**Mock-up project Documentation**

This mock-up project provides APIs for user authentication, adding cards, and listing boards. The APIs are implemented using the express.js framework and mongoose for MongoDB database modeling and querying.

### **Installation**

To install the project, clone the repository from GitHub and run npm install to install the dependencies:

*git clone https://github.com/myusername/mock-up-project.git*

*cd mock-up-project*

*npm install*

### **Configuration**

Before running the project, you need to configure the database connection and authentication settings. Create a .env file in the root directory of the project and add the following environment variables:

*DB\_URI=mongodb://localhost:27017/mock-up-project*

*JWT\_SECRET=mysecretkey*

*EMAIL\_USERNAME=myemail@example.com*

*EMAIL\_PASSWORD=mypassword*

Replace mongodb://localhost:27017/mock-up-project with your MongoDB database URI, mysecretkey with a secret key for JWT authentication, and myemail@example.com and mypassword with your email credentials for password reset emails.

### **Usage**

To start the project, run npm start:

*npm start*

The project will start running on http://localhost:3000.

User authentication APIs

The project provides the following APIs for user authentication:

### **Sign up**

POST /api/signup

Creates a new user account.

Request body:

{

"name": "John Doe",

"email": "johndoe@example.com",

"password": "mypassword"

}

Response body:

{ "user": { "name": "John Doe", "email": "johndoe@example.com" }, "token": "..." }

**Login**

POST /api/login

Logs in an existing user.

Request body:

{ "email": "johndoe@example.com", "password": "mypassword" }

Response body:

{ "user": { "name": "John Doe", "email": "johndoe@example.com" }, "token": "..." }

**Forgot password**

POST /api/forgot-password

Sends a password reset email to the user.

Request body:

{ "email": "johndoe@example.com" }

**Card adding APIs**

The project provides the following APIs for adding cards:

**Add card**

POST /api/cards

Adds a new card to the database.

Request body:

{ "title": "Card title", "description": "Card description" }

Response body:

{ "title": "Card title", "description": "Card description" }

**Board listing APIs**

The project provides the following APIs for listing boards:

**List boards**

GET /api/boards

Lists all the boards in the database.

Response body:

[ { "title": "Board title", "cards": [ { "title": "Card title", "description": "Card description" } ] } ]

**Data schema**

The project uses the following data schema for MongoDB:

**User schema**

{ name: String, email: String, password: String }

**Card schema**

{ title: String, description: String, projectName: String }