

# Assignment: Setting Up Database Connection, Models, Migrations, and Authentication Controllers

## Objective:

In this assignment, you will set up the database connection, define models, create migrations, and implement the controllers to handle user authentication (Sign Up and Login). This will ensure that user data is stored and retrieved securely in the database.

---

## Tasks:

### 1. Set Up Database Connection

- Inside your **backend** folder, create a file called **config/database.js**.
- Use **Sequelize** to establish a connection to your PostgreSQL database.
- Configure Sequelize to read database credentials from **.env** file.
- Export the configured Sequelize instance.

### 2. Create a User Model

- Inside the **models** folder, create a file called **User.ts**.
- Define a **User** model with the following attributes:
  - **id** (Primary Key, UUID, Auto-generated)
  - **fullName** (String, Required)
  - **email** (String, Required, Unique)
  - **password** (String, Required)
  - **createdAt** (Timestamp, Default to Now)
  - **updatedAt** (Timestamp, Default to Now)
- Sync the model with the database.

### 3. Create a Migration File

- Inside the **migrations** folder, create a migration file.
- Use Sequelize CLI to generate the migration file.
- Define the same schema as the **User model**.
- Run the migration to create the **Users** table in the database.

### 4. Set Up User Authentication in Controllers

- Inside the **controllers** folder, create a new file called **authController.ts**.
- Implement the following functions:

- **signup:**
  - Hash the password before saving.
  - Save the user in the database.
  - Return a success message or an error.
- **login:**
  - Check if the user exists.
  - Compare the hashed password.
  - Generate a JWT token on successful login.
  - Return the token and user data.

## 5. Update the Routes

- In the **routes** folder, create a new file **authRoutes.js**.
- Define routes for:
  - **POST /signup** → Calls the signup function.
  - **POST /login** → Calls the login function.
- Import and use this router in **server.js** or **index.js**.

## Submission Requirements:

- Push your changes to your GitHub repository.
- Share the repository link.
- Test all API endpoints using **Postman or Thunder Client**.
- Ensure your database connection and migrations work correctly.

---

## Bonus Task (Optional):

- Implement JWT authentication middleware.
- Add validation for email and password.

## Deadline:

Submit your completed work by **[1:00 PM March 7, 2025]**.

---

This assignment will help you understand how to connect a Node.js backend to a PostgreSQL database using Sequelize, define models and migrations, and handle authentication securely.

Good luck! 🚀