

MODERN TOOLS FOR SCALABLE WEB DEVELOPMENT

V. TALDYKIN

Polotsk State University, Belarus

In this work we analyze modern approaches to web development and explore a set of tools, including the vue.js JavaScript framework and the cloud based virtual Firebase server for the subsequent creation of an adaptive dynamic website of the cinema with the possibility of three-dimensional display of halls with the screen, a convenient interactive process of choosing places for reservations and a custom admin panel.

In this work, we analyze modern approaches to web development and study a promising set of tools, including vue.js [1] JavaScript framework for component application development, a library of graphic components quasar [2], a scalable real-time database Firestore, an effective authentication system, cloud data storage and hosting from Firebase (google) [3], as well as the most relevant CSS specifications, such as flex [4] and grid [5] layout for efficient construction of responsive layout of web pages, and tools for compact presentation of the html code – (template engine pug [6]) and of the CSS styles (CSS-preprocessor stylus [7]).

Using this stack of web-technologies, we develop a modern adaptive dynamic web-site of a cinema network with the possibility of three-dimensional display of halls and a convenient interactive process of choosing places for booking. The created unique administration system allows authorized administrators directly on the site to manage data of the web application and receive various statistical information on the activity of the users.

To ensure the best perception of the site by users, will be implemented an interactive 3d-model of the cinema hall, which is used (on wide browser screens) when booking a place to demonstrate the advantages and disadvantages of specific places when watching a movie. On the site, users have the opportunity to view a movie poster and detailed information for each of the films After registration or authorization by e-mail, the user will have the opportunity to choose the date and time of the session and book a ticket for the movie he liked. Upon successful booking of a ticket, the user will see an electronic ticket with a unique reservation code. Using this code, the user will be able to pay for the ticket he booked at the box office immediately before watching the movie.

Information about movies and cinema sessions will be stored in the Firebase real-time Firestore database. To provide the ability to quickly manipulate this data, will be created a separate administrator page directly on the website. Only authorized users with administrator rights will have access this page, and it will be possible to delete / add / edit a movie on it, including uploading a poster (image) and trailer (video file) to the Firebase cloud storage. There is also ability for administrator to edit sessions, view occupied seats (with details) on a two-dimensional plan of the cinema, manually add occupied seats (when offline buying tickets), as well as edit other session parameters (ticket price, etc.).

Important components of the frontend of modern web sites are: organic design, adaptive layout and interactivity. Note that website design should be carried out within the framework of a single own or existing design system, and to date, the most popular is material design from google [8]. Adaptive layout of the site involves the use of transformations to display page content depending on the device (desktop computer, tablet, smartphone, etc.) used for web surfing. For example, on the smartphone's screens, the site's navigation menu can change from the line display mode to the form of a drop-down menu when a button is pressed. And finally, the interactivity of the site implies the presence of graphical interface components that respond to user actions (drop-down menus, tooltips, etc.).

Development of the user interfaces (frontend parts) of such sites using only the standard (and only possible) html, CSS and JavaScript web languages is an extremely difficult task, which is why today we have acquired:

1. html template engines, which allows creating reusable compact html-code; one of the most popular is pug [6];
2. CSS-preprocessors, which allows to create compact CSS styles; one of the most popular is stylus [7];
3. JavaScript frameworks, which allow scalable component development of dynamic web applications; one of the most popular is vue.js [1] and its associated library of graphical components named quasar [2].

In recent years, serverless systems have become more relevant for the backend part of applications, allowing one to use already configured virtual servers, the functionality of which can be close to the capabilities

obtained when using own server. One of the most common serverless systems is Firebase from google [3], which allows to significantly reduce labor costs when developing applications due to the lack of the need to independently configure the server and deploy the application to the server.

The task of deploying a modern application on the own server becomes extremely difficult due to the need to develop a security system, optimize the load, back up data, etc. In this regard, the use of cloud technologies and serverless application architectures is becoming more and more popular. In this case the custom server is replaced with a virtual one provided by the provider of the corresponding service. At the same time, part (and sometimes significant) of the important opportunities that exist when using the own server is certainly lost, in particular, the processing of critical and private data that cannot be fully processed on the client side for security reasons. So, for example, validation of data before saving to the database can certainly be done on the client side, and it is really carried out for quick reaction of the application to incorrectly entered data, which significantly reduces the load on the server, however, on the client side the data can always be falsified, which makes it necessary mandatory revalidation of data on the server side. A convenient compromise is achieved by the Firebase system, which offers a complete serverless system with a wide range of features for developers, including:

1. Hosting - full hosting for web-applications. Created or updated web-application can be uploaded to Firebase using one console command, it is also possible to connect the own domain name for free;
2. Storage - cloud storage that allows users to display relevant files in a web application, as well as dynamically save user files;
3. Cloud Firestore - scalable real-time nosql database;
4. Cloud Functions - a system that makes it possible to load in Firebase specific own JavaScript functions that are performed during specified operations (reading from the database, registering a new user, etc.). This feature is significant because it allows to partially remove the main limitation of serverless approaches - the inability to execute essentially server-side code (for example, analysis of the correctness of CAPTCHA recognition, etc.);
5. Authentication - a system for registering and authenticating users in a web application by email and password, phone number, social network account, etc.

Today, due to the rapid development and growing popularity of the Internet, for almost any company, organization, or a thematically related group of people, it is important to place relevant information on the Internet that is accessible to citizens interested in it. Information on the Internet is placed in the form of static web sites or dynamic web applications, and in many cases duplication of the site / application functionality in the form of a mobile or desktop application is also important. The set of web technologies considered in this work makes it possible to develop attractive applications with minimal time.

REFERENCES

1. Vue, javascript framework, official site [Electronic resource]. – Mode of access: <https://vuejs.org> – Date of access: 01.03.2020.
2. Quasar, javascript library/framework, official site [Electronic resource]. – Mode of access: <https://quasar.dev> – Date of access: 01.03.2020.
3. Firebase, Google cloud service, official site [Electronic resource]. – Mode of access: <https://firebase.google.com/> – Date of access: 01.03.2020.
4. Flex css layout, documentation [Electronic resource]. – Mode of access: https://www.w3schools.com/css/css3_flexbox.asp – Date of access: 01.03.2020.
5. Grid css layout, documentation [Electronic resource]. – Mode of access: https://www.w3schools.com/css/css_grid.asp – Date of access: 01.03.2020.
6. Pug, html- template engine, official site [Electronic resource]. – Mode of access: <https://pugjs.org> – Date of access: 01.03.2020.
7. Stylus, css preprocessor, official site [Electronic resource]. – Mode of access: <http://stylus-lang.com> – Date of access: 01.03.2020.
8. Material design, documentation [Electronic resource]. – Mode of access: <https://material.io/design/> – Date of access: 01.03.2020.