

Usage of Self Created Mind Map Environment in Education and Software Industry

Inessa Krasnokutska^{ORCID 0000-0002-7034-7291} ✉, and Maksym Ridush

Chernivtsi National University, Chernivtsi, Ukraine
i.krasnokutska@chnu.edu.ua, maksridush@gmail.com

Abstract. The report describes the use of mind maps as a modern tool of learning and teaching. It describes possibilities of using mindmapping for teachers in classical universities or mentors and developers in IT-companies for presenting new material, developing a course or evaluating developers' understanding, as well as for developers studying a specific subject, preparing for examinations and high-quality memorization of information. Authors designed and developed their own new software solution for easy creating and editing and sharing mind maps with an intuitive user interface and efficient abilities. The report contains description of implementation of mind maps usage in software development during training of students in Chernivtsi National University and software engineers in IT-company SoftServe. The process of developing new one-page application for creating mind maps using the React library in the JavaScript language is being described.

Keywords. mind maps, online environment, modern learning methods, web-technologies in mentoring.

1 Development New Web Environment for Mind Maps Creating

Educational facilities have access to world information resources, new means and methods of teaching and learning with scientific and educational systems have emerged [1]. The educational process as information exchange includes processes of visualization, transfer, preservation, and processing of information [2, 3]. The visual communication is the most natural way of human interaction. In education, the importance of visual communication is the most obvious, as its methods carry out the management of the educational process when students use resources for independent learning. Mind mapping is a technique of thought visualization that helps to express and perceive information in better and more effective way [4].

This paper contains an overview of a process of development of client web application, which can be used to display, build and edit mind maps. The available tools and approaches to the development of web applications such as platform, architecture, frameworks and libraries were analyzed. In a process of research, there were created a list of functional requirements for **MVP** and they were implemented.

Developed software allows users easily create and edit mind maps, has the ability to export **PNG** images and an intuitive user interface created using a component approach and combining components from user-defined packages and components developed by the author. The list of functional requirements for the application was compiled, and software code that satisfies them was developed.

For development, the text editor of the code **VS Code** was used, with some extensions for ease development. **Git** was used as a version control system, the program code is located in an open repository on the **GitHub** service.

1.1 Analysis and Selection of Tools for Implementation of the Application

As a platform for the application was chosen **web platform**, because it does not require additional installation and it is universal for all devices: personal computer, laptop, tablet, mobile phone, and does not depend on the installed operating system.

When choosing the application architecture, preference was given to **client-server architecture** and client application development, although the option of server-side development for **MVC**-based relationship was also considered. Client applications are separate applications with their own code-base and usually interact with the server using **HTTP** queries. Server applications process requests and return an **HTTP** response, usually with a body in the **JSON** format.

Since the end product is a web application, then the obvious choice of programming language is JavaScript. For development **ES6** standard approved in 2015, has been used.

When choosing a framework, the preference is given to **React** because it does not have an **MVC** on its own. **React** is a declarative, productive and flexible **JS** library for developing user interfaces. One of the key features of **React** is the use of **JSX** syntax, an extension of the **JS** syntax with **HTML** tags.

Semantic UI React was used to stylize the user interface. It contains a **Semantic UI** style library and integrates with **React**. Mind maps have a tree structure and, accordingly, are objects for display by libraries that build trees and graphs, so for implementation was selected **Cytoscape.js** – a library for analyzing and visualizing graphs [5].

1.2 Functional Requirements

The list of functional requirements for an **MVP** software product was developed. **MVP** describes the minimum viable product with a set of capabilities that is sufficient for effective usage.

Free open source crossover editor **VS Code** was used as development environment. **VS Code** has built-in support for a distributed version control system **Git** and convenient features for debugging software code. The editor has built-in support for **JS** and its dialects, including **ES6** and **JSX**. To create the skeleton of the project, the **Create-react-app** tool was used. It contains most of the mechanisms necessary for the development of tools, such as a local static server, auto prefixers for **CSS** files, scripts to run a project assembly, and built-in support for the **Webpack** collection.

The installation of libraries for the project was handled with **NPM**, the package manager for **JS**. In addition to the **Create-react-app** packages installed during the project initialization, **Cytoscape.js**, **Semantic UI React**, and some others are added to the **package.json** configuration file. All listed packages can be found in the **NPM** registry.

