Code-a-thon

on

"Online Application for Rural Healthcare Awareness using Java"

SWASTH SAARTHI

Submitted to

School of Computer Science Engineering and Information Systems

Department of Computer Applications

Submitted by:

(Team name: Software Srijan)

Stuti Rajeev(24MCA0173)

Barsha Routh(24MCA0164)

Ish Jaiswal(24MCA0172)

Srishti Jalan(23MCA0170)

Nidhi Tulsyan(24MCA0187)

Acknowledgement

We would like to express my sincere gratitude to **Dr. Senthil Murugan** for their invaluable guidance and continuous support throughout this project. Their expertise and feedback have helped me to refine and improve our work significantly.

A special thanks to our **Head of Department (HOD), Dr. Vijayan E**, for fostering an environment of learning and innovation, and for providing the resources needed for the successful completion of this hackathon project.

We would also like to thank the organizers of the **Coda-a-thon** for creating this platform, which has allowed us to apply my skills in a real-world scenario. The project was developed using **Eclipse IDE**, which greatly enhanced the efficiency of coding and debugging.

Lastly, I would like to thank our friends, and fellow participants for their constant encouragement and collaboration throughout this journey.

List of content

1.Project Abstract

The Healthcare Awareness Web Application is designed to enhance public health literacy by offering an integrated platform for healthcare information, event updates, and easy access to medical consultations. This Java-based web application enables users to stay informed about healthcare through educational blogs, book doctor appointments, and receive updates on health-related events such as vaccination drives, health camps, and wellness workshops.

The system is built with a user-friendly interface that allows users to register, log in, and access a wide range of healthcare content, including articles on disease prevention and healthy living, contributed by professionals and verified sources. Users can also search for upcoming health events in their locality and book appointments with doctors.

Key features include:

- **Blog Module**: Provides users with access to educational healthcare articles and tips, categorized by various medical conditions.
- **Event Module**: Alerts users about upcoming healthcare-related events in their area, such as vaccination camps, health camps, and wellness seminars.
- **Appointment Module:** Allows users to browse and book appointments with healthcare professionals based on availability and specialization.

The system includes a robust database for managing user profiles, doctor information, event schedules, and appointment bookings, ensuring seamless access to essential healthcare services. Developed using Java with JSP and Java Servlets, and managed through Eclipse IDE and Oracle database, the project prioritizes usability, scalability, and data security. The application is designed to be a one-stop solution for users seeking health awareness and easy access to medical services, thus fostering a healthier and more informed community.

2.Page Descriptions

Landing Page (landing.jsp):
The first page users see upon accessing the application. It provides an overview of the application, highlighting key features and services.

User Details Page (userDetails.jsp):
Displays the personal information of the logged-in user, including their health records and appointment history.

Register Page (register.jsp):
Allows new users to create an account by entering their personal details. This information is stored in the database for future access.

Footer Page (footer.jsp):
Contains copyright information and additional links, providing a consistent footer across all pages.

Blog Card Page (blogCard.jsp):
Displays a collection of health-related blog posts. Each post includes a title, a brief description, and a link to read more.

Health Login Page (healthLogin.jsp):
Provides a login form for users to access their accounts. Users can enter their credentials to log in.

Add Blog Page (addBlog.jsp):
Enables authorized users to create new blog posts by entering the title and content. This page is restricted to administrative users.

Appointment Page (appointment.jsp):
Allows users to schedule appointments with healthcare providers. Users can select a date, time, and healthcare professional.

