Step 7: Deployment Options

In this phase, we'll discuss some of the options available to deploy your Spring Boot REST API.

Deployment Options

- **Standalone Jar File**
  - Spring Boot applications can be packaged as standalone Jar files. You can run them using the **java -jar** command. This is useful for testing or staging environments.
- **Docker Container**
  - You can containerize your application using Docker, which makes it easier to deploy across various environments.
- **Cloud Services**
  - Platforms like AWS, Azure, and Heroku offer managed services for Java applications, providing scalability and ease of deployment.
- **Web Server**
  - You can also deploy your application to a traditional web server, although this is less common for Spring Boot applications.

Documentation

Update your **README.md**:

markdown

Copy code

```
## Step 7: Deployment Options

Several options exist for deploying our Spring Boot REST API:

1. Standalone Jar File: Useful for staging and testing.
2. Docker Container: Offers ease of deployment across environments.
3. Cloud Services: Managed services on platforms like AWS, Azure, and
Heroku.
4. Web Server: Less common for Spring Boot, but still an option.
```

Commit Changes

- No code changes to commit here, but you may want to commit your updated README.

Conclusion

You've now completed the entire cycle from project setup to development, testing, and discussing deployment options. You've built a RESTful API using Spring Boot, connected it to an SQL database, and prepared it for the next stages, which could include more advanced features, user authentication, and finally, deployment.