

# Wishlist Management Application

## Introduction:-

This is a Wishlist Management REST API built with Spring Boot. It allows users to manage their wishlists by adding and deleting items.

## Prerequisites:-

Java 17  
MySQL  
Maven  
Postman (for testing API endpoints)

## Setup & Installation

### 1.Clone the repository:-

```
git clone https://github.com/itsmenitesh/wishlist_management.git
```

### 2.Update application properties:-

```
spring.datasource.url=jdbc:mysql://localhost:3306/databaseName  
spring.datasource.username=username  
spring.datasource.password=password  
spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver  
spring.jpa.hibernate.ddl-auto=update  
  
spring.jpa.properties.hibernate.show_sql=true  
spring.jpa.properties.hibernate.use_sql_comments=true  
spring.jpa.properties.hibernate.format_sql=true
```

### 3.Build the application:-

```
mvn clean install
```

### Run the application:-

After the project is built successfully, you can run the Spring Boot application. Usually, Spring Boot applications have a main class annotated with `@SpringBootApplication`. You can run this class directly from your IDE, or by using the Maven : `mvn spring-boot:run`

The application will start and run on: <http://localhost:8080>

### Running Tests:-

To run the unit tests, use the following command: `mvn test`

### API Endpoints:-

You can use Postman to test the API endpoints. Here are some examples:

**Register a new user:-** POST <http://localhost:8080/register>

Body (raw JSON):

```
{  
  "firstName": "firstName",  
  "lastName": "lastName",  
  "username": "username",  
  "password": "password"  
}
```

**Authenticate a user:-** POST <http://localhost:8080/login>

Body (raw JSON):

```
{  
  "username": "username",  
  "password": "password"  
}
```

Note: after hitting this you will get a JWT token you have to use that JWT token to do other tasks.

Add an item to a user's wishlist: POST <http://localhost:8080/api/wishlist>

Body (raw JSON):

```
{  
  "itemName": "New Wishlist Item"  
}
```

Note: You need to include the JWT token in the Authorization header for this request.

**Get a user's wishlist:-** GET <http://localhost:8080/api/wishlist>

Note: You need to include the JWT token in the Authorization header for this request.

**Delete a item from user's wishlist by id:-** DELETE <http://localhost:8080/api/wishlist/{id}>

## **Deployment**

When deploying this application, consider the following:

Secure your application by using HTTPS.

Use a separate MySQL instance for production with secure credentials.

Consider using a cloud provider like AWS, Google Cloud, or Azure for hosting your application.