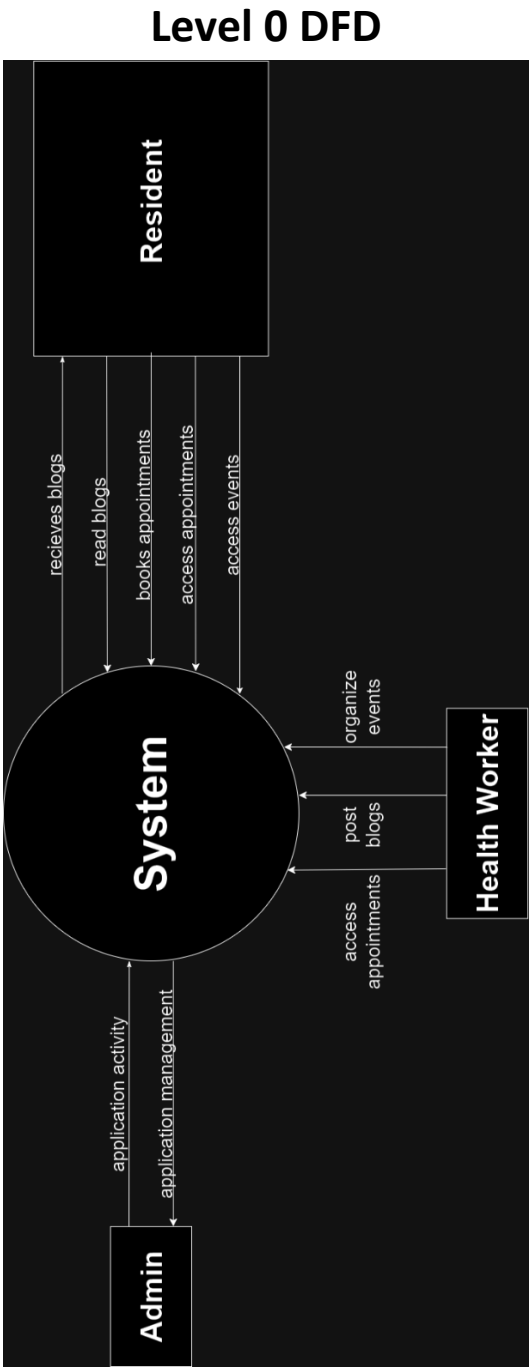
View Appointment Page (viewAppointment.jsp):
Displays a list of appointments scheduled by the user, along with details such as date, time, and provider information.

Events Page (events.jsp):
Lists upcoming health awareness events in the community. Users can view details and register for these events.

Add Events Page (addEvents.jsp):
Allows authorized users to add new events to the application, including details such as date, time, location, and description.

View Resident Details Page (viewResidentDetails.jsp):
Allows healtworkers to view Residents details already registered in a application.

3.Data Flow Diagram –



4.Database Design –

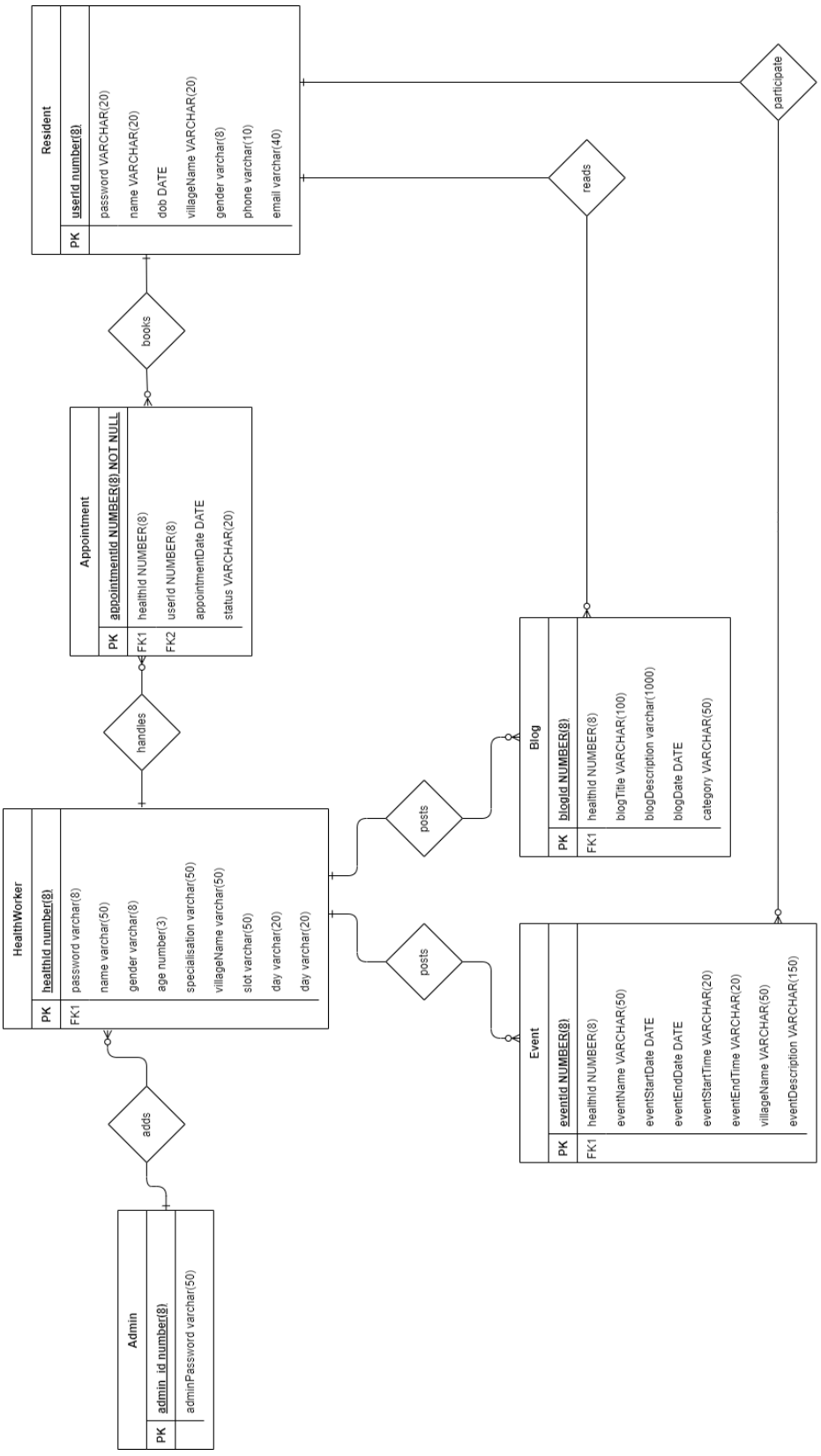
Entities:

- **Resident**: Represents users who access the platform.
- **HealthWorker:** Represents healthcare providers.
- **Event:** Posted by HealthWorker.
- **Appointment:** To book appointment for a user.
- **Blog-**To Display the articles about health care Workers.

Relationships:

- A **User** can book multiple**Appointments with Doctors.**
- A **Doctor** provides multiple **add blog, events.**

ER Diagram –



Tables Overview:

1. **Resident Table**
    - **Columns:** UserID, Name, Password, Email, DOB(Date of Birth), Location, Gender
2. **HealthWorker(Doctor) Table**
    - **Columns:** Hid, Hname, Password, Gender, Name,age,Spec_id, village, Slot, day
3. **Event Table**
    - **Columns:** eventid, Name, eventStartDate, eventEndDate, eventStartTime, eventEndTime, healthId, eventDescription.
4. **Appoinment Table**
    - **Columns:** AppointmentId, healthId, userId, appointmentDate, Status.
5. **District Table**
    - **Columns:** districtId, districtName.
6. **Village Table**
    - **Columns:** districtId, villageName
7. **Blog Table**
    - **Columns:** blogId, Blog_title, blogDescribtion, blogDate, healthId, category.

5. Module Description

4.1. User Module (Healthcare Worker or Residents)

- Functionality: Allows users to register, login, and access healthcare information. Users can also book consultations and view alerts.
- Input: Username, Password, Appointment Booking Details.
- Output: Access to articles, booking confirmation, alerts.

4.2. Appointment Booking Module

- Functionality: Manages the system's articles, alerts, and doctor information.
- Input: Admin login, article and alert details.
- Output: Updated content and healthcare resources.

4.3. Event Module

- Functionality: Event added by Health care workers(Doctors) for there residents(Users). Users can view and participate in various events(Special Vaccination Capms ,Marathons etc)
- Input: Event details, date and time schedule.
- Output: Users registration for Event.

4.5. Blog Module

- Functionality: Provides users with access to educational healthcare articles and tips, categorized by various medical conditions.
- Input: Search keywords.
- Output: List of relevant articles.

4.4. Admin Module

- Functionality: Manages Healthcare workers details and keeps tract of the information circulating in the platform. Can add Healthcare workers and remove.
- Input:User_id of healthcare workers.
- Output: Message relevant to the Operation.

6. Technologies and Tools Used

1. Programming Language: Java

- **Version:** Java SE 22
- **Description:** Java is a high-level, object-oriented programming language used for developing platform-independent applications. In this project, Java is used to implement the core business logic, including the interaction between the frontend and backend, data processing, and handling requests/responses in a web-based environment.
- **Key Features in this Project:**
  - Exception handling to ensure smooth execution of the application.
  - Multithreading for handling multiple requests simultaneously in a web environment.
  - JDBC (Java Database Connectivity) to interact with the Oracle database.
  - JSP
  - JAVA Servlet

2. Integrated Development Environment (IDE): Eclipse

- **Version:**Eclipse 11g
- **Description:** NetBeans is a popular open-source integrated development environment (IDE) used for Java development. It provides tools and frameworks to facilitate easy and efficient development, debugging, and deployment of Java applications.
- **Role in the Project:**
  - Development and testing of Java code.
  - Project management and version control.
  - Integrated support for building and deploying the web application using Apache Tomcat.

3. Database: Oracle Database

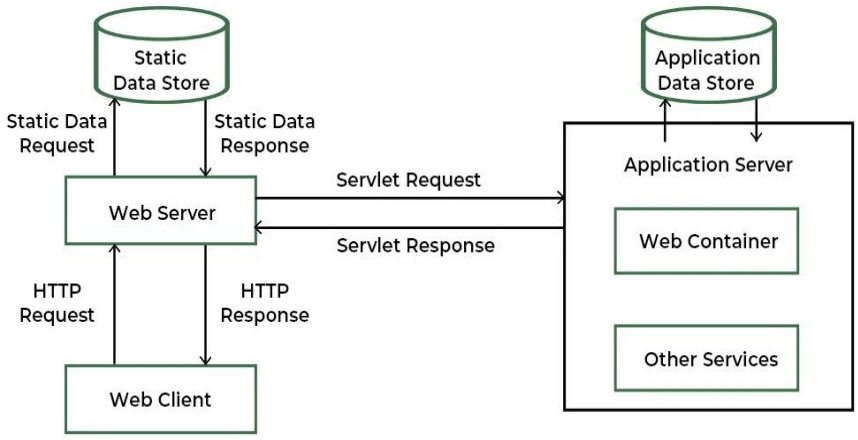
- **Version:** Oracle 11g Express Adition.

- **Description:** Oracle Database is a relational database management system (RDBMS) that provides high-performance data storage, retrieval, and manipulation. Oracle is known for its scalability, security, and robustness in handling large amounts of data.
- **Role in the Project:**
    - Store and manage data related to the project, such as user records, Health Worker information.
    - The Java application interacts with the Oracle database using JDBC for CRUD operations (Add, Read, Update, Delete).
- **Key Features Used:**
    - SQL for add User information and to store Healthworker data .

4. Web Server: Apache Tomcat

- **Version:** Tomcat 10.1.0
- **Description:** Apache Tomcat is an open-source web server and servlet container designed to deploy Java Servlets and JSPs (JavaServer Pages). It serves as the middleware between Java applications and HTTP clients, handling requests and responses in a web-based environment.
- **Role in the Project:**
    - Deploy and manage the web application created using Java and JSP.
    - Handle incoming HTTP requests, process the Java Servlet or JSP, and generate dynamic content to be sent to the client.
- **Integration:**
    - Integrated with eclipse for seamless development and deployment.
    - Serves the application's frontend while interacting with the backend (Oracle Database) through the Java business logic.

- **Working of webserver:**



Hardware Requirements

Minimum Hardware Requirements

- **Processor:** Intel i5 (or equivalent) and above.
- **RAM:** 8 GB or higher.
- **Hard Disk Space:** 10 GB free space for project development and database management.
- **Operating System:** Windows 10 or Linux-based OS (Ubuntu 20.04 recommended).
- **Other:** Stable internet connection for accessing and managing cloud-based tools (if applicable).

Software Requirements

Software Stack Used

- **Java SE Development Kit (JDK):** Java 22.
- **NetBeans IDE:** Eclipse11g.
- **Oracle Database:** Oracle 11g.
- **Apache Tomcat:** Tomcat 10.1.0.
- **JDBC Driver:** Oracle JDBC Driver (ojdbc11.jar for Java 8+).

Architecture of Project:

Frontend: JSP (JavaServer Pages) for creating dynamic web content and Bootstrap for styling.