Object Oriented Analysis and Design using Java
Self-Learning: MVC Framework
Lab Week 9

Vanshika Goel	PES1UG20CS484	Section H	Roll No 40

1. MVC Architecture Pattern

The Model-View-Controller (MVC) pattern is an architectural design pattern that separates an application into three interconnected parts, namely Model, View, and Controller. Each of these components has its own unique responsibilities, as follows:

- Model: This represents the data and business logic of the application. The model is responsible for managing the application data and providing the services that manipulate this data.
- View: This represents the user interface of the application. The view is responsible for presenting the data to the user in a meaningful way and enabling user interaction with the application.
- Controller: This represents the application logic that mediates between the model and the view. The controller is responsible for receiving and processing user input, updating the model, and updating the view.

The MVC architecture pattern is widely used in web development, where the model represents the data stored in the database, the view represents the web page displayed to the user, and the controller represents the server-side code that handles user requests and updates the model and view accordingly.

2. Advantages of MVC Pattern

The MVC pattern has several advantages that make it a popular choice for software development, including:

- Separation of concerns: The MVC pattern separates the application logic into three distinct parts, which makes it easier to develop, test, and maintain the code.
- Reusability: The MVC pattern promotes reusability of code, as each component can be used in multiple applications or projects.
- Scalability: The MVC pattern allows for easy scalability of the application, as each component can be modified or replaced without affecting the others.
- Testability: The MVC pattern makes it easier to test the application, as each component can be tested independently.

3. Features of Spring MVC Framework

Spring MVC is a popular framework for building web applications based on the MVC architecture pattern. It provides several features that simplify the development of web applications, including:

- DispatcherServlet: This is the core of the Spring MVC framework, which receives requests and sends them to the appropriate controller for processing.
- Controllers: These are responsible for processing user requests and returning a response. Controllers are Java classes that handle user input and interact with the model and view.
- Model: This represents the application data and business logic. The model is typically implemented using JavaBeans or POJOs (Plain Old Java Objects).
- View: This represents the user interface of the application. Spring MVC supports several view technologies, including JSP (JavaServer Pages), Thymeleaf, and Velocity.
- HandlerMapping: This is responsible for mapping incoming requests to the appropriate controller.
- Interceptors: These are used to intercept requests and perform pre-processing or post-processing tasks.
- Form handling: Spring MVC provides several features for handling HTML forms, including form validation, data binding, and form submission.
- Integration with other Spring modules: Spring MVC can be easily integrated with other Spring modules, such as Spring Security, Spring Data, and Spring Boot.

In conclusion, the MVC architecture pattern is a widely used design pattern that promotes separation of concerns, code organization, and reusability. The Spring MVC framework provides several features that simplify the development of web applications based on the MVC architecture pattern. Spring MVC is a popular choice for building web applications due to its ease of use, scalability, and integration with other Spring modules.

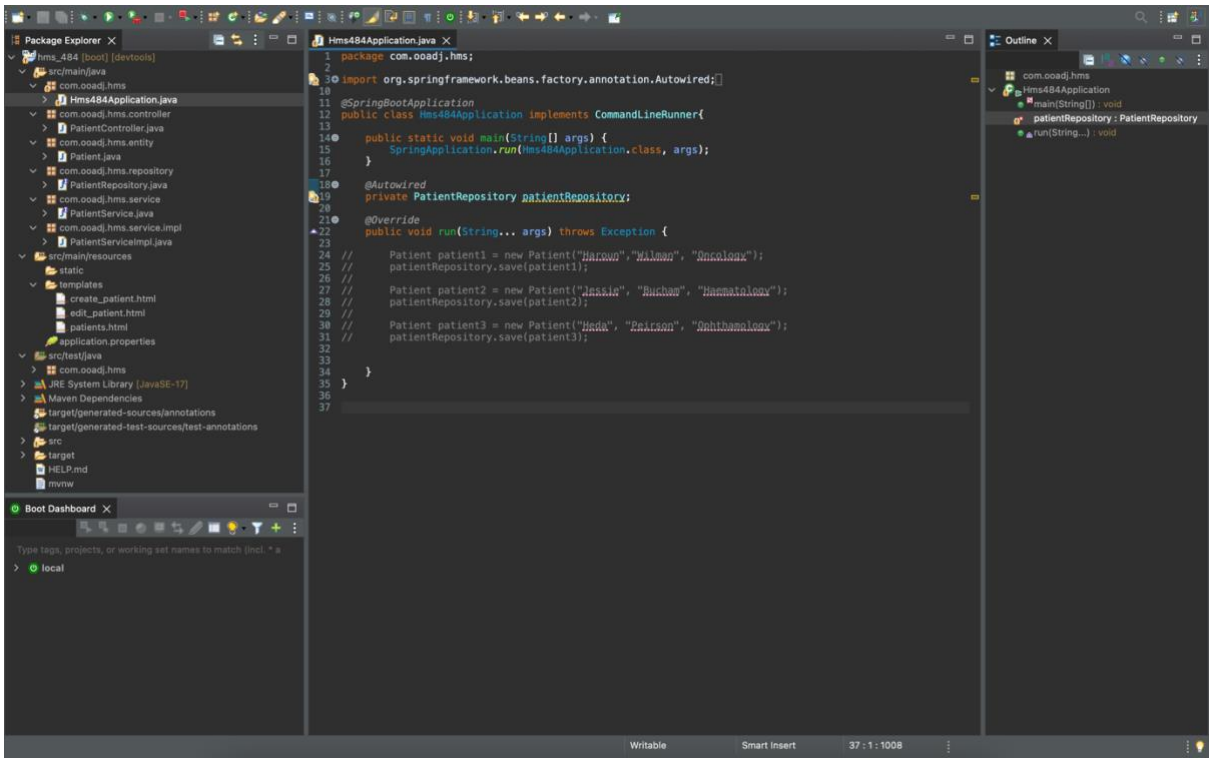
4. Problem definition with description of the chosen scenarios

The application I chose is a Hospital Management System, specifically a Patient Management System.

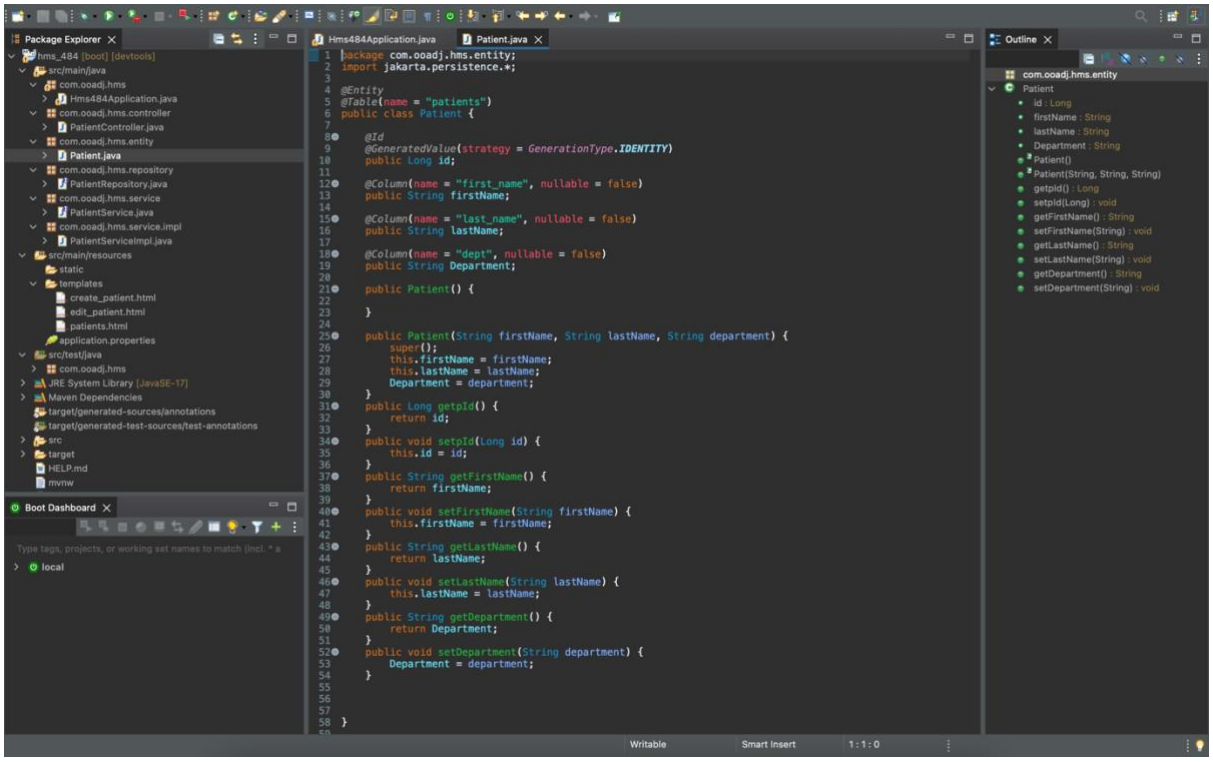
The application I built using the Spring framework, Java as well HTML allows CRUD operations on a database on MySQL.

