A Project Report on

POSHAN ABHIYAAN

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

In partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

G.RISHIKA (16BD1A054D)

G.MANASA (16BD1A054N)

K.SWETHA (16BD1A055K)

Under the esteemed guidance of

Mrs. JYOSHNA BEJJAM

Assistant Professor

Department of CSE

Department of Computer Science and Engineering

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTUH

3-5-1206, Narayanaguda, Hyderabad - 500029

2019 – 2020

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTU, Hyderabad

3-5-1206, Narayanaguda, Hyderabad - 500029.

CERTIFICATE

This is to certify that the project entitled **"POSHAN ABHIYAAN"** being submitted by **G.RISHIKA (16BD1A054D), G.MANASA (16BD1A054N), K.SWETHA (16BD1A055K)** students of **Keshav Memorial Institute of Technology, JNTUH** in partial fulfillment of requirements of the award of the Degree of **Bachelor of Technology in Computer Science and Engineering** as a specialization is a record of bonafide work carried out by them under my guidance and supervision in the academic year 2019 – 2020.

GUIDE
Mrs. Jyoshna Bejjam
Assistant Professor

HOD
Dr. S. Padmaja
CSE DEPARTMENT

Submitted for the Project Viva Voice examination held on …….

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project report entitled **"POSHAN ABHIYAAN"** is done in the partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhere else.

G.RISHIKA (16BD1A054D)
G.MANASA (16BD1A054N)
K.SWETHA (16BD1A055K)

ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work.

We render our thanks to **Dr. Maheshwar Dutta**, M.Tech., Ph.D.,Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. S. Nitin**, Director and **Mrs. Deepa Ganu**, Dean Academic for providing an excellent environment in the college.

We are also thankful to **Dr. S. Padmaja**, Head, Department of CSE for providing us with both time and amenities to make this project a success within the given schedule.

We are also thankful to our guide **Mrs. Jyoshna Bejjam**, for his/her valuable guidance and encouragement given to us throughout the project work.

We would further like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project.

We sincerely thank our friends and family for their constant motivation during the project work.

G.RISHIKA (16BD1A054D)
G.MANASA (16BD1A054N)
K.SWETHA (16BD1A055K)

CONTENTS

CHAPTER-4

CHAPTER-5

ABSTRACT

The aim of Poshan Abhiyaan is to promote holistic nutrition among women and children through/with the help of IT. So, a system is designed to monitor and trigger alarms if some women/child have not come for upcoming dose.

As there are multiple vaccines and tests to be taken, there are chances that a person may forget to take their vaccine/ test. This may leave the person impaired for life depending on the kind of vaccine he forgets to take. Thus, we provide a system where the user only has to enter the child's date of birth and the start date of pregnancy in order to receive reminders to take the child's and pregnant's vaccine respectively.

Additionally, information about diet, yoga and nutrition are provided for both children and women. Also, a daily meal plan suggestion will be given to the pregnant lady based on her trimester. A chatbot is also available for anyone who would like to get a little insight about the diet, nutrition and vaccination.

LIST OF FIGURES

1.INTRODUCTION

1.1 PURPOSE OF THE PROJECT

Our Web Application facilitates young parents in tracking the immunization of their children. The application takes away pressure of having to remember life critical information amidst juggling work and home for the parents. The objective of this Portal, is to provide healthcare related information to the citizens of India and to serve as a single point of access for consolidated health information. Women and children have always been a focal audience in its vision.

Our Application aims to provide Immunization to both mother and child. Immunization is the process whereby a person is made immune or resistant to an infectious disease, typically by the administration of a vaccine. Vaccines are substances that stimulate the body's own immune system to protect the person against subsequent infection or disease.

Pregnancy is not an absolute contraindication to any vaccination. On the contrary, some vaccines are strongly recommended for pregnant women during the prenatal period. Therefore, the prenatal visit is an ideal time to assess a woman's need for vaccines.

The period after delivery and before discharge from the hospital is an ideal time to administer both live and inactivated vaccines. It ensures that both the woman and her child will be protected from preventable diseases after leaving the birthing facility, when they are especially vulnerable.

1.2 PROBLEMS WITH EXISTING SYSTEM

A person needs to enquire about the vaccinations either with the doctor or look it up online, make a note of it and should remember to go to the doctor on that day. There are web sites which

provide information regarding the vaccinations, but, they don't provide any kind of alerts and notifications for the users. A person might forget to take his vaccination or may end up taking it too late.

Sometimes, a person may not receive complete information as to what vaccine should be taken at what time. Or they might receive wrong information. Either way, a person might end up not taking his vaccination. This may leave the person impaired for life.

1.3 PROPOSED SYSTEM

The proposed system is of a web application that is helpful to keep track of the vaccinations that needs to be taken during pregnancy and after delivery for the baby as well as mother. The user will have to mention his/her details. Then the user will have to choose from the options whether pregnant or delivered and then the user will have to mention the month of pregnancy if she is pregnant or the age of the baby if she delivered. Depending on the details provided by the user the web application will display the list of vaccinations and nutrition that needs to be taken at that particular time from the database.

An Email Alert will be sent to the user one week prior to the vaccination to remind the user of vaccination. Another Email will be sent 3 days before the vaccination and one more on the day of the vaccination. After the vaccination is done the user has to update the status of the vaccination as done by clicking on the checkbox otherwise an alert for the vaccination which is not yet done is sent for a week to remind the user of vaccination.

A list of vaccination centers will be displayed from which the user can visit any nearest center of his or her choice to get the vaccinations done. The Nutrition that needs to be taken during this time is also mentioned. For better communication the website has provided a chatbot for the users, they can use the chatbot for any kind of queries.

The web application is user friendly and has easily understandable user interface which in turn enables easy operation in the case of an emergency. Once the details have been submitted the

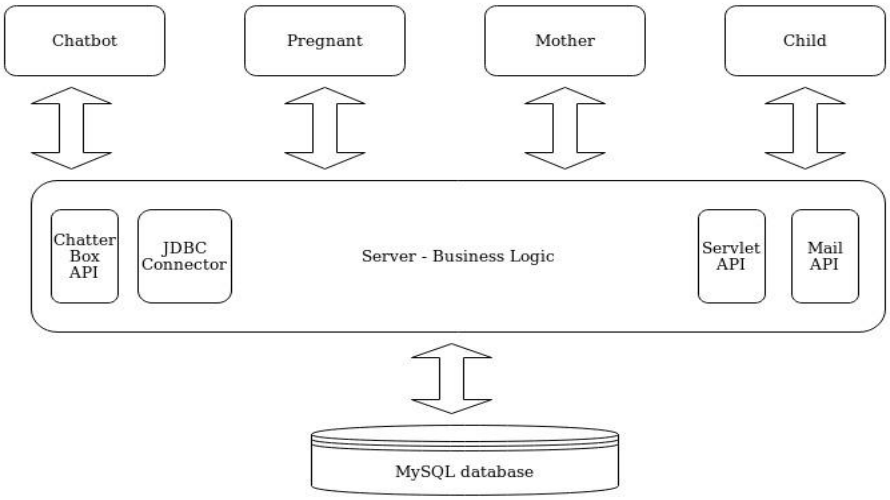
web application connects to the database and retrieves information in the form of a query from the database.

1.4 SCOPE OF THE PROJECT

Our application allows users to register providing the information. User can start their journey with our application from their pregnancy period. They can even join at any month of their pregnancy and after delivery for their child welfare.

Our Application provides the users all the information needed, regarding all the vaccinations and the nutrition required during the particular period in their journey from pregnancy to their child's 18 years of age. We send Alerts for the users regarding vaccination dates. The alerts are given throughout the week of dosage. We also provide the vaccination centers in their city.

1.5 ARCHITECTURE DIAGRAM



2. SYSTEM REQUIREMENT SPECIFICATIONS

What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirementphase.)
The SRS phase consists of two basic activities:

Problem/Requirement Analysis

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of thisphase.

Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software

development. A good SRS should satisfy all the parties involved in the system.

2.1 Requirements Specification Document

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application.

There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and nonfunctional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

The purpose of SRS (Software Requirement Specification) document is to describe the external behaviour of the application developed or software. It defines the operations, performance and interfaces and quality assurance requirement of the application or software. The complete software requirements for the system are captured by the SRS.

This section introduces the requirement specification document for Cancer Detection using Machine Learning which enlists functional as well as non-functional requirements.

2.1.1 Functional Requirements

For documenting the functional requirements, the set of functionalities supported by the system are to be specified. A function can be specified by identifying the state at which data is to be input to the system, its input data domain, the output domain, and the type of processing to be carried on the input data to obtain the output data.
Functional requirements define specific behaviour or function of the application. Following are the functional requirements:

The input design is the link between the information system and user. It compromises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a useable form for processing can be achieved by inspecting the computer to read data from a written or printed documented or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input design considered for the following things:

- What data should be given asinput?
- How the data should be arranged orcoded?
- The dialog to guide the operating personnel in providinginput
- Methods for preparing input validations and steps to follow when erroroccur.

2.1.1.1. Input design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management correct information from the computerizedsystem.

2.1.1.2. It is achieved by creating user-friendly screens for data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulations can be performed. It also provides viewingfacilities.

2.1.1.3. When the data is entered, it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

1. Enter name and details
2. Tracking the records