1.3 Implementation of the Application

React-router was used for configuration routing. It provides navigation component package for declarative routing description.

Using **Cytoscape.js** library, the initial mind map was created. It displays when user opens the application. It contains descriptive information about the features and abilities of the application.

To give the user the opportunity to conveniently and efficiently editing of mind map, a context menu was created. It is placed directly above the active element. The context menu allows use to create new node, edit information in the current node, add a picture to the node and delete current node.

Before the application release user interface was improved with new features: colored connections between primary and secondary map elements, linear edges were replaced with more appealing **Bezier curves**. To implement edge thinning depending on a distance from map root it can be used a recursive tree traversal algorithm, but because acyclic map support can be implemented in future releases we used **Belman-Ford algorithm**, which is built-in in **Cytoscape.js**. The algorithm computes shortest paths from a root node to all other elements. Then the edge width is calculated as an inversely proportional value to the distance from the main node.

1.4 Application Release and Usability Overview

The deployed application is accessible over the internet. A built-in bundler from **create-react-app** was used for the release version.

The developed application is essentially a web-client, therefore, it only needs a static server to be deployed. For this purpose, we used a free static hosting on **GitHub** platform, where the application codebase is situated. To reduce the amount of manual work the process of deployment was automated. To do so we used **gh-pages** library, that runs **build** command and automatically copies build results into a separate git branch "gh-pages". **GitHub** then automatically updates the static server with new releases that are pushed into "gh-pages" branch. The application is accessed through a link hosted on **GitHub** domain: <https://makalkin.github.io/mindblown/>.

By launching the application on a local server or by using live version one can start making mind maps by following initial instructions.

2 Recommendations for Future Improvements

Firstly, it is worth to create automated tests for the developed application to prevent regression. This can be done with a help of frameworks that serve just this purpose like **Mocha**, **Jasmine** or **Jest**. It is important to take into the account compatibility of frameworks and technologies used in the application and a testing framework.

An important aspect of application improvement is the ability to automatically place elements in all directions from the main node and support of acyclicity. A custom extension for **Cytoscape.js** can be implemented to achieve the former.

To make application collaborative it needs functionality for cooperative editing of a mind map and cloud saving. This can be implemented with cloud services like **Firebase** or similar, which do not require user to own a dedicated server, and so collaboration can be achieved with help of event streams and web-sockets.

For simplicity, the color of main branches is generated randomly and inherited by all its descendants. There is always room for optimisation and other functional improvements of software. The feedback and expectations of the target audience are valuable pieces of information that should be used to set priorities. So before further improvements, we need to collect statistics, analytics, and user feedback.

The advantage of the application is its simplicity and minimalistic **UI**, which turns all available space except browser header into a canvas. One can use a fullscreen feature of a browser for full immersion. There's also no registration required prior to using the application.

As a drawback, the application is dependent on **Cytoscape.js**. It has helped us to implement features fast with help of its built-in methods but at the same time, it made the application limited, because it relies on a support of the library by its developers. For future releases, it is recommended to implement extensions for the library or implement a standalone library, but that is quite time-consuming and demanding task.

References

1. Krasnokutska, I., Kovalchuk, O.: Using Blogs in Teaching English to Philology Students. *Advanced Education* 7, 146-153 (2017). DOI 10.20535/2410-8286.97295
2. Nikulova, G. A., Podobnyh, A. V.: Tools of Visual Communication – Infographics and Metadesign. *Obrazovatelnye Tehnologii i Obschestvo* 13(2), 369–387. Available: http://ifets.ieee.org/russian/depository/v13_i2/pdf/14r.pdf (in Russian)
3. Edwards, S., Cooper, N.: Mind Mapping as a Teaching Resource. *The Clinical Teacher* 7(4), 236–239 (2010). DOI 10.1111/j.1743-498X.2010.00395.x
4. Gordeeva, A. Y.: Intellect Maps as a Tool for Formation of Foreign Language Communicative Competence for Future Philologists. *Inozemni Movy* 4(72), 51–58 (2012). (in Ukrainian)
5. Franz, M., Lopes, C.T., Huck, G., Dong, Y., Sumer, O., Bader, G.D.: Cytoscape.js: a graph theory library for visualisation and analysis. *Bioinformatics* 32(2), 309–311 (2016). DOI 10.1093/bioinformatics/btv557