5. Complete screenshot of the code

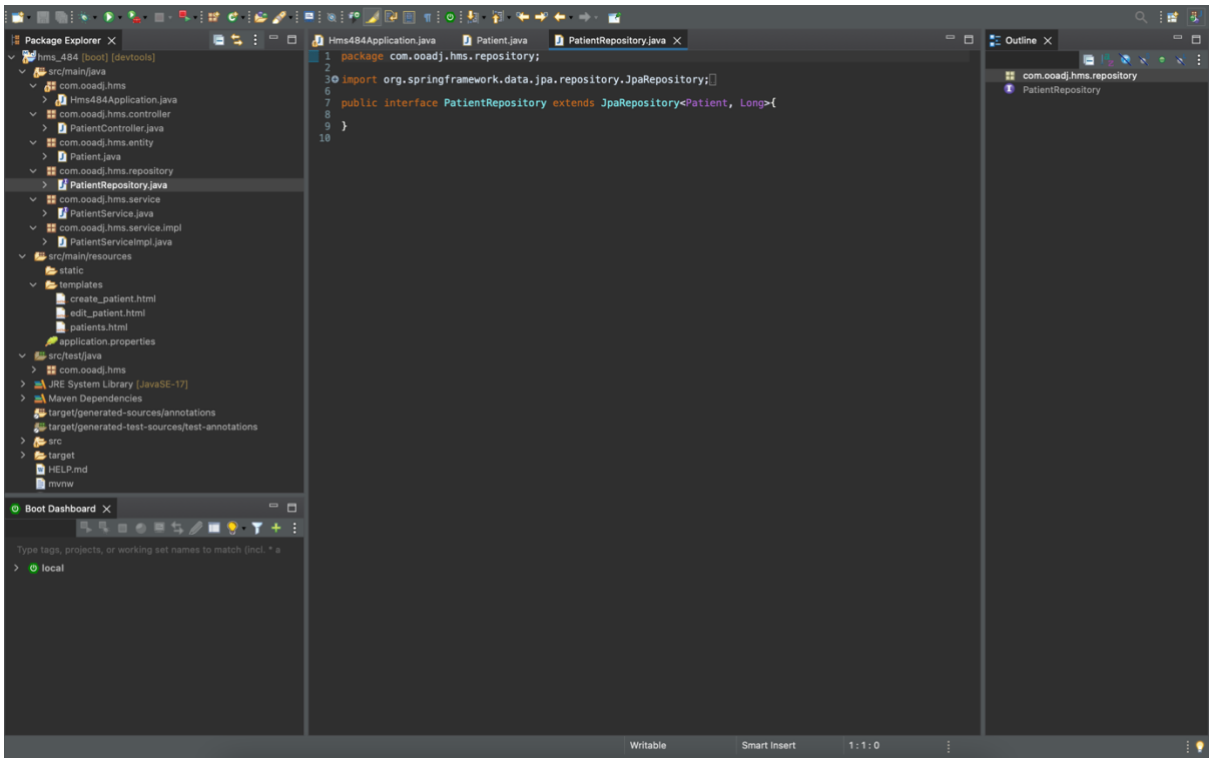
Hms484Application.java



Patient.java

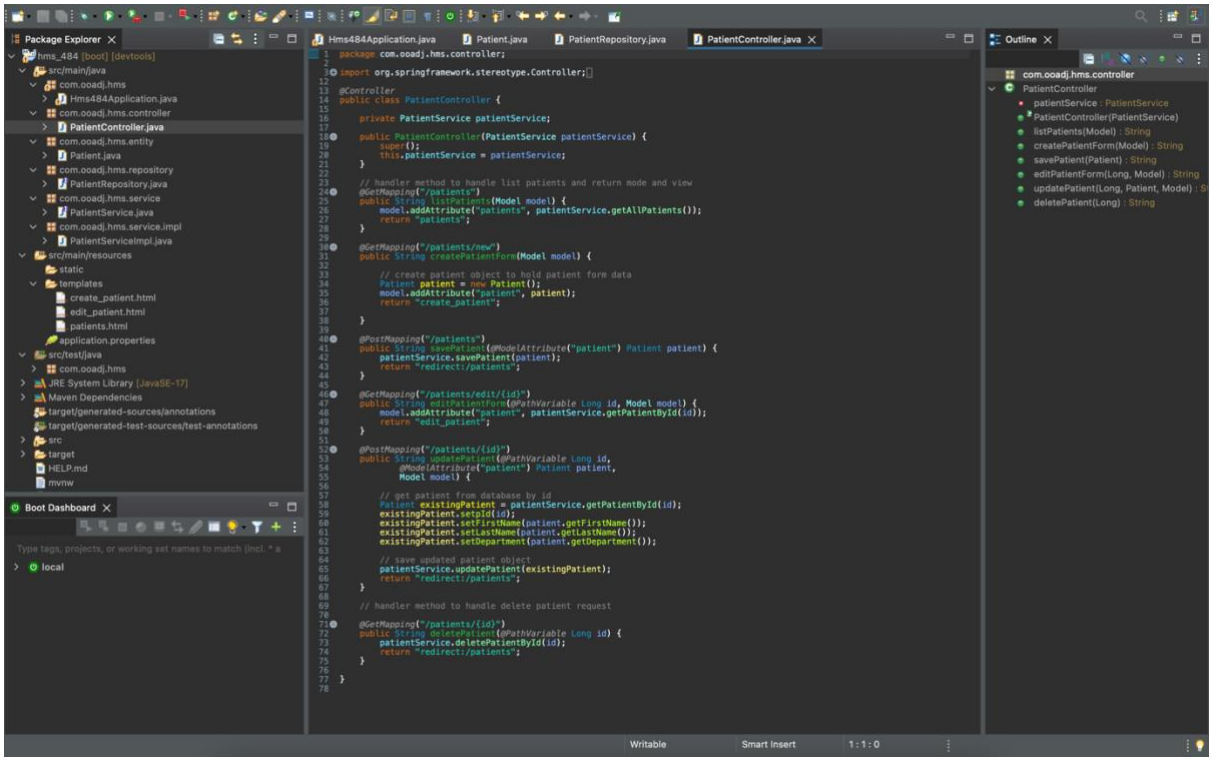


PatientRepository.java



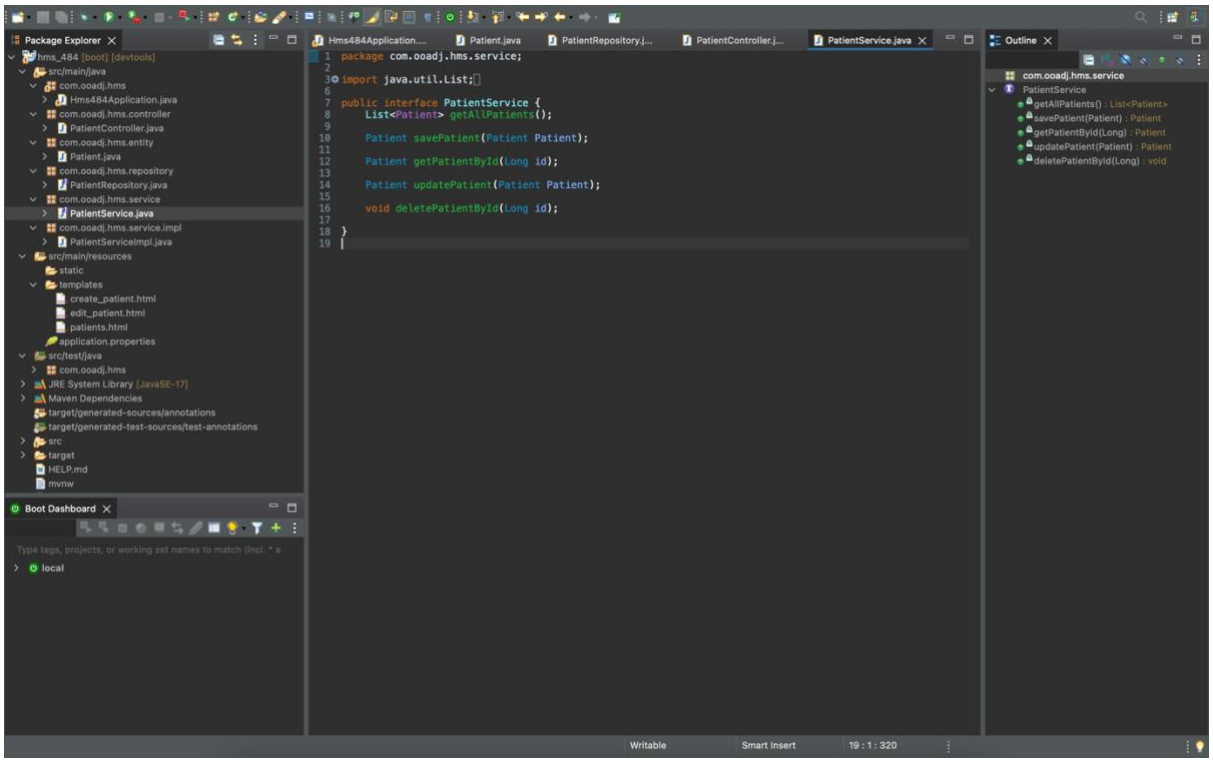

```java
package com.ooadj.hms.repository;

import org.springframework.data.jpa.repository.JpaRepository;

public interface PatientRepository extends JpaRepository<Patient, Long>{

}
```

PatientController.java

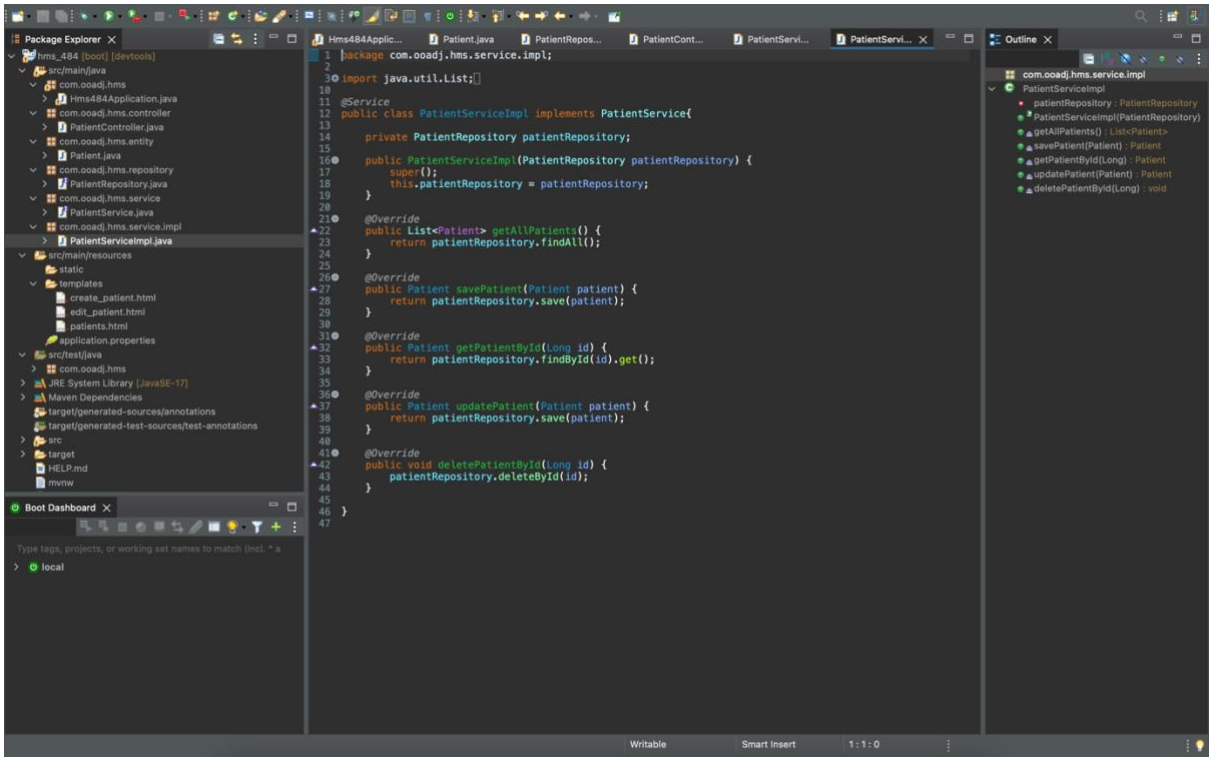



```java
package com.ooadj.hms.controller;

import org.springframework.stereotype.Controller;

@Controller
public class PatientController {

    private PatientService patientService;

    public PatientController(PatientService patientService) {
        super();
        this.patientService = patientService;
    }

    // handler method to handle list patients and return mode and view
    @GetMapping("/patients")
    public String listPatients(Model model) {
        model.addAttribute("patients", patientService.getAllPatients());
        return "patients";
    }

    @GetMapping("/patients/new")
    public String createPatientForm(Model model) {

        // create patient object to hold patient form data
        Patient patient = new Patient();
        model.addAttribute("patient", patient);
        return "create_patient";

    }

    @PostMapping("/patients")
    public String savePatient(@ModelAttribute("patient") Patient patient) {
        patientService.savePatient(patient);
        return "redirect:/patients";
    }

    @GetMapping("/patients/edit/{id}")
    public String editPatientForm(@PathVariable Long id, Model model) {
        model.addAttribute("patient", patientService.getPatientById(id));
        return "edit_patient";
    }

    @PostMapping("/patients/{id}")
    public String updatePatient(@PathVariable Long id,
            @ModelAttribute("patient") Patient patient,
            Model model) {

        // get patient from database by id
        Patient existingPatient = patientService.getPatientById(id);
        existingPatient.setId(id);
        existingPatient.setFirstName(patient.getFirstName());
        existingPatient.setLastName(patient.getLastName());
        existingPatient.setDepartment(patient.getDepartment());

        // save updated patient object
        patientService.updatePatient(existingPatient);
        return "redirect:/patients";
    }

    // handler method to handle delete patient request

    @GetMapping("/patients/{id}")
    public String deletePatient(@PathVariable Long id) {
        patientService.deletePatientById(id);
        return "redirect:/patients";
    }
}
```

