

JAVA PROGRAMMING
Course Code – R1UC304C
SEM–III (Session 2025-26)



HEALTHCARE MANAGEMENT SYSTEM

TEAM MEMBERS:

PIYUSH KUMAR (24SCSE1010146)

YOGANSHU (24SCSE1010335)

RANA PRATAP SINGH (24SCSE1010076)

Project Introduction

- ◆ • Web-based Healthcare Management System
- ◆ • Manages doctors, patients, appointments
- ◆ • Uses Java Servlet technology

Objectives of the Project

- ◆ • Implement Servlet-based web application
- ◆ • Perform CRUD operations using JDBC
- ◆ • Secure healthcare data
- ◆ • Session-based authentication
- ◆ • User-friendly interface

Technologies Used

- ◆ • Frontend: JSP, HTML, CSS
- ◆ • Backend: Java Servlets
- ◆ • Database: MySQL
- ◆ • Server: Apache Tomcat 9
- ◆ • IDE: NetBeans
- ◆ • JDK: 17+

System Architecture

- ◆ • MVC based architecture
 - ◆ • View: JSP Pages
 - ◆ • Controller: Servlets
 - ◆ • Model: MySQL Database
-
- ◆ Flow: JSP → Servlet → DB → Servlet → JSP

Servlet Implementation

- ◆ • LoginServlet
 - ◆ • LogoutServlet
 - ◆ • DoctorServlet
 - ◆ • PatientServlet
 - ◆ • AppointmentServlet
-
- ◆ Each servlet extends HttpServlet

LoginServlet Working

- ◆ • Accepts username & password
- ◆ • Validates user using JDBC
- ◆ • Creates HttpSession
- ◆ • Redirects to dashboard
- ◆ • Handles invalid login

Session Management

- ◆ • Session created after login
- ◆ • Session attribute: username
- ◆ • Protected pages using session check
- ◆ • Logout destroys session

Doctor Management Module

- ◆ • Add doctor details
- ◆ • View doctor list
- ◆ • Data stored in MySQL
- ◆ • Handled using DoctorServlet

Patient Management Module

- ◆ • Register patient
- ◆ • View patient details
- ◆ • Data handled by PatientServlet

Database Design

- ◆ • user table
- ◆ • doctor table
- ◆ • patient table
- ◆ • appointments table

- ◆ Each table has primary key

Code Quality & Execution

- ◆ • Clean package structure
- ◆ • No SQL in JSP
- ◆ • Reusable DBConnection class
- ◆ • Proper exception handling
- ◆ • Efficient execution

Error Handling

- ◆ • Invalid login handling
- ◆ • Session validation
- ◆ • Try-catch blocks
- ◆ • User-friendly messages

Innovation / Extra Effort

- ◆ • Session-based security
- ◆ • Professional UI using CSS
- ◆ • Gradient buttons
- ◆ • Card-based layout
- ◆ • Scalable design

Project Flow

- ◇ Login → Dashboard
- ◇ → Add / View Doctor
- ◇ → Add / View Patient
- ◇ → Logout

Advantages

- ◆ • Secure
- ◆ • Easy data management
- ◆ • User-friendly
- ◆ • Scalable
- ◆ • Time efficient

Future Enhancements

- ◆ • Online appointment booking
- ◆ • Admin roles
- ◆ • Email notifications
- ◆ • Report generation

Conclusion

- ◆ • Successfully implemented Java Servlets
- ◆ • Secure session handling
- ◆ • Clean & modular code
- ◆ • Professional healthcare system

THANKS