# API Documentation Outline

This document provides details for each endpoint in the authentication and authorization cycle. It includes the endpoint, request body, and a step-by-step guide on how to use each API.

## Register User

Endpoint: POST /register

Description: Registers a new user by saving their username, email, and hashed password in the database.

Request Body (JSON):

* username: "your\_username"
* email: "your\_email@example.com"
* password: "your\_password"

Expected Response:

Status: 201 Created

Body (JSON):

* message: "User registered successfully"

Error Response:

Status: 400 Bad Request

Body (JSON):

* error: "Detailed error message here"

## Login User

Endpoint: POST /login

Description: Authenticates the user and provides a JWT token for access to protected routes.

Request Body (JSON):

* email: "your\_email@example.com"
* password: "your\_password"

Expected Response:

Status: 200 OK

Body (JSON):

* token: "JWT\_TOKEN\_HERE"

Error Response:

Status: 401 Unauthorized

Body (JSON):

* message: "Invalid credentials"

## Access Protected Route

Endpoint: GET /protected

Description: Access a protected route that requires a valid JWT token.

Request Headers:

* Authorization: Bearer JWT\_TOKEN\_HERE

Expected Response:

Status: 200 OK

Body (JSON):

* message: "This is a protected route"
* user:
* id: "USER\_ID\_HERE"

Error Response:

Status: 401 Unauthorized

Body (JSON):

* message: "No token provided, authorization denied"

Status: 401 Unauthorized

Body (JSON):

* message: "Token is not valid"

## Create Board

Endpoint: POST /accounts/:accountId/boards

Description: Creates a new board for a specific account.

Request Body (JSON):

* title: "Board Title"
* description: "Board Description"

Request Parameters:

* accountId: The ID of the account to which the board will be added.

Expected Response:

Status: 201 Created

Body (JSON):

* \_id: "BOARD\_ID"
* title: "Board Title"
* description: "Board Description"

Error Response:

Status: 404 Not Found

Body (JSON):

* message: "Account not found"

## Create Section

Endpoint: POST /boards/:boardId/sections

Description: Creates a new section in a specific board.

Request Body (JSON):

* title: "Section Title"

Request Parameters:

* boardId: The ID of the board to which the section will be added.

Expected Response:

Status: 201 Created

Body (JSON):

* \_id: "SECTION\_ID"
* title: "Section Title"

Error Response:

Status: 404 Not Found

Body (JSON):

* message: "Board not found"

## Create Card

Endpoint: POST /sections/:sectionId/cards

Description: Creates a new card in a specific section.

Request Body (JSON):

* title: "Card Title"
* description: "Card Description"

Request Parameters:

* sectionId: The ID of the section to which the card will be added.

Expected Response:

Status: 201 Created

Body (JSON):

* \_id: "CARD\_ID"
* title: "Card Title"
* description: "Card Description"

Error Response:

Status: 404 Not Found

Body (JSON):

* message: "Section not found"

## Step-by-Step Usage

1. Register a New User:
2. Send a POST request to /register with the user details.
3. Log in the User:
4. Send a POST request to /login with the user's email and password.
5. Receive a JWT token in the response.
6. Access a Protected Route:
7. Send a GET request to /protected with the Authorization header containing the JWT token.
8. Create a Board:
9. Send a POST request to /accounts/:accountId/boards with the board details and the account ID.
10. Create a Section:
11. Send a POST request to /boards/:boardId/sections with the section details and the board ID.
12. Create a Card:
13. Send a POST request to /sections/:sectionId/cards with the card details and the section ID.

By following these steps, you can test each endpoint to ensure that the authentication, board, section, and card management features are working correctly.