosystem at present, where does that leave the community? While having choices is a good thing, it can be difficult for organizations to know where to start, what they need to be successful, and why they need it. As user expectations grow for how an application should perform and behave (load faster, run more smoothly, be responsive, feel native, etc.), it gets ever more challenging to find the right balance between the productivity needs of the development team and the project's ability to launch and succeed in its intended market.

Chasing the latest tools and technologies can inhibit velocity and the achievement of significant milestones in a project’s development cycle, risking time to market and customer retention. At a certain point an organization needs to define its problems and needs, and then make a decision from the available options, understanding the pros and cons so that it can better anticipate the long-term viability and maintainability of the product.

The list below shows us key concepts and philosophies that ensure our decisions will help solve the challenges we have come to expect from developing software for the front end:

- Leverage the latest features available in the JavaScript language to support more elegant, consistent, and maintainable source code (like import / export (modules), class, and async/await).
- Provide a stable and mature local development environment with low-to-no maintenance (that is, no global development dependencies for developers to install or maintain, and intuitive workflows / tasks).
- Adopt a single package manager to manage front-end and build dependencies.
- Deploy optimized, feature-based bundles (packaged HTML, CSS, and JS) for smarter, faster distribution and downloads for users. Combined with HTTP/2, large gains can be made here for little investment to greatly improve user experience and performance.

A New Stack. In this series, our focus is on three core components of a front-end development stack. For each component, we will look at the tool that we think brings the best balance of dependability, productivity, and maintainability to modern JavaScript application development, and that are best aligned around our desired principals.

Package Management: Yarn. Yarn shows itself off by ensuring that the version of a vendor dependency installed today will be the exact same version installed by a developer next week. It is imperative that this process is frictionless and reliable, distributed and at scale, because any downtime prevents developers from being able to code or deploy their applications. Yarn aims to address these concerns by providing a fast, reliable alternative to the npm cli for managing dependencies, while continuing to leverage the npm registry as the host for public Node packages. Plus it is backed by Facebook, an organization that has scale in mind when developing their tooling.

Application Bundling: webpack™. The orchestration of building a front-end application, which is typically comprised of a mix of HTML, CSS, and JS, as well as binary formats like images and fonts, can be tricky to maintain and even more challenging to orchestrate. So how does one turn a code base into an optimized, deployable artifact? Gulp? Grunt? Browserify? Rollup? SystemJS? All of these are great options that provide their own strengths and weaknesses, but we need to make sure the choice reflects our intended principals as discussed above.

Webpack is a build tool specifically designed to package and deploy web applications comprised of any kind of potential assets (HTML, CSS, JS, images, fonts, and so on) into an optimized payload to deliver to users. We want to take advantage of the latest language features like import / export and class to make our code future-facing and clean, while letting the tooling orchestrate the bundling of our code such that it is optimized for both the browser and the user. webpack can do just that, and more!

What About Front-End Frameworks? Different applications call for different approaches to their development based on many factors like team experience, scope and size, organizational preference, and familiarity with concepts like reactive or functional programming. It is believed that evaluating and choosing any ES2015 compatible library or framework, be it Angular 2 or React, should be based on characteristics specific to the given situation.

Conclusion. Summarizing, we can say that software development is a complicated process, requiring attention and solutions. We took a brief look at the revolution in JavaScript development over the last few years. We have discussed some of the challenges and opportunities facing the front-end development community.

REFERENCES

1. Steve Sounders, High Perfomance Web Apps. – 2007. – P. 1 – 14.
2. Branneman E., Babel JS // documentation. – 2016. –P. 3-9.
3. Douglas Crockford, JavasScript the Good Parts. – 2008. – P. 120 – 131.
4. Kyle Simpson, You don’t know JS: ES6 and Beyond. – 2016. – P. 1 – 5.

5. Owen Buckley, wtfrontend https://www.linux.com/blog/learn/2017/7/modern-day-front-end-development-stack – 2017. – P. 1 – 4.
6. Robert Cecil Martin, Clean Code. – 2014. – P. 40 – 63.

Бобко Євгеній Олександрович

Київський національний університет технологій і дизайну

м. Київ

Науковий керівник – Роєнко Л. В.

MOTIVATIONAL MANAGEMENT IN TOURISM

Human resources management is one of the most important components of tourist company management. The high level of the employee motivation is a guarantee of the ever-increasing level of tourist services quality.

Based on the features of the qualitative characteristics of the service, it is not possible to make a good impression on a customer without involving a human factor in it. It is this factor that stimulates a potential client to pay significantly more than this service could cost in a situation where the influence of the human factor would be less significant. Motivational management involves creating priority conditions for the creative and highly professional activity of the immediate service provider.

The purpose of the research is to justify the role of motivation in improving the activities of the travel company on the basis of systematization of existing knowledge concerning this issue.

According to the aim the following **tasks** have been identified: consider and analyse the existing points of view relevant to the given problem, explain and specify the role of motivation in travel company management, identify the main activities in human resources management in the sphere of tourism.