CS4032-Web Programming
Assignment 05
BSCS – 6C & 6E

Instructor	Email
Sumaira Mustafa	sumaira.mustafa@nu.edu.pk

Maximum Marks :60

Submission Guidelines:

a. Submit GitHub branch link on the Google Classroom before the deadline.

b. Using Ai tools is allowed, **but you should have complete understanding of every concept used.**

c. Late submissions will not be accepted.

d. Plagiarism will result in zero marks for the assignment.

e. You can use any Frontend technologies of your choice. Again, **you should have a complete understanding of it.**

Project Deliverable 2:

For Deliverable 2 of web programming project, considering the front-end modules already developed and the Express.js backend created, we can focus on expanding the application's functionality.

- **User Profile Module**
  1. Allow users to view and edit their profile information (such as name, email, profile picture, etc.).
  2. Implement features like profile picture upload, bio section, and preferences/settings management.

- **Dashboard Module**:
  1. Create a dashboard where users can view personalized content, statistics, notifications, or any relevant information.
  2. Implement widgets or sections for different types of content or data visualization.

- **Content Management Module**
  1. Extend the content management functionality to include features like creating, editing, and deleting various types of content (e.g., articles, posts, images, videos).

- **Search, pagination and Filtering Module**:
  1. Enhance the search functionality to allow users to search for specific content within the application.

2. Implement advanced filtering options to refine search results based on various criteria.

- **Social Media Integration Module:**

    1. Integrate social media features such as sharing content on social platforms, fetching social media feeds, or enabling social login/authentication such as (Facebook, Gmail, LinkedIn, Github)

- **Admin Panel Module**

    1. Develop an admin panel interface for administrators to manage users, content, settings, and other administrative tasks.
    2. Include features like user management, content moderation, analytics/dashboard for administrators.

Instructions:

User Profile Module:

1. **View/Edit Profile Information:**
    - Create routes in Express.js to handle requests for viewing and editing user profile information.
    - Design front-end pages to display user profile information and provide forms for editing.
    - Implement validation on the backend to ensure data integrity.
2. **Profile Picture Upload, Bio Section, and Preferences/Settings Management:**
    - Allow users to upload profile pictures using a file upload mechanism (such as Multer middleware in Express.js).
    - Create fields in the user database schema to store bio information and user preferences/settings.
    - Implement frontend forms for users to input bio information and manage preferences/settings.

Dashboard Module:

1. **Personalized Content and Statistics:**
    - Retrieve relevant data from the backend based on user preferences or activity history.
    - Design dashboard widgets or sections to display personalized content and statistics using HTML/CSS/JavaScript (any front-end library or framework).
2. **Widgets or Sections for Different Content or Data Visualization:**
    - Implement different sections or widgets to display various types of content or data visualization.
    - Use libraries like Chart.js or D3.js for data visualization if needed.

Content Management Module:

1. **Creating, Editing, and Deleting Content:**
    - Create routes in Express.js to handle CRUD (Create, Read, Update, Delete) operations for content management.
    - Design front-end interfaces for users to create, edit, and delete content.
    - Implement validation and authorization checks on the backend to ensure only authorized users can perform these actions.

Search, Pagination, and Filtering Module:

1. **Enhanced Search Functionality:**
    - Implement full-text search functionality using a search engine library like Elasticsearch or integrating with a database's search capabilities.
    - Design frontend search interfaces with input fields and search buttons.
2. **Pagination and Advanced Filtering:**
    - Implement pagination to display search results in multiple pages.
    - Add advanced filtering options like category filters, date filters, or price range filters using dropdowns or checkboxes.

Social Media Integration Module:

1. **Social Login/Authentications:**
    - Integrate OAuth authentication with social media platforms like Facebook, Google (Gmail), LinkedIn, and GitHub using Passport.js middleware in Express.js.
    - Implement frontend buttons or links for users to log in/register using their social media accounts.
2. **Sharing Content on Social Platforms:**
    - Add social sharing buttons to content pages to allow users to share content on their social media profiles.
    - Implement backend APIs to generate shareable links or embed codes for shared content.

Admin Panel Module:

1. **User Management:**
    - Create admin-specific routes and interfaces for managing users (e.g., view user list, edit user details, delete users).
2. **Content Moderation:**
    - Implement features for admins to moderate user-generated content (e.g., approve or reject posts, flag inappropriate content).
3. **Analytics/Dashboard for Administrators:**
    - Design dashboard widgets or sections to display analytics and statistics relevant to administrators (e.g., user demographics, content engagement metrics).
    - Implement backend logic to retrieve and process data for the admin dashboard.

Ensure to follow the MVC (Model-View-Controller) structure for better organization and maintainability of code. Also, consider security aspects such as input validation, authentication, and authorization throughout the implementation.

Important Notes:

1. In this project, user authentication is facilitated using MongoDB, while other aspects of the project can utilize various relational databases. ORM models are employed to interact with the databases efficiently.
2. During authentication, user history containing login date and time along with their information is captured.

Login Date/Time	Client Information
2018-03-27T19:52:48.000Z	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36
2018-03-27T19:53:04.000Z	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36

3. Hashing passwords.