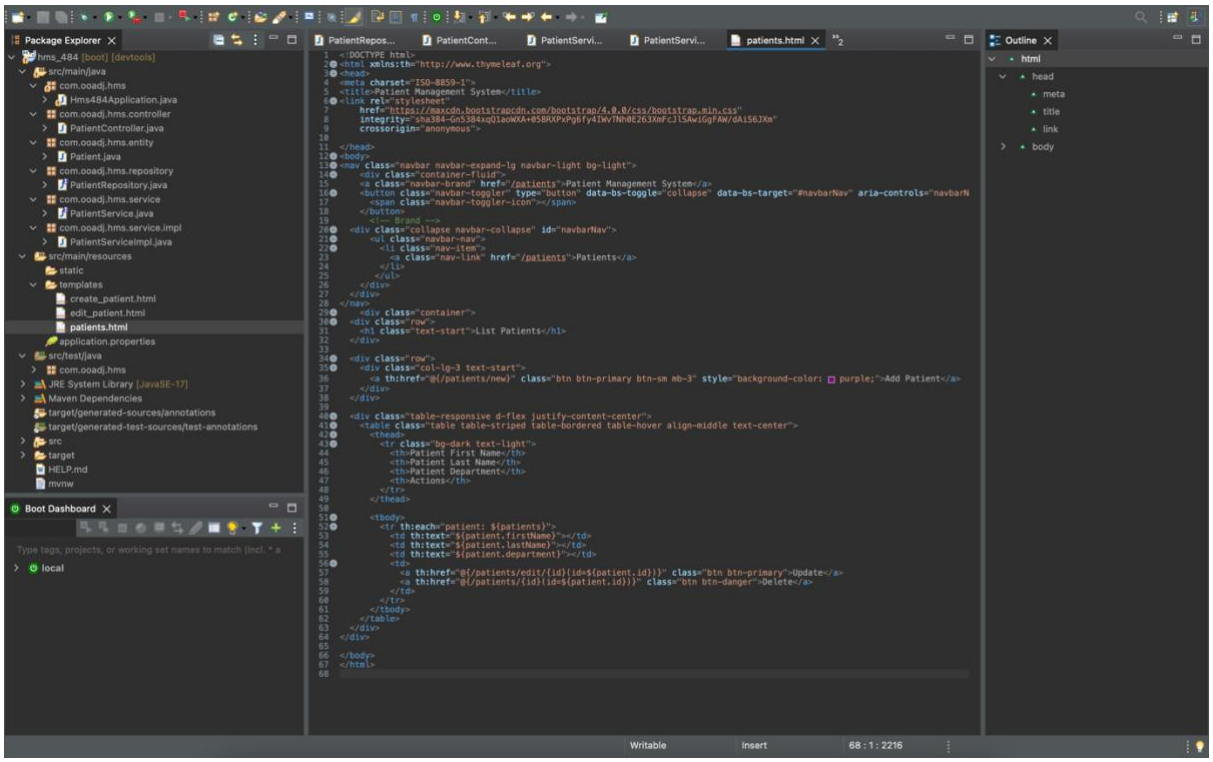
PatientService.java



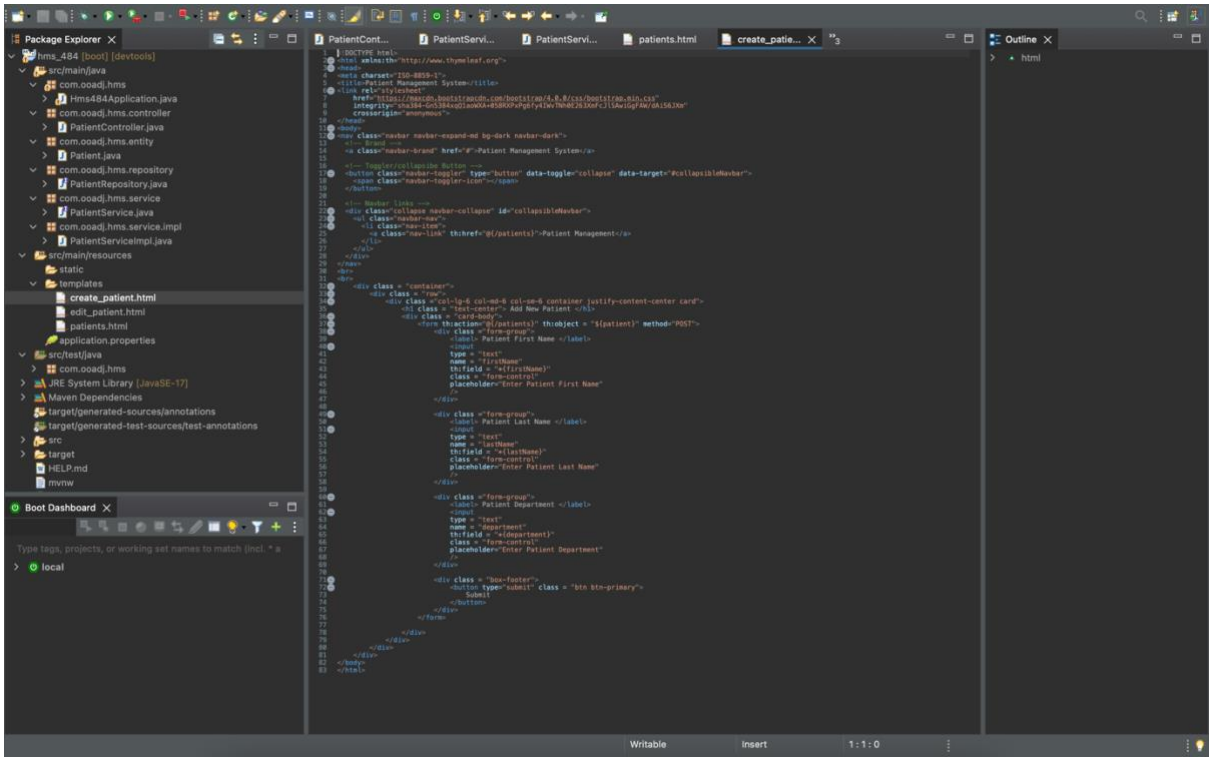
PatientServiceImpl.java



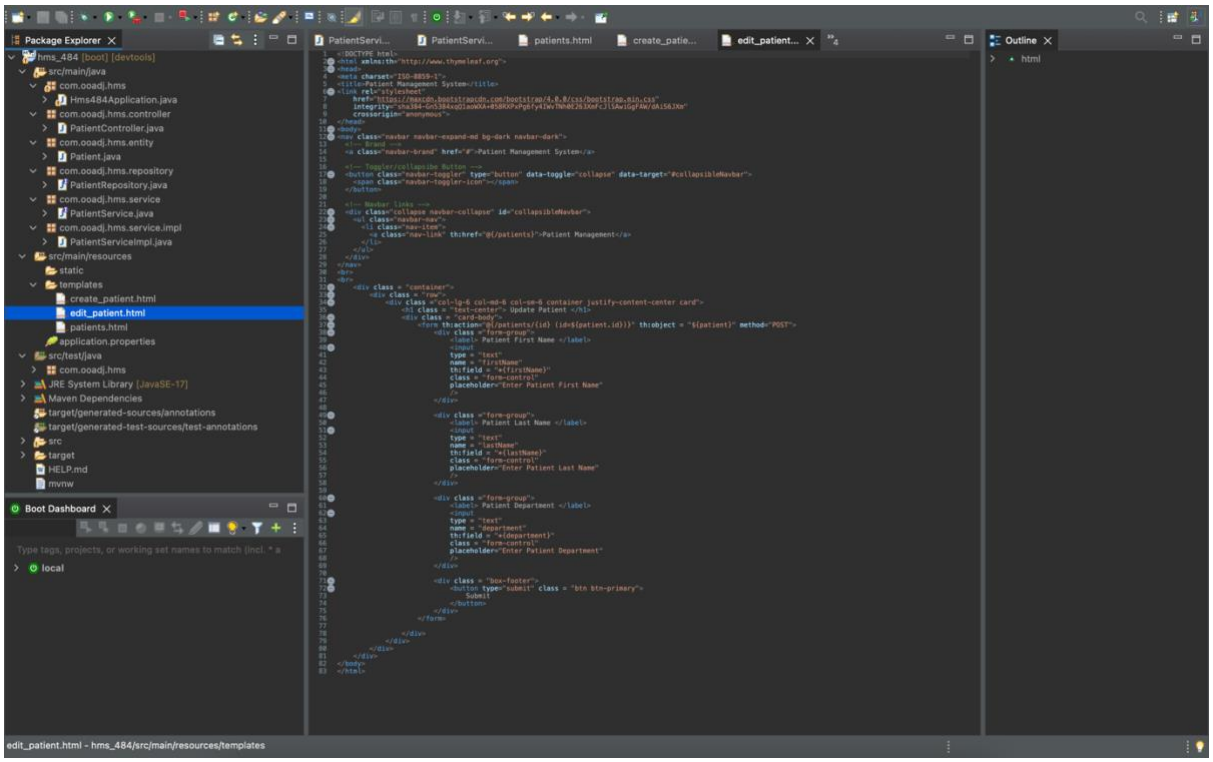
patients.html



create_patient.html

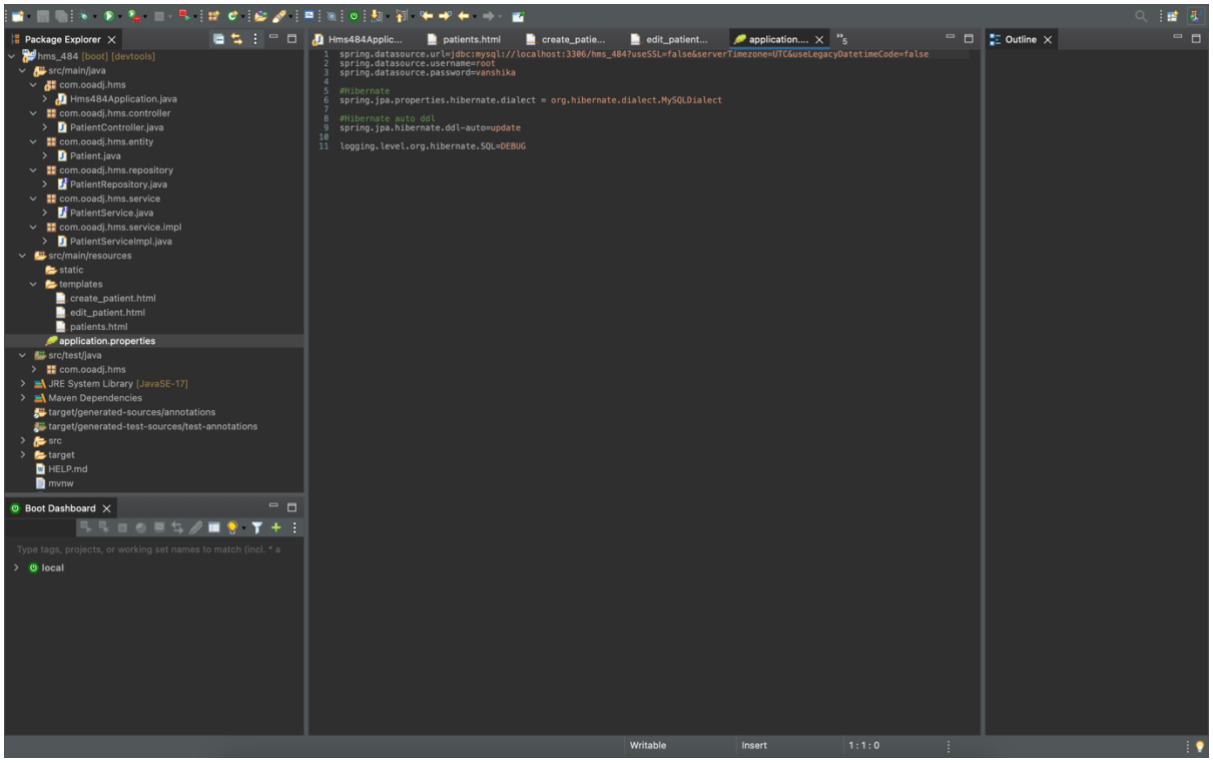


edit_patient.html

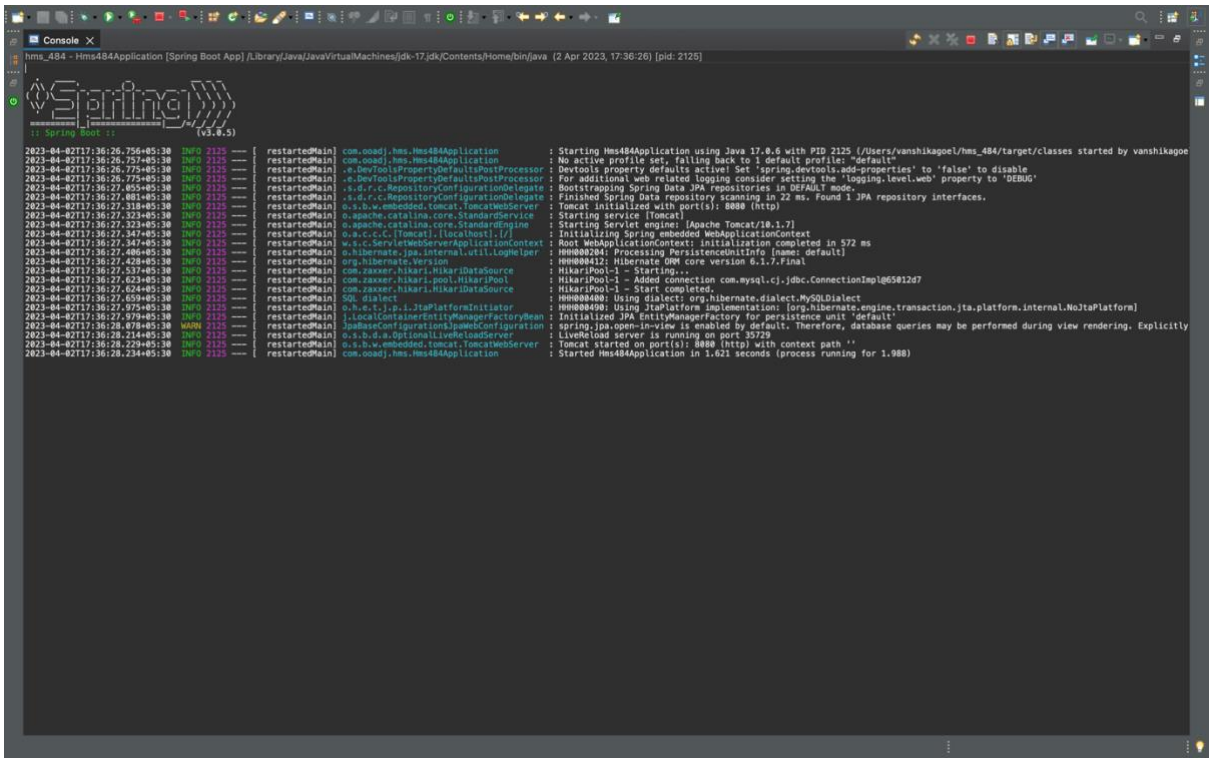


application.properties file

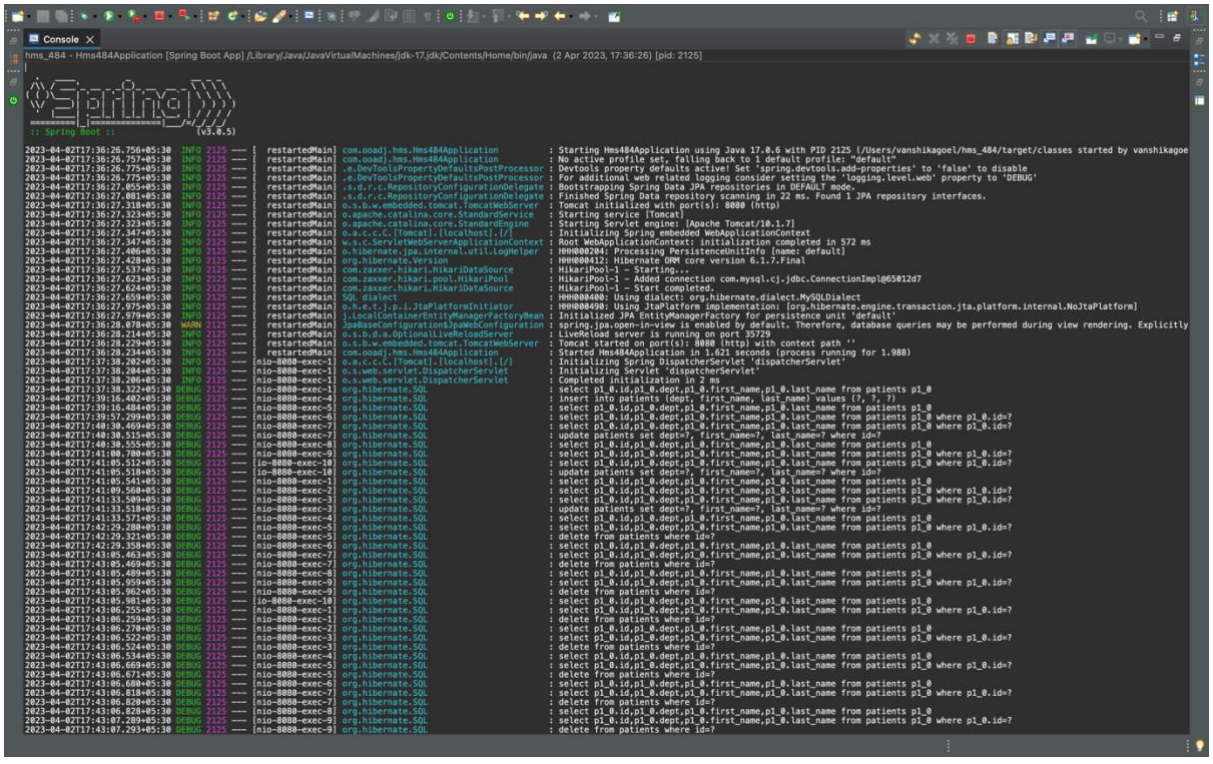
(It has the MySQL Database name mentioned as well as the username and password for MySQL server)



6. Screenshot of console with application running

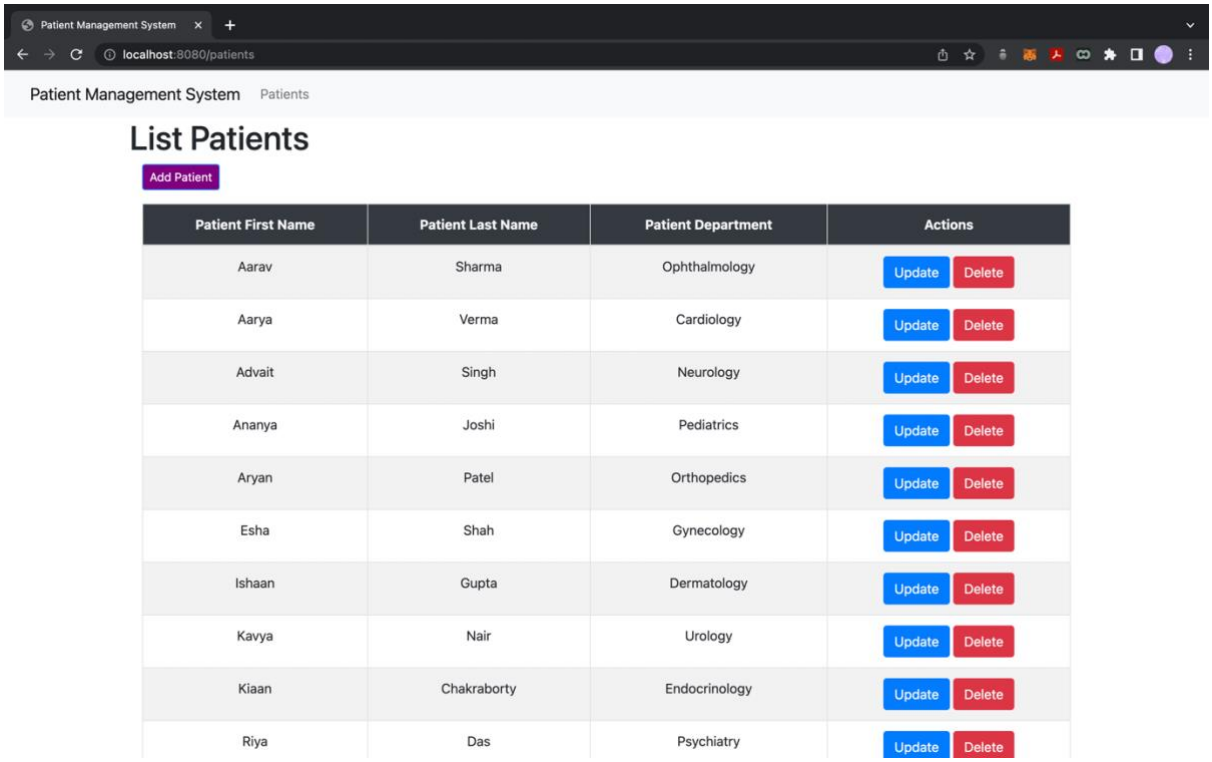


Console view of all CRUD operations being performed by the application on the MySQL server

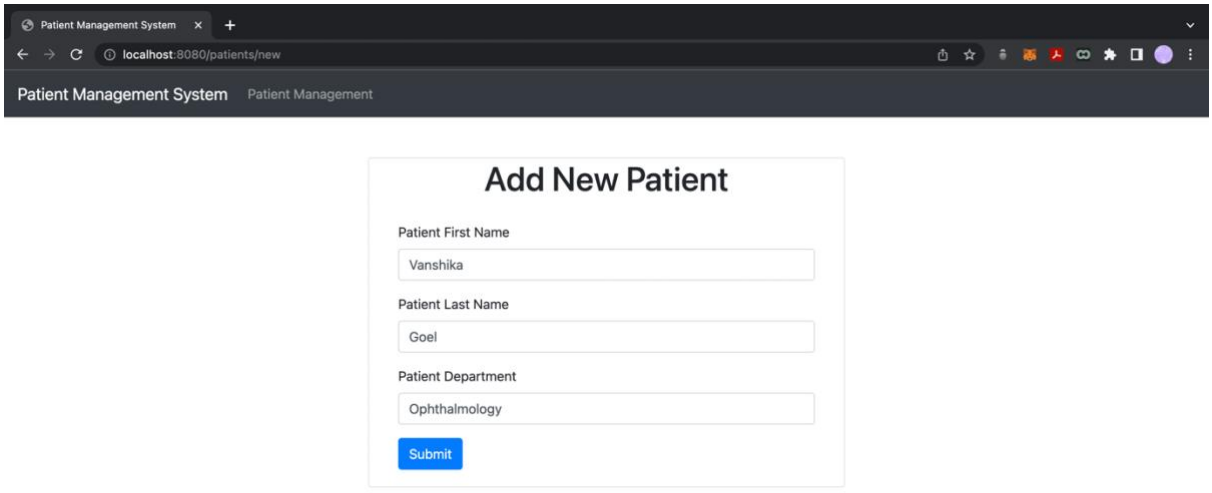


7. **Screenshot of UIs related to the two scenarios with values, outputs, errors (if any)**

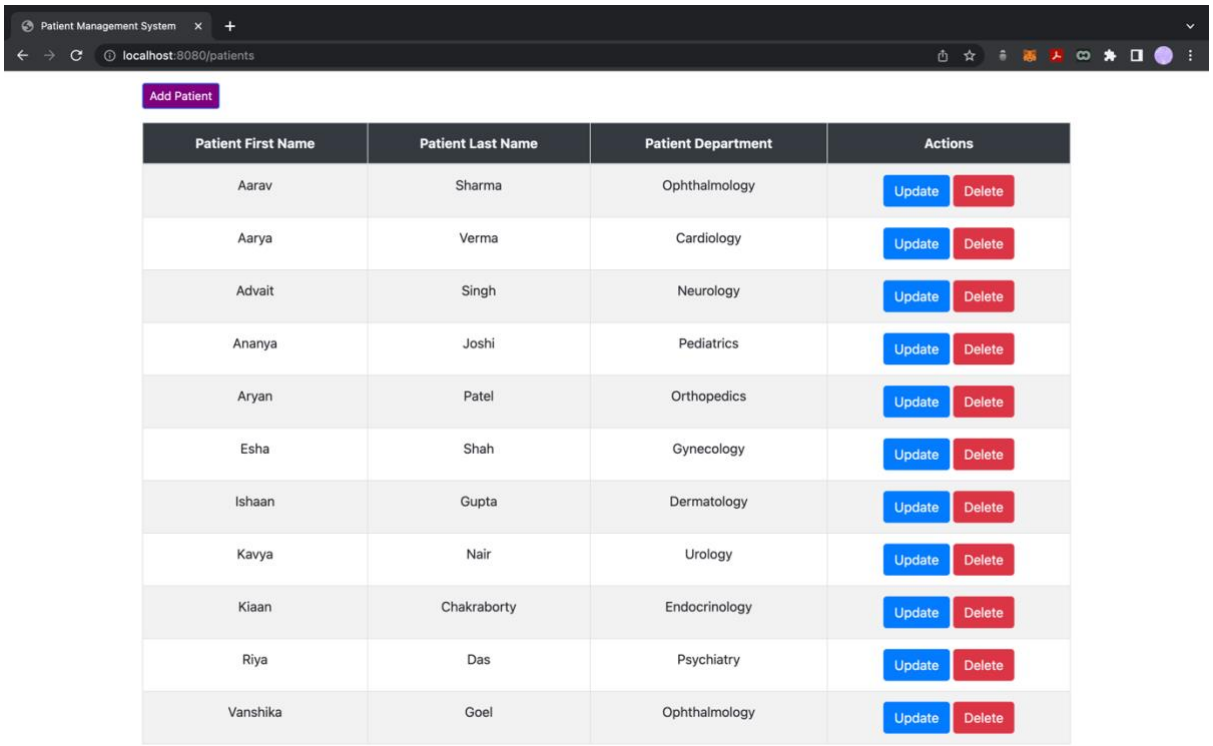
Patient Management System Home Page



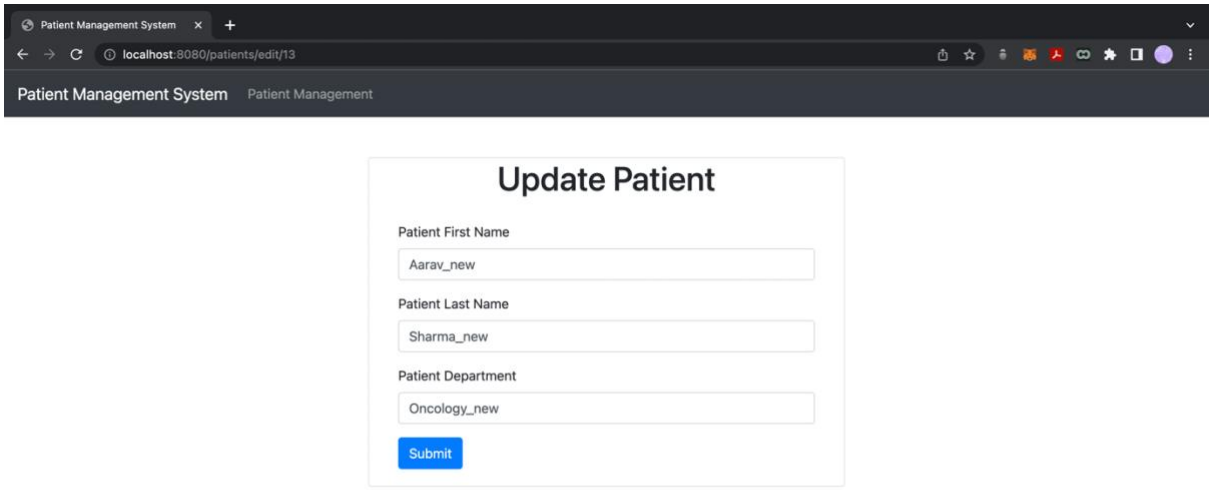
Add New Patient Page



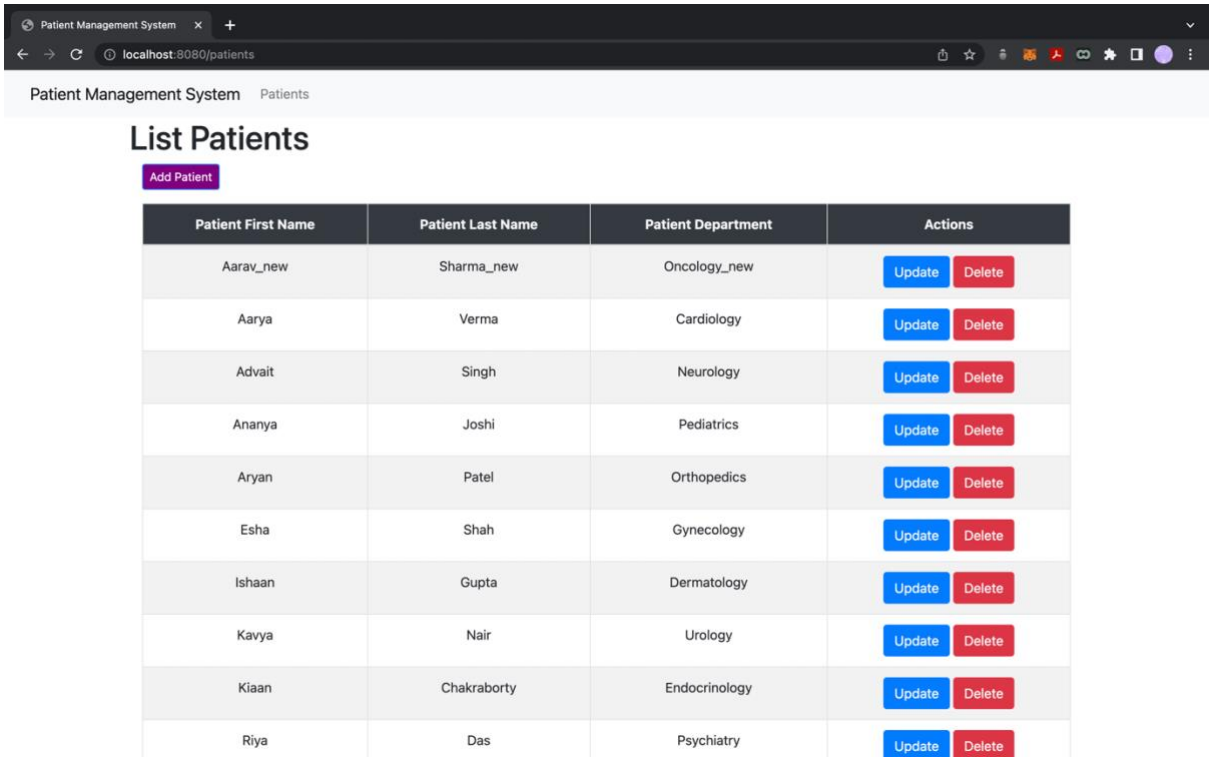
New Patient Details Added



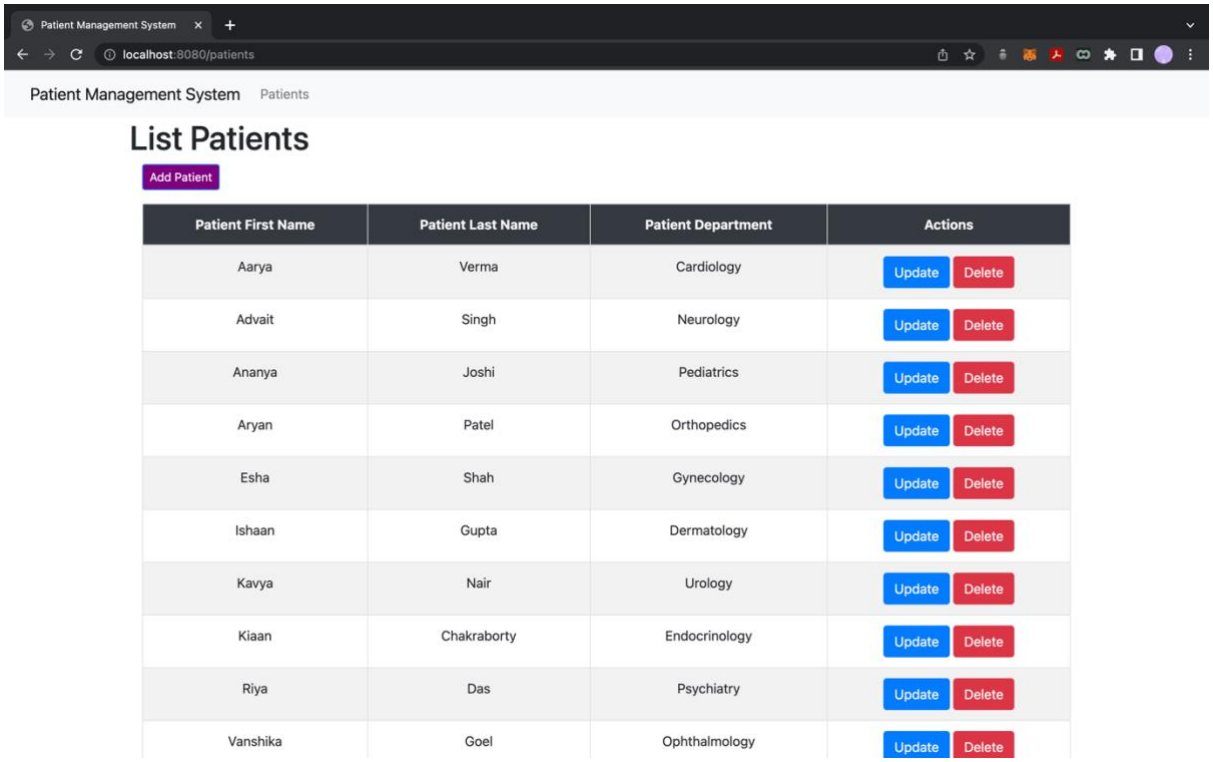
Update Patient Details



Patient Details Updated as shown above and below



Patient Details Deleted

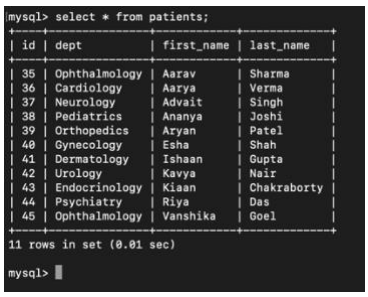


8. Screenshot of database with data items

MySQL Database Before Addition, Updating or Deletion of Data

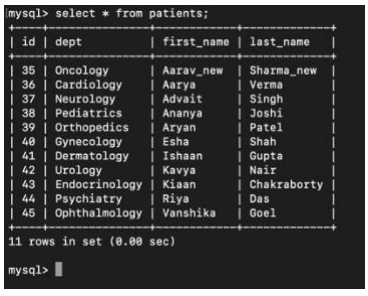
```
Vanshikas-MacBook-Air /Users/vanshikagoel$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 683
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use hms_484;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------+
| Tables_in_hms_484 |
+------------------+
| patients         |
+------------------+
1 row in set (0.01 sec)

mysql> select * from patients;
Empty set (0.00 sec)

mysql> INSERT INTO patients (dept, first_name, last_name) VALUES
    -> ('Ophthalmology', 'Aarav', 'Sharma'),
    -> ('Cardiology', 'Aarya', 'Verma'),
    -> ('Neurology', 'Advait', 'Singh'),
    -> ('Pediatrics', 'Ananya', 'Joshi'),
    -> ('Orthopedics', 'Aryan', 'Patel'),
    -> ('Gynecology', 'Esha', 'Shah'),
    -> ('Dermatology', 'Ishaan', 'Gupta'),
    -> ('Urology', 'Kavya', 'Nair'),
    -> ('Endocrinology', 'Kiaan', 'Chakraborty'),
    -> ('Psychiatry', 'Riya', 'Das');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
```

After Adding New Patient Details, as shown in the previous point.



```
mysql> select * from patients;
+----+---------------+------------+-------------+
| id | dept          | first_name | last_name   |
+----+---------------+------------+-------------+
| 35 | Ophthalmology | Aarav      | Sharma      |
| 36 | Cardiology    | Aarya      | Verma       |
| 37 | Neurology     | Advait     | Singh       |
| 38 | Pediatrics    | Ananya     | Joshi       |
| 39 | Orthopedics   | Aryan      | Patel       |
| 40 | Gynecology    | Esha       | Shah        |
| 41 | Dermatology   | Ishaan     | Gupta       |
| 42 | Urology       | Kavya      | Nair        |
| 43 | Endocrinology | Kiaan      | Chakraborty |
| 44 | Psychiatry    | Riya       | Das         |
| 45 | Ophthalmology | Vanshika   | Goel        |
+----+---------------+------------+-------------+
11 rows in set (0.01 sec)

mysql>
```

After Updating Patient Details, as shown in the previous point.



```
mysql> select * from patients;
+----+---------------+------------+-------------+
| id | dept          | first_name | last_name   |
+----+---------------+------------+-------------+
| 35 | Oncology      | Aarav_new  | Sharma_new  |
| 36 | Cardiology    | Aarya      | Verma       |
| 37 | Neurology     | Advait     | Singh       |
| 38 | Pediatrics    | Ananya     | Joshi       |
| 39 | Orthopedics   | Aryan      | Patel       |
| 40 | Gynecology    | Esha       | Shah        |
| 41 | Dermatology   | Ishaan     | Gupta       |
| 42 | Urology       | Kavya      | Nair        |
| 43 | Endocrinology | Kiaan      | Chakraborty |
| 44 | Psychiatry    | Riya       | Das         |
| 45 | Ophthalmology | Vanshika   | Goel        |
+----+---------------+------------+-------------+
11 rows in set (0.00 sec)

mysql>
```

After Deleting Patient Details, , as shown in the previous point.

```
mysql> select * from patients;
+----+---------------+------------+-------------+
| id | dept          | first_name | last_name   |
+----+---------------+------------+-------------+
| 36 | Cardiology    | Aarya      | Verma       |
| 37 | Neurology     | Advait     | Singh       |
| 38 | Pediatrics    | Ananya     | Joshi       |
