**Assignment Submission System**

This system is a backend application built using maven, Spring Boot and MongoDB. It allows users (students) to submit assignments and admins to accept or reject them.

**Features :**

* **Users** can:
  + Register and log in.
  + Upload assignments.
* **Admins** can:
  + Register and log in.
  + View assignments.
  + Accept or reject assignments.

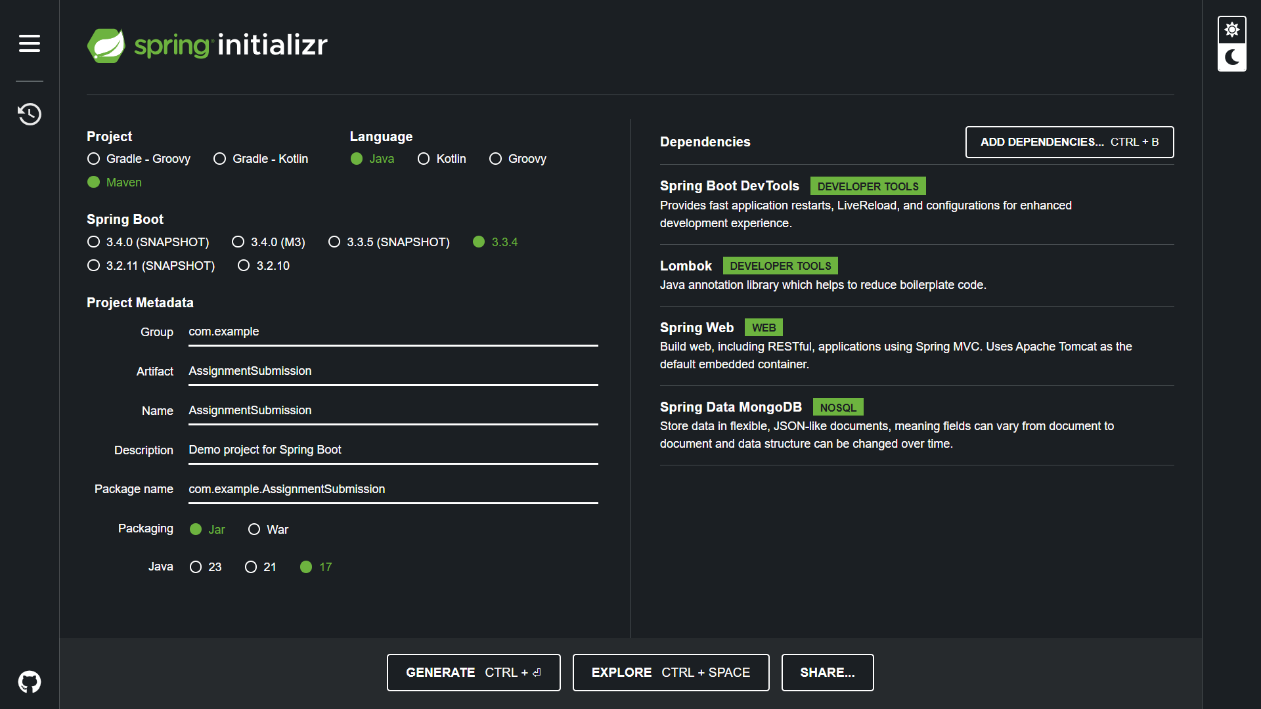
**Prerequisites :**

To run this system locally, you need to have the following software installed:

* **Java 17** (or higher)
* **Maven 3.6** (or higher)
* **MongoDB** (either local or using MongoDB Atlas for cloud hosting)

**Add Spring Dependencies :**

Add following dependencies and click on “GENERATE”

****

**Configure MongoDB**

The application uses MongoDB for data storage. You can set up MongoDB either locally or use a cloud service like Mongodb Atlas.

* **Local MongoDB**: Make sure MongoDB is running on your local machine on the default port 27017 or use custom port number.
* **MongoDB Atlas**: If you're using MongoDB Atlas, you will need the connection string for your MongoDB cluster.

**Set Up Environment Variables**

Create a .env file (or directly edit application.properties) to configure the connection to MongoDB. The connection string will be placed in the application.properties file under the src/main/resources folder.

**For application.properties:**

spring.data.mongodb.uri=mongodb+srv://<username>:<password>@<cluster-url>/<dbname>?retryWrites=true&w=majority

Replace <username>, <password>, <cluster-url>, and <dbname> with your MongoDB credentials.

**Build the Project**

Run the following command to build the project using Maven:

mvn clean install

This will download all dependencies and compile the project.

**Run the Application**

Once the build is complete, you can run the Spring Boot application by executing:

mvn spring-boot:run

**Access the Application**

The application will start running at http://localhost:8080. You can interact with the API using tools like **Postman**.

**API Endpoints**

**Authentication**

**User Registration :**

POST /register/user

**Request Body**:

json

{

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

"password": "password123",

"name": "User Name"

}

**Admin Registration :**

POST /register/admin

**Request Body**:

json

{

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

"password": "adminpassword",

"name": "Admin Name"

}

**User/Admin Login :**

POST /login

**Request Body**:

json

{

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

"password": "password123"

}

**Upload Assignment (User) :**

POST /assignments/upload

**Request Body**:

json

{

"task": "Math Homework",

"admin": "adminName",

}

**Accept Assignment (Admin) :**

POST /assignments/{id}/accept

**Example**:

POST /assignments/12345/accept

**Reject Assignment (Admin) :**

POST /assignments/{id}/reject

**Example**:

POST /assignments/12345/reject

**View All Assignments (Admin) :**

GET /assignments

**Database Structure**

The system uses three collections in MongoDB:

1. **Users**:
   * Fields: id, name, email, password
2. **Admins**:
   * Fields: id, name, email, password
3. **Assignments**:
   * Fields: id, title, admin, status

**Testing**

You can use **Postman** to test the endpoints.

**Example to Accept an Assignment:**

-X POST http://localhost:8080/assignments/12345/accept

**Example to Reject an Assignment:**

-X POST http://localhost:8080/assignments/12345/reject