TEDMOB Technical Assessment 2024

Project Overview

Objective

Develop a web application using Next.js for the frontend, Express.js for the backend, and PostgreSQL for the database. The application will integrate Pixabay's API to display and search for images and videos, and include features such as authentication, user profiles, and a favorites system.

Features

1. Home screen that utilizes Pixabay's data.
2. Full integration of Pixabay's APIs:
    - Search
    - Filter
    - Images
    - Videos
3. Authentication and Authorization:
    - JWT for session management.
    - Verification via email using NodeMailer and Mailtrap.
4. User Profiles and Favorites management.
5. Password reset functionality.
6. Deployment to a private GitHub repository.

Deadline

The project must be completed by **June 17, 2024**.

Resources

- Pixabay API Documentation: Pixabay API Docs

Tools and Technologies

Frontend

- **NEXT.js**: For building the user interface.
- **Redux**: For state management (optional, depending on complexity).
- **Axios**: For API requests.

Backend

- **Express.js**: For building the RESTful API.
- **JWT**: For authentication and session management.
- **NodeMailer**: For sending verification and reset password emails.
- **Mailtrap**: For email testing in development.
- **Sequelize**: ORM for PostgreSQL (optional, for easier database interactions).

Database

- **PostgreSQL**: For storing user data, favorites, etc.

Other Tools

- **GitHub**: For version control and collaboration.
- **Postman**: For API testing.
- **VS Code**: Recommended code editor.

Project Architecture

Frontend

- **Components**: Modular and reusable UI components.

- **Pages**: Different routes for the application (Home, Profile, Login, etc.).
- **State Management**: Using React's useState and useEffect hooks, or Redux for more complex state.

Backend

- **Routes**: Define endpoints for different functionalities (authentication, Pixabay data, user profile).
- **Controllers**: Handle the logic for each endpoint.
- **Services**: Interact with external APIs and the database.
- **Middlewares**: For handling authentication, authorization, and error handling.

Database Schema

- **Users**: Table for user information (username, email, password, etc.).
- **Favorites**: Table linking users to their favorite images/videos.
- **Tokens**: Table for storing JWT tokens for sessions.

Implementation Steps

Development

1. **Home Screen**:
    - Fetch and display data from Pixabay API.
    - Implement search and filter functionalities.
2. **Authentication**:
    - Set up JWT authentication for login and signup.
    - Use NodeMailer to send verification emails.
    - Set up Mailtrap for development email testing.
3. **User Profile and Favorites**:

- Create endpoints to manage user profiles.
- Allow users to add/remove favorites.
- Display user's favorites on their profile page.

4. **Password Reset**:
    - Implement functionality to send reset password emails.
    - Create endpoints for resetting the password.

Testing and Deployment

1. **Testing**:
    - Use Postman for API endpoint testing.
    - Write unit and integration tests for critical parts of the application.
2. **Deployment**:
    - Set up a private GitHub repository.
    - Push the code and invite collaborators.
    - Ensure environment variables and secrets are properly managed.

Detailed Task Breakdown

Deployment

1. **GitHub**:
    - Create a private repository.
    - Send Invitation to **ibrahimtarhini01**
2. **Environment Setup**:
    - Manage environment variables for development and production.
    - Ensure secure handling of API keys and secrets.

Conclusion

This document outlines the requirements and steps necessary to develop the web application using the specified technologies. Following this plan will help ensure a structured and efficient development process, leading to a robust and fully functional application.