| 39 | Orthopedics   | Aryan      | Patel       |
| 40 | Gynecology    | Esha       | Shah        |
| 41 | Dermatology   | Ishaan     | Gupta       |
| 42 | Urology       | Kavya      | Nair        |
| 43 | Endocrinology | Kiaan      | Chakraborty |
| 44 | Psychiatry    | Riya       | Das         |
| 45 | Ophthalmology | Vanshika   | Goel        |
+----+---------------+------------+-------------+
10 rows in set (0.01 sec)

mysql>
```

Complete view of all operations performed on MySQL Database

```
mysql> use hms_484;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------+
| Tables_in_hms_484 |
+------------------+
| patients          |
+------------------+
1 row in set (0.01 sec)

mysql> select * from patients;
Empty set (0.00 sec)

mysql> INSERT INTO patients (dept, first_name, last_name) VALUES
    -> ('Ophthalmology', 'Aarav', 'Sharma'),
    -> ('Cardiology', 'Aarya', 'Verma'),
    -> ('Neurology', 'Advait', 'Singh'),
    -> ('Pediatrics', 'Ananya', 'Joshi'),
    -> ('Orthopedics', 'Aryan', 'Patel'),
    -> ('Gynecology', 'Esha', 'Shah'),
    -> ('Dermatology', 'Ishaan', 'Gupta'),
    -> ('Urology', 'Kavya', 'Nair'),
    -> ('Endocrinology', 'Kiaan', 'Chakraborty'),
    -> ('Psychiatry', 'Riya', 'Das');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from patients;
+----+---------------+------------+-------------+
| id | dept          | first_name | last_name   |
+----+---------------+------------+-------------+
| 35 | Ophthalmology | Aarav      | Sharma      |
| 36 | Cardiology    | Aarya      | Verma       |
| 37 | Neurology     | Advait     | Singh       |
| 38 | Pediatrics    | Ananya     | Joshi       |
| 39 | Orthopedics   | Aryan      | Patel       |
| 40 | Gynecology    | Esha       | Shah        |
| 41 | Dermatology   | Ishaan     | Gupta       |
| 42 | Urology       | Kavya      | Nair        |
| 43 | Endocrinology | Kiaan      | Chakraborty |
| 44 | Psychiatry    | Riya       | Das         |
| 45 | Ophthalmology | Vanshika   | Goel        |
+----+---------------+------------+-------------+
11 rows in set (0.01 sec)

mysql> select * from patients;
+----+---------------+------------+-------------+
| id | dept          | first_name | last_name   |
+----+---------------+------------+-------------+
| 35 | Oncology      | Aarav_new  | Sharma_new  |
| 36 | Cardiology    | Aarya      | Verma       |
| 37 | Neurology     | Advait     | Singh       |
| 38 | Pediatrics    | Ananya     | Joshi       |
| 39 | Orthopedics   | Aryan      | Patel       |
| 40 | Gynecology    | Esha       | Shah        |
| 41 | Dermatology   | Ishaan     | Gupta       |
| 42 | Urology       | Kavya      | Nair        |
| 43 | Endocrinology | Kiaan      | Chakraborty |
| 44 | Psychiatry    | Riya       | Das         |
| 45 | Ophthalmology | Vanshika   | Goel        |
+----+---------------+------------+-------------+
11 rows in set (0.00 sec)

mysql> select * from patients;
+----+---------------+------------+-------------+
| id | dept          | first_name | last_name   |
+----+---------------+------------+-------------+
| 36 | Cardiology    | Aarya      | Verma       |
| 37 | Neurology     | Advait     | Singh       |
| 38 | Pediatrics    | Ananya     | Joshi       |
| 39 | Orthopedics   | Aryan      | Patel       |
| 40 | Gynecology    | Esha       | Shah        |
| 41 | Dermatology   | Ishaan     | Gupta       |
| 42 | Urology       | Kavya      | Nair        |
| 43 | Endocrinology | Kiaan      | Chakraborty |
| 44 | Psychiatry    | Riya       | Das         |
| 45 | Ophthalmology | Vanshika   | Goel        |
+----+---------------+------------+-------------+
10 rows in set (0.01 sec)

mysql>
```