**Scientific Journal of Astana IT University**

ISSN (P): 2707-9031

ISSN (E): 2707-904X

**DOI**: Kanbakova Zhanel

**Given Name Surname**

BDA-2303, Big Data Analysis, Astana IT University

230486@astanait.edu.kz

**Final Project Instructions for WEB Technologies 2**

“Creating a Portfolio Platform using Node.js”

**Individual Submission:** Each student must complete this project individually.

**Submission Requirements**: Upload your project on Moodle with a GitHub link included in the comment

section before defending.

**Original Work:** Projects must be unique in both code and design; similarities will result in a 0 grade.

**Server:** Your server should run on port 3000, and your submission must include a package.json file listing

all dependencies.

**MongoDB:** Ensure that your MongoDB allows IP access from any location.

**Navigation Bar:** All pages must include a navigation bar with a clean and user-friendly layout.

**Documentation**: Include a README.md file with setup instructions, API usage details, and key design

decisions.

**Footer:** Add your name and group number in the footer.

**Final Project Report: Portfolio Platform using Node.js**

**Introduction**

This project is an individual submission for the Web Technologies 2 course, designed to create a portfolio platform using Node.js. The application incorporates essential features such as user authentication, role-based access control, portfolio management with CRUD operations, API integration with data visualization, and email notifications. The platform is designed to run on port 3000 and leverages MongoDB for database management.

The project focuses on providing a responsive and user-friendly interface, with modular and clean code following industry best practices.

**Features Overview**

**1. Authentication and Authorization**

User Registration

Users can register through a dedicated registration page by providing the following details:

Fields: Username, Password, First Name, Last Name, Age, Gender, and 2FA (Two-Factor Authentication) setup.

Security: Passwords are hashed using bcrypt before storing them securely in MongoDB.

Email Notifications: Upon successful registration, users receive a welcome email via Nodemailer.

User Login

Registered users can log in by providing:

Username and password.

If 2FA is enabled, users must also enter a verification code from their authenticator app.

Successfully authenticated users are redirected to the main page.

Role-Based Authorization

Role-based access control with two roles:

Admin: Full access to manage content (create, update, delete portfolio items).

Editor: Limited access to create portfolio items only.

Role information is stored in the database and verified during route access to enforce permissions.

Security Measures

Password Security: Passwords are hashed with bcrypt.

2FA: Ensures an extra layer of security.

Session Management: Secured with session cookies.

**2. Portfolio Management (CRUD Operations)**

Main Page

Displays portfolio items created by the admin.

Items are presented in a carousel format, with the following attributes:

Three Images: Rotating in a carousel.

Title: Describes the portfolio item.

Description: Provides additional details about the item.

Timestamps: Tracks creation, update, and deletion dates.

Admin Features

Admin users can perform all CRUD operations:

Create: Add new portfolio items with images, titles, and descriptions.

Update: Modify existing items.

Delete: Remove portfolio items from both the interface and the database.

Editor Features

Editors can create new portfolio items but cannot edit or delete existing ones.

**3. API Integration and Visualization**

API Integration

Integrated two new third-party APIs (excluding weather or previously used APIs):

Financial Data API: Provides live cryptocurrency prices and financial trends.

News API: Fetches the latest headlines related to technology and business.

Data Visualization

Created interactive charts and graphs using Chart.js to visualize data fetched from the APIs:

Financial Data Visualization: Displays live cryptocurrency trends with interactive charts.

News Analysis: Shows data trends, such as the most frequent keywords or article distribution by category.

Role Restriction:

Editors can create visualizations using API data.

Regular users are restricted from accessing visualization creation features.

**4. Notifications with Nodemailer**

Welcome Email: Sent automatically upon successful user registration.

Event-Based Notifications:

New item creation.

Item updates.

Failed login attempts (after three incorrect attempts).

**5. Project Organization and Design**

Clean Code and Structure

The project follows a modular structure, ensuring code is organized and easy to maintain.

Dependencies are listed in the package.json file.

Responsive Design

The interface is built using EJS templates and CSS for responsive and visually appealing designs.

The navigation bar and footer are consistent across all pages.

Footer

Includes the developer’s name and group number.

Navigation Bar

A user-friendly navigation bar is present on all pages, allowing seamless redirection.

Documentation

The project includes a detailed README.md with:

Setup instructions.

API usage details.

Design decisions.

Password for the admin account and 2FA setup guidance.

Registration Page

Users can sign up with required details and set up 2FA.

Login Page

Users can log in and, if 2FA is enabled, verify their identity with a code.

Admin Page

CRUD operations for portfolio management.

Portfolio Display

Main page showing a carousel with portfolio items.

Data Visualization

Financial trends and news analysis charts.

Grading Criteria Fulfillment

Feature

Authentication and Authorization: User registration, 2FA, hashed passwords, role-based access

Portfolio Management (CRUD): Portfolio items with carousel and full CRUD support

API Integration and Visualization: Two new APIs with interactive charts

Notifications: Nodemailer for welcome and event-based emails

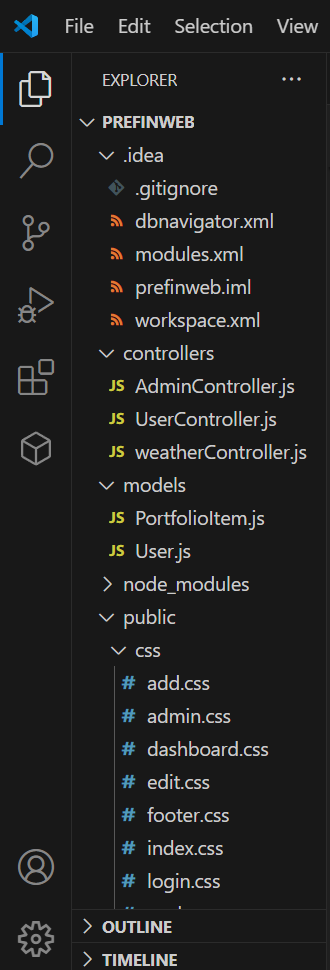
Clean Code and Responsive Design: Modular code, EJS templates, and CSS for responsiveness

Documentation: README with setup instructions and details

**Conclusion**

The Portfolio Platform is a robust application demonstrating proficiency in web technologies and full-stack development. It integrates authentication, role-based access control, CRUD operations, API-driven data visualization, and notification systems. The project meets all specified requirements and showcases modular code, secure design practices, and a user-friendly interface.

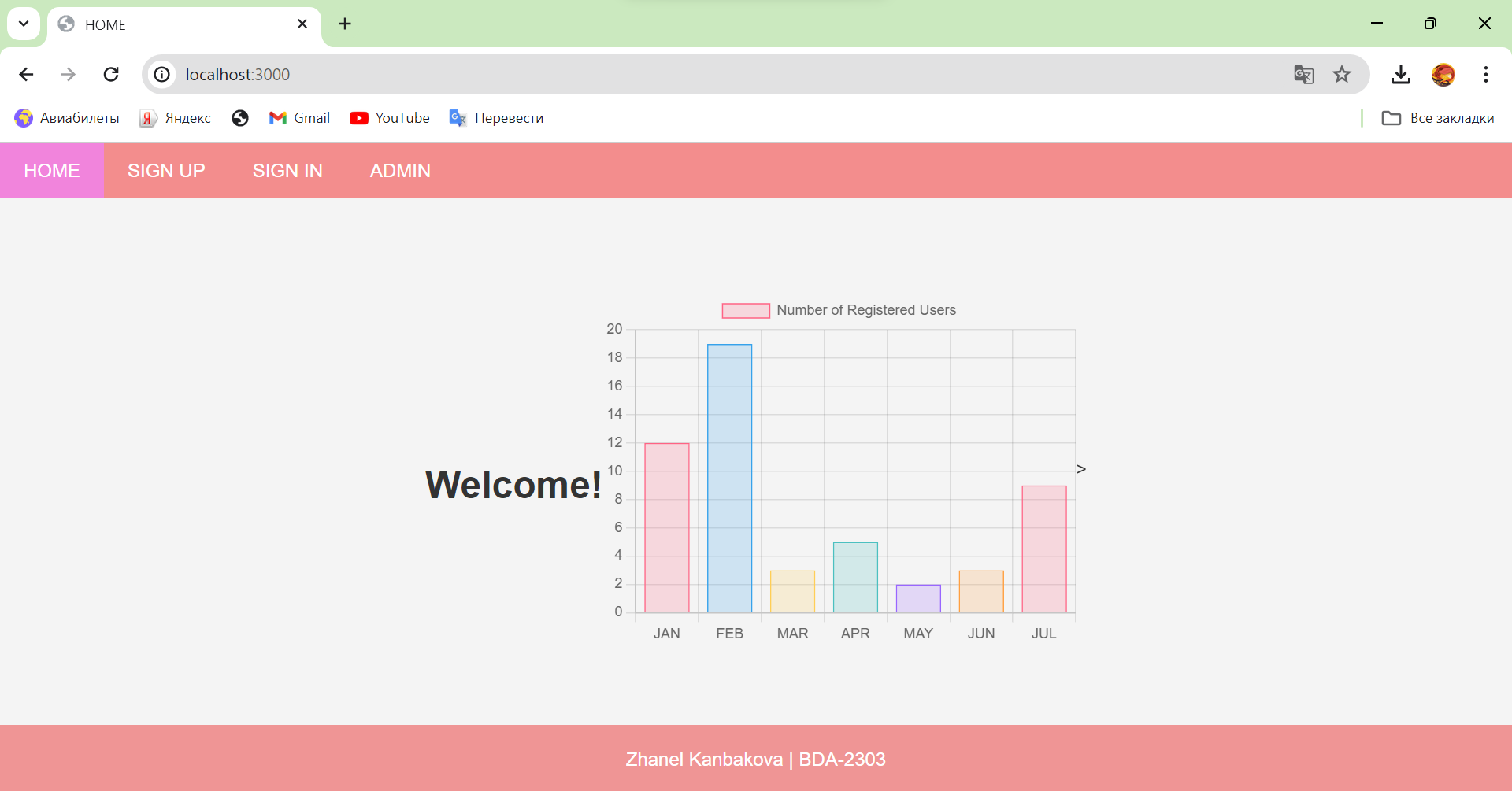
**PROJECT STRUCTURE:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение

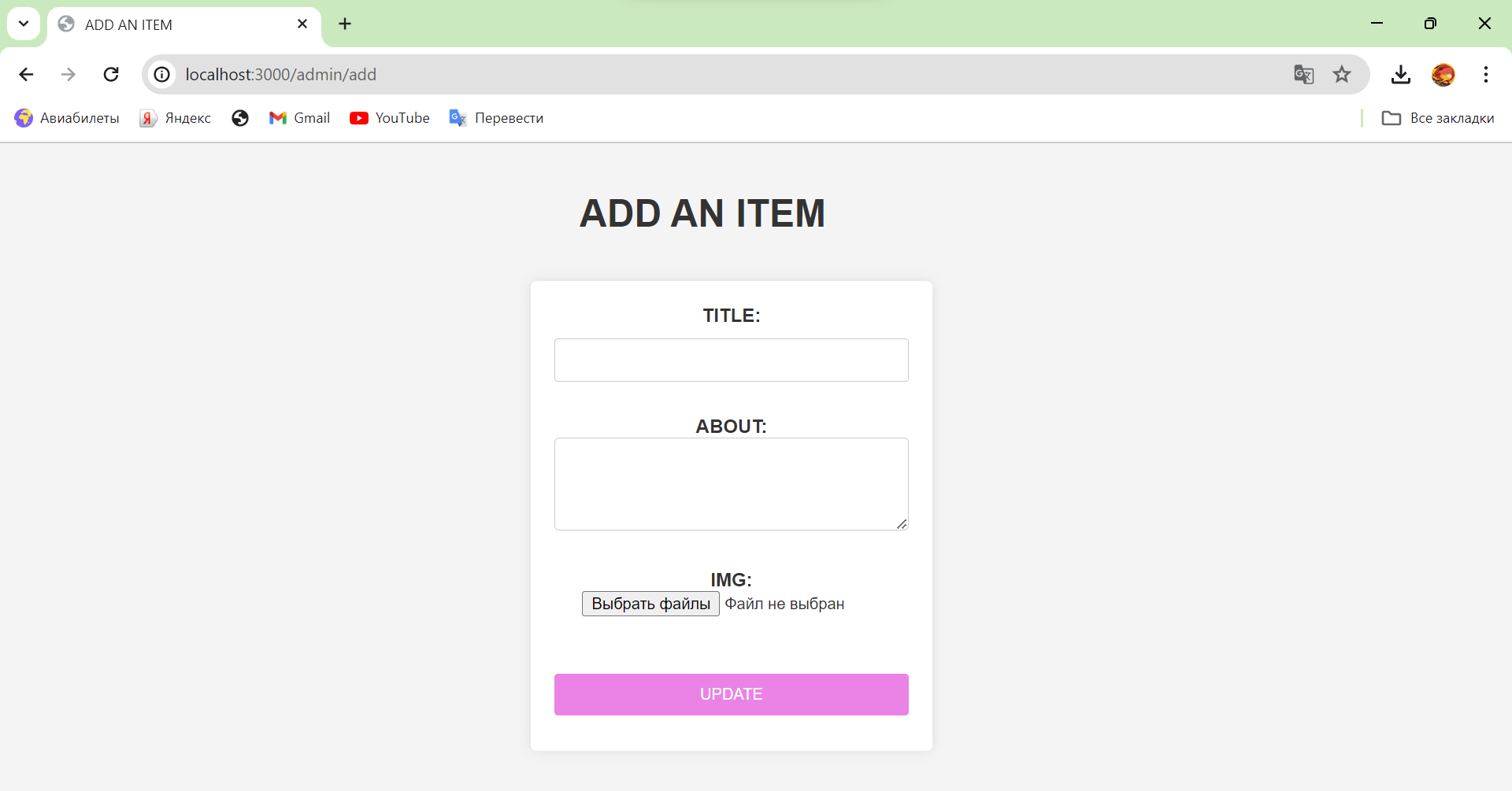
Автоматически созданное описание**

**WEBSITE:**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, диаграмма

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание**