Full Stack Development with MERN

API Development and Integration Report

Date	7July 2024
Team ID	SWTID1719938419
Project Name	Connectify - Social Media App (MERN)
Maximum Marks	

Project Title: Connectify - Social Media App (MERN)
Date: 18th July 2024
Prepared by: Tharun kumar S

Objective

The objective of this report is to document the API development progress and keyaspects of thebackendservices implementation for the Social Media Application project.

Technologies Used

- **Backend Framework:** Node.js with Express.js .
- **Database:** MongoDB .
- **Authentication:** bcrpet ,body-parser ,JWT.

Project Structure

Provide a screenshot of the backendproject structure with explanations for key directories andfiles.

Key Directories and Files

1. /controllers

a. Contains functionsto handle requestsand responses.

2. /models

b. Includes Mongooseschemas and models for MongoDB collections.

3. /routes

c. Defines the API endpointsand links them to controller functions.

4. /middlewares

d. Custom middleware functions for request processing.

5. /config

e. Configuration files for database connections, environment variables, etc.

API Endpoints

A summaryof the main API endpoints and their purposes:

CreatePostView

POST /api/posts/- This endpoint would typically handle the creation of a new post**.**

GET /api/sidebar-data/- Retrieves data to populate the sidebar with relevant information, such as trending topics, user recommendations, or related posts

PUT /api/user/preferences: Updates user preferences related to the post creation experience, such as default settings, privacy options, or notifications**.**

ExploreView

GET /api/posts/- Retrieves a list of posts based on various criteria such as trending, latest, or recommended posts**.**

POST /api/posts/new/- Allows authenticated users to create new posts.

GET /api/user/preferences/-Retrieves user preferences related to post sorting, filters, or display settings.

POST /api/posts/like/- Allows users to like or interact with a specific post.

GET /api/sidebar-data/- Retrieves data to populate the sidebar with relevant information, such as trending topics, user recommendations, or related posts.

LoginView

POST /api/users/login/-Authenticates a user based on provided credentials (email and password)

POST /api/users/signup/-Registers a new user with the application.

MessengerView

GET/ /api/messages/- Retrieves all the conversations for the logged-in user. This includes details about the recipient and the messages exchanged in each conversation.

GET/api/messages/:id/- Fetches a specific conversation by its ID. This is used to load the messages for a particular chat when a user selects a conversation**.**

POST /**api/messages/-** Sends a new message in a specific conversation. This updates the conversation with the new message.

GET /api/users/:id/- Retrieves the details of a specific user by their ID. This might be used to get information about the recipient of a message or the logged-in user.

AUTH /api/auth/login/- Authenticates a user and returns a token if the credentials are valid. This token is used for subsequent authenticated requests

AUTH /api/auth/register/- Registers a new user and returns a token if the registration is successful.

GET /api/auth/status/- Checks if the current user is logged in and returns their authentication status.

PostView

GET/api/posts/-Retrieves all posts available. This is used to display a list of posts on the platform

GET/api/posts/:id/- Fetches a specific post by its ID. This is used to display the details of a particular post when a user selects it.

CREATE/api/posts/- Creates a new post. This is used when a user submits a new post to the platform.

UPDATE/api/posts/:id/- Updates an existing post by its ID. This is used when a user edits an existing post.

DELETE/api/posts/:id/- Deletes a specific post by its ID. This is used when a user deletes a post from the platform.

GET /api/posts/:id/comments/- Retrieves all comments for a specific post. This is used to display comments associated with a particular post.

CREATE/api/posts/:id/comments/-Creates a new comment for a specific post. This is used when a user adds a comment to a post.

ProfileView

UPDATE/api/posts/:id/- Updates an existing post by its ID. This is used when a user edits an existing post.

DELETE/api/posts/:id/- Deletes a specific post by its ID. This is used when a user deletes a post from the platform.

GET /api/posts/:id/comments/- Retrieves all comments for a specific post. This is used to display comments associated with a particular post.

CREATE/api/posts/:id/comments/-Creates a new comment for a specific post. This is used when a user adds a comment to a post.

AUTH /api/auth/login/- Authenticates a user and returns a token if the credentials are valid. This token is used for subsequent authenticated requests

AUTH /api/auth/register/- Registers a new user and returns a token if the registration is successful.

SearchView

GET /api/posts/-Retrieves all posts available. This is used to display a list of posts on the platform.

GET/api/posts/:id/-Fetches a specific post by its ID. This is used to display the details of a particular post when a user selects it.

CREATE/api/posts/-Creates a new post. This is used when a user submits a new post to the platform.

UPDATE /api/posts/:id/- Updates an existing post by its ID. This is used when a user edits an existing post.

DELETE /api/posts/:id/- Deletes a specific post by its ID. This is used when a user deletes a post from the platform.

GET/api/posts/:id/comment/-Retrieves all comments for a specific post. This is used to display comments associated with a particular post.

CREATE /api/posts/:id/comments/-Creates a new comment for a specific post. This is used when a user adds a comment to a post.

AUTH_LOGIN/api/auth/login/-Authenticates a user and returns a token if the credentials are valid. This token is used for subsequent authenticated requests.

AUTH/api/auth/register/-Registers a new user and returns a token if the registration is successful.

GET/api/auth/status/- Checks if the current user is logged in and returns their authentication status.

SignupView

GET/api/posts/- Retrieves a list of all posts available on the platform**.**

GET/api/posts/:id/- Fetches a specific post by its ID, used to display the details of a selected post.

CREATE /api/posts/-Creates a new post, allowing users to submit new content.

UPDATE/api/posts/:id/- Updates an existing post by its ID, allowing users to edit their submitted content.

DELETE/api/posts/:id/- Deletes a specific post by its ID, allowing users to remove their content

Integration with Frontend

The backendcommunicates with the frontend via RESTful APIs. Key pointsof integrationinclude:

- **User Authentication:** Tokens are passed betweenfrontend and backendto handle authentication.
- **Data Fetching:** Frontend components make API callsto fetch necessary data for displayand interaction.

Error Handling and Validation

Describe the error handling strategy and validation mechanisms:

- **Error Handling:** Error handlingusing Try catchblocks.
- **Validation:** Input validation using libraries like JWT web token or express-validator.

Security Considerations

Outline the security measuresimplemented:

- **Authentication:** Secure token-based authentication.
- **Data Encryption:** Encrypt sensitive data at rest and in transit.