Web Technologies 2 (Back End): Final Project

Instructor: Rakhymbayeva Nazerke

Student: Sabina Burshakbayeva

Group: BDA-2304

AITU, 2024

**Overview**

In this project, I have developed a portfolio management application that allows users to log in, create portfolios, and manage portfolio items. The application includes various features such as user authentication, two-factor authentication (2FA), image uploading, and portfolio creation. Additionally, I have integrated an external news API to display the latest news on the login page.

The project involves several key components:

* User Authentication and Registration
* Portfolio Creation and Management
* News API Integration
* User Interface (UI) Design and Styling
* User Authentication and Registration

The user authentication system allows users to log in and register. The registration process requires a username, email, password, and some additional personal information, including first name, last name, age, and gender. The password is securely hashed using the bcrypt library to ensure it is stored safely in the database.

Once a user is registered, they can log in using their credentials. During the login process, the system verifies the user's username and password, and if the login is successful, the user is redirected to the portfolio page. For added security, two-factor authentication (2FA) is implemented using the speakeasy library. This ensures that only authorized users can access their accounts.

**Portfolio Creation and Management**

The main functionality of the application revolves around creating, updating, and deleting portfolio items. Registered users can create portfolios by submitting a title, description, and images. These portfolios are stored in the database and are displayed on the user's profile page.

For each portfolio, users can upload up to three images. The image files are stored in the server’s public directory, and the paths to the images are saved in the database. Users can also update existing portfolios by modifying the title, description, and images. If new images are uploaded, they are added to the existing images, and the portfolio is updated accordingly.

Additionally, the application allows users to delete their portfolios. The portfolio management system is accessible only to authenticated users, and only users with the "admin" role can delete portfolios.

**News API Integration**

To enhance the user experience, the application integrates with the NewsAPI to display the latest news articles on the login page. This is done by making an API request to the NewsAPI, which returns a list of top headlines. The news articles are dynamically displayed on the login page.

The integration allows users to stay informed about current events while logging into the platform. If there are any issues with fetching the news, a message is displayed notifying users of the problem.

**User Interface Design and Styling**

The user interface was designed with simplicity and usability in mind. The login, registration, and portfolio pages all have a clean and modern look. Key elements, such as buttons and form inputs, are styled with pink and light coral colors to maintain a consistent and visually appealing theme throughout the application.

In particular:

* The login form is centered on the page with input fields for username, password, and 2FA code.
* The portfolio creation form allows users to upload images and add descriptions.
* The news section on the login page is displayed in a container with each article presented as a clickable link.
* CSS was used to ensure the forms, buttons, and containers are well-aligned and easy to navigate. Media queries were employed to ensure the application is responsive on different devices, making it accessible from both mobile and desktop screens.

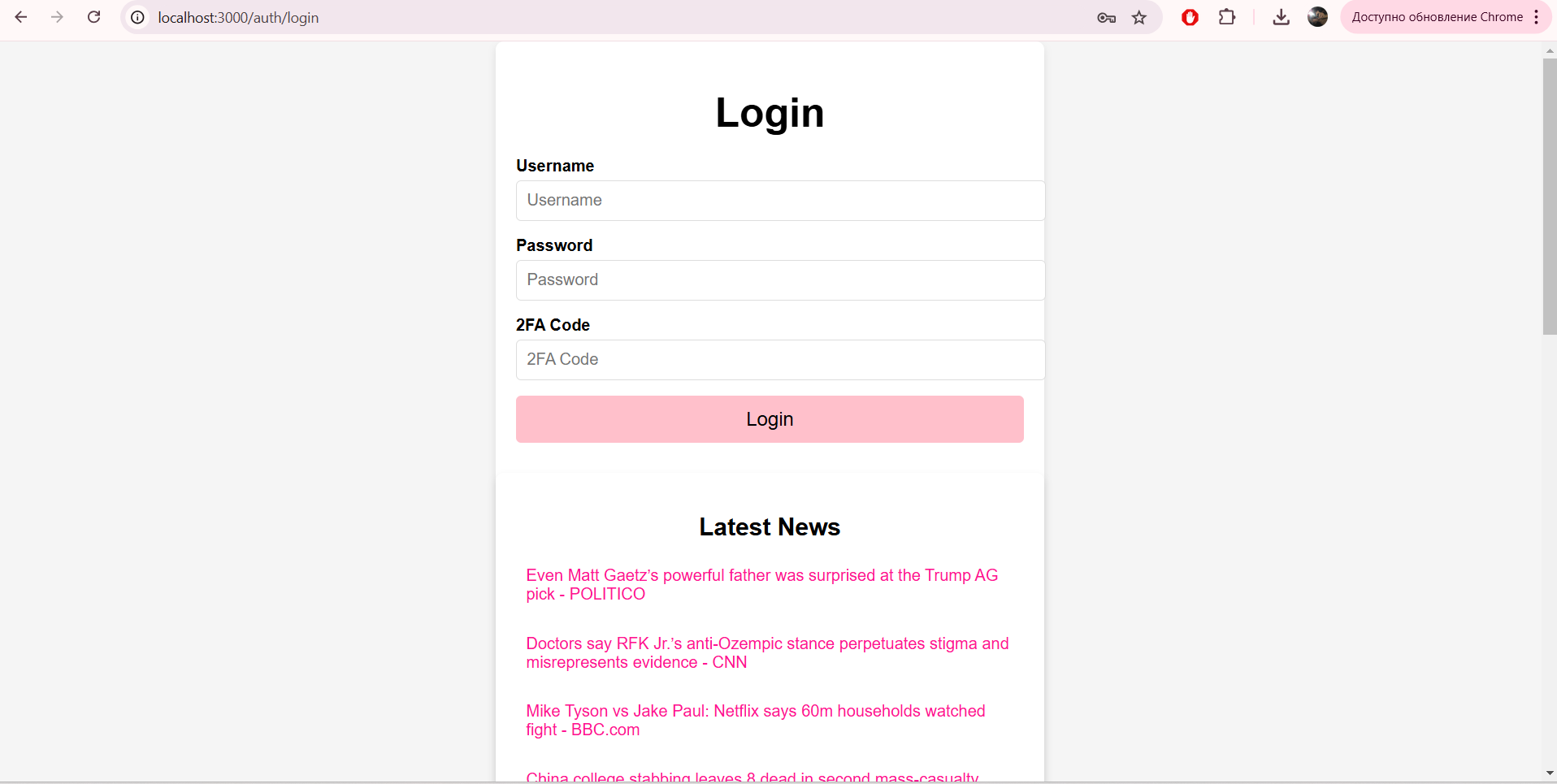
**Form Validation**: Ensuring that the registration form correctly validated the user’s input, especially for the password and email fields. This was handled by using the required attribute and ensuring the email was in the correct format.

Image Uploading: Handling file uploads posed a challenge initially, as the images had to be saved on the server and associated with the correct portfolio item. This was solved by using the multer middleware, which handled the file storage and ensured that the image paths were saved correctly in the database.

Responsive Design: Ensuring that the user interface worked well on all screen sizes required careful use of CSS, especially when dealing with the portfolio item layout and the form elements. Media queries were used to adjust the layout for different devices.

**Here are some screenshots:**

**A screenshot of a computer

Description automatically generated**

A screenshot of a food

Description automatically generated

A screenshot of a computer

Description automatically generated

**Conclusion**

This project successfully implements a portfolio management system with user authentication, portfolio creation, and integration with an external news API. The application provides a secure and intuitive interface for users to manage their portfolios while staying up to date with the latest news.

**Future improvements could include:**

* Adding more complex roles and permissions for user access control.
* Implementing additional features like comments or ratings for portfolios.
* Fixing qrcode.
* This project has greatly enhanced my skills in web development, particularly with Node.js, Express, MongoDB, and front-end design using HTML and CSS.