**CIS 634 / EEC 521**

**Doctor Appointment Management System**

**DEVELOPERS GUIDE**

**Team:**

* Mayank Bhati - 2811695
* Sravani Kovvuru - 2822999
* Havilah Pramodini Kopuri - 2862656

**Abstract**

This guide helps understand developers about the Doctor’s Appointment Management System that manages data for appointment and consultation of patients with doctors.

This system aims to reduce the time of doctor’s consultation with a vision to include pharmacies and tests.

This is a MVC and API based system with Java framework Spring-boot with Eclipse IDE including Thymeleaf UI.

Database used is MySQL(a relational database).

**Terminologies or Keywords:**

API(Application Programmable Interface)

MVC(Model-View-Controller)

DBMS(Database Management System)

**Language specification: -**

**User Interface(Front-end):**

Thymeleaf UI including HTML5,CSS3 & JavaSscript

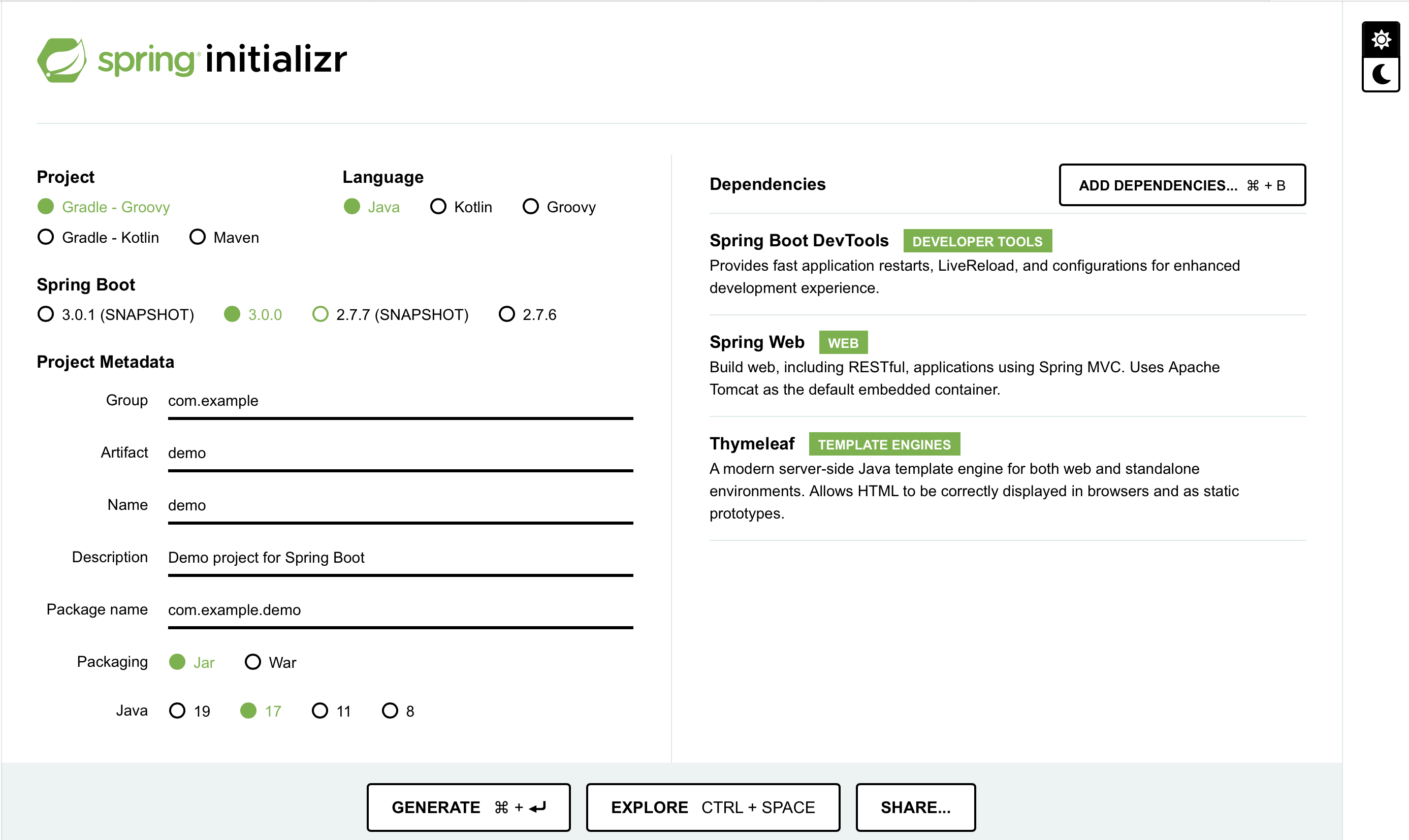
**Developer Interface(Back-end):**

Spring-Boot(Java - version 11), Apache Tomcat, Maven, Linux(distro: Ubuntu) OS

**Database:**

MySQL

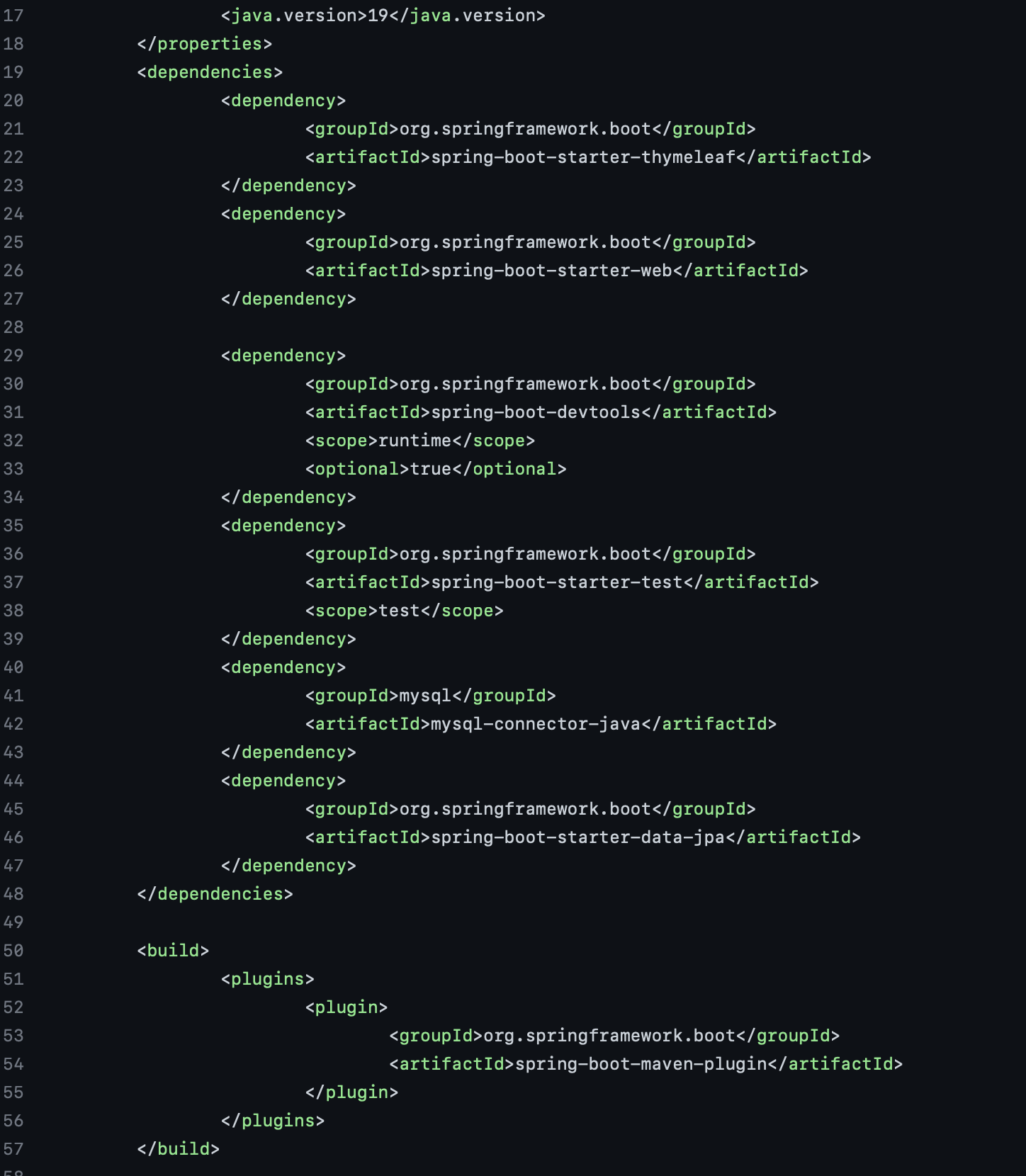
**Configuration:**

**Spring & Dependencies:**

**POM.xml file**

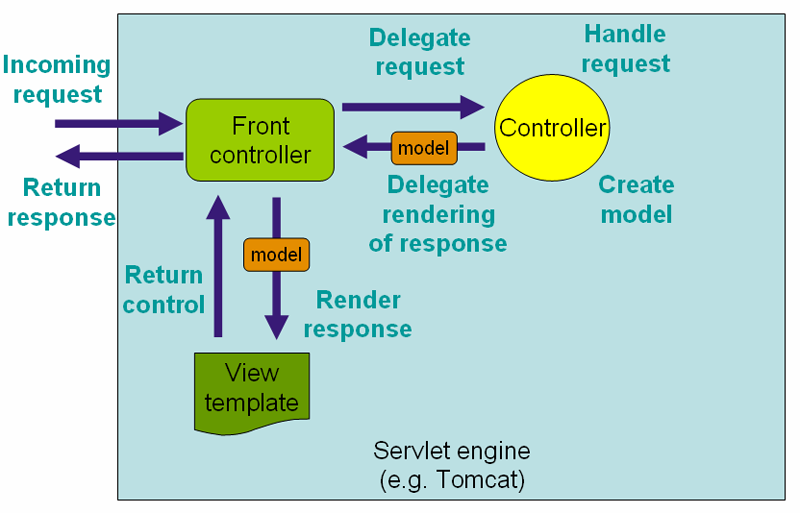
**POM**(**Project Object Model)** or pom.xml file is the main configuration file for spring framework that is being used by web-server to boot the application.

This file includes all the decencies of the spring along with their artifactID and versions. Every time the application boots up, it checks for the pom.xml dependencies.

****

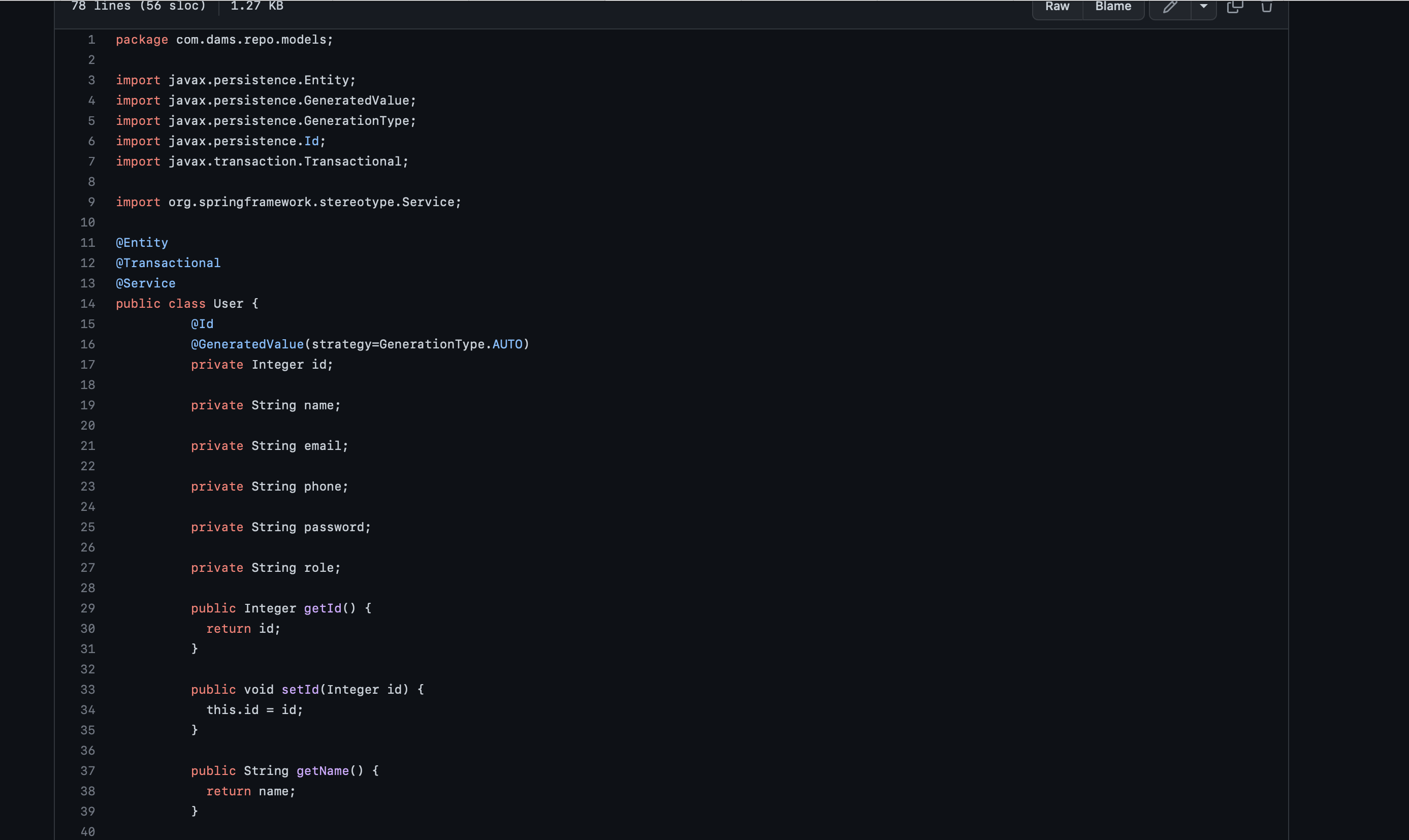
**Object Oriented Model:**

We have used Object Oriented Model to develop this application:



**Application Structure:**

**Login**

**Structure:**

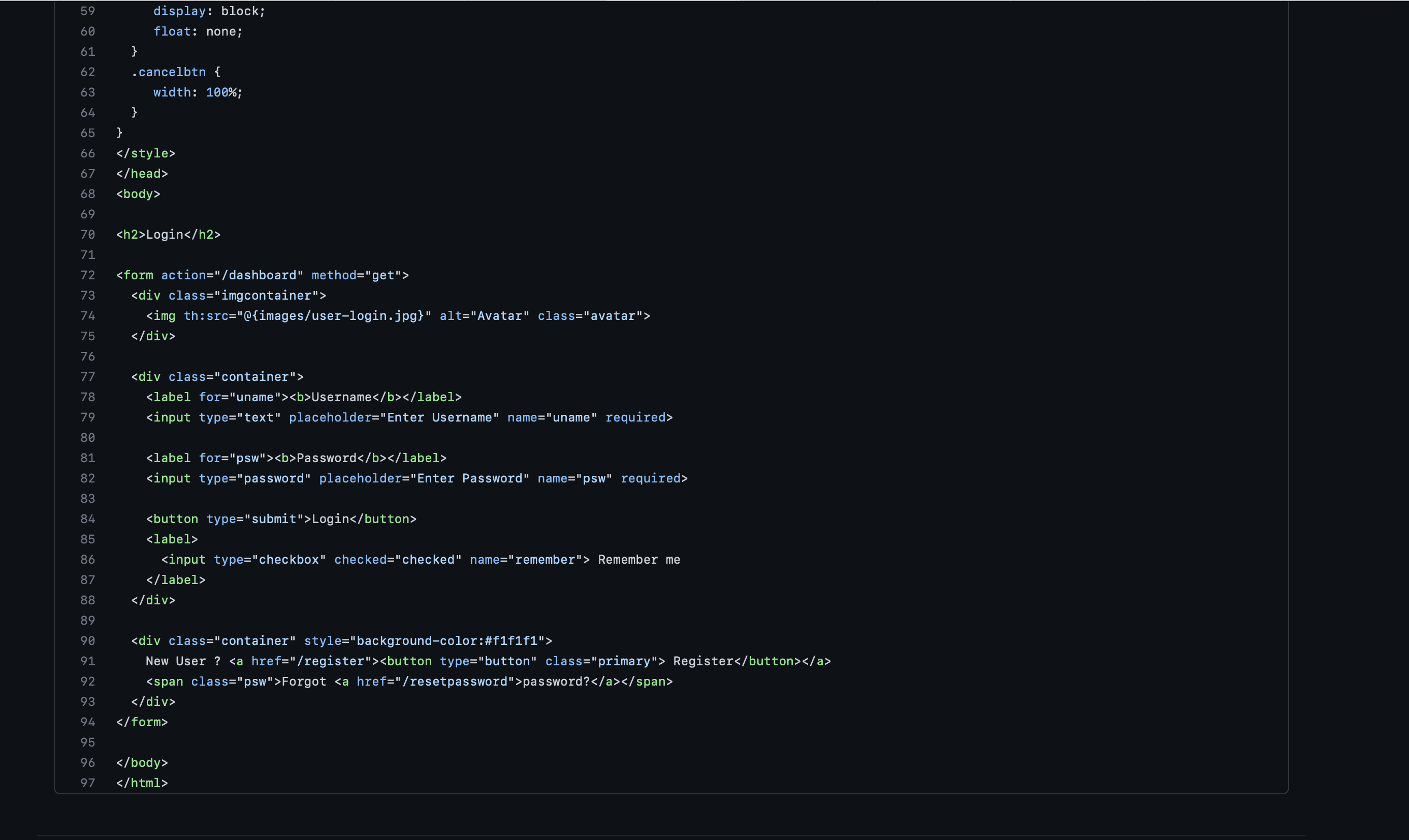
Model: User.java

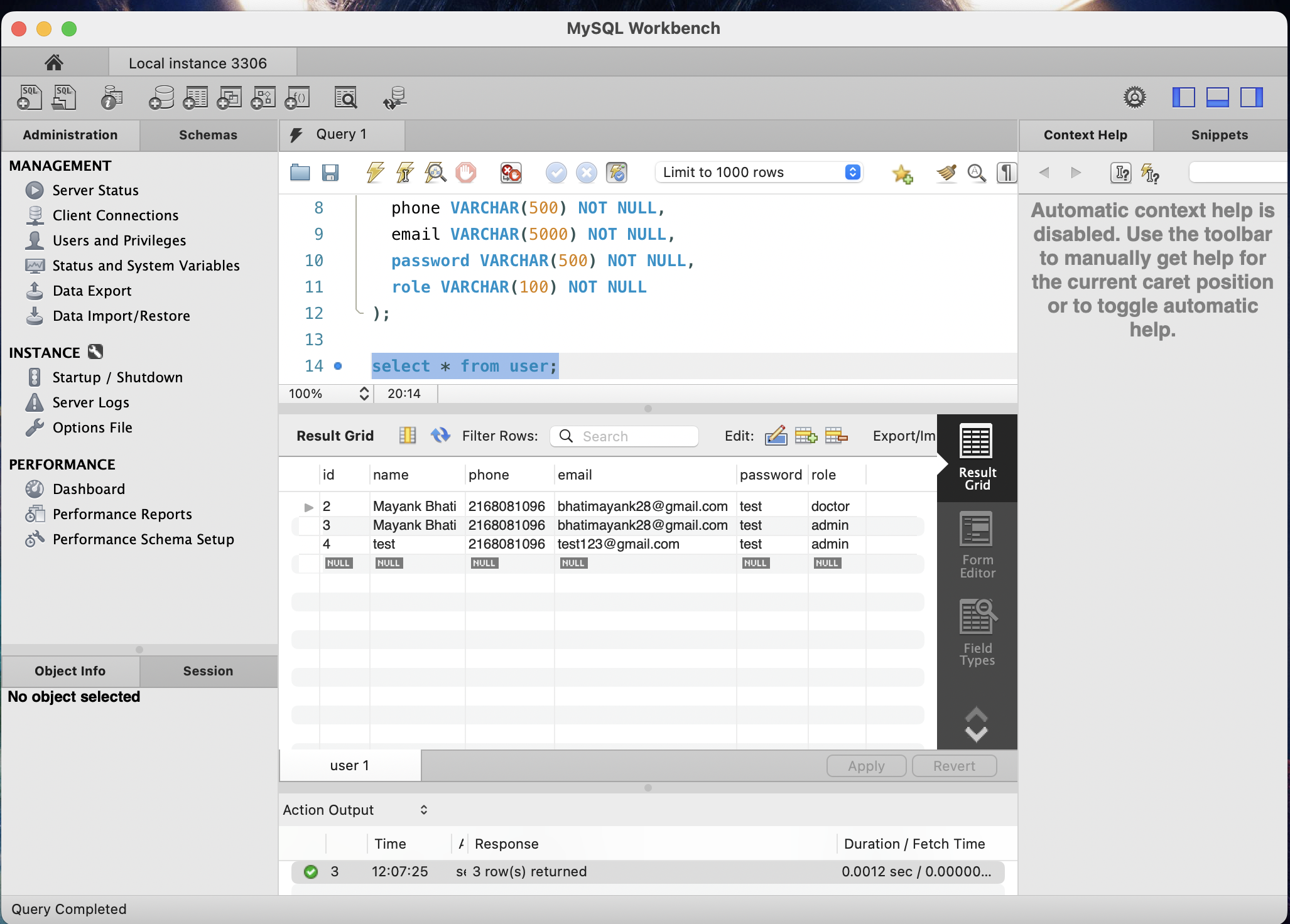
Controller: DamsController.java

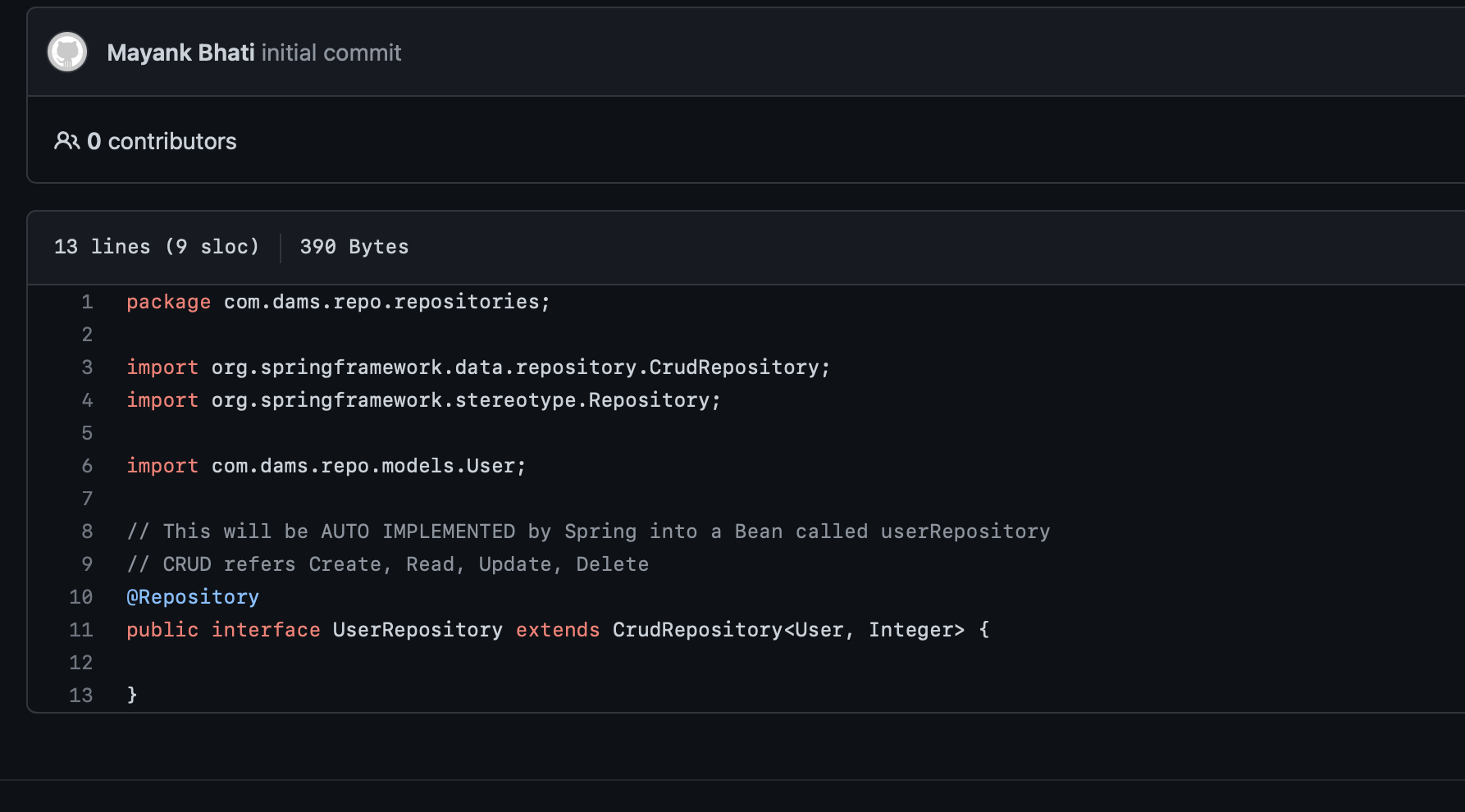
View: login.html

**Flow:**

Login Page -> Enter Credentials -> check for authentication -> return

Login.html using the Thymeleaf template:

**User Table:**

**UserRepository.java** handles all the CRUD operations:

Spring uses Hibernate ORM to map the database.The flow will be as follows



ORM: Object Relation Model

JPA: Java Persistence API that works for data persistence and Java.persistence package contains the JPA classes and interfaces.

**Dashboard:**

Once the credentials is valid, user will be redirected to the dashboard as per the role of them.

We have different dashboard for different roles. Administrator will have access to all the dashboard and can add doctor and Patients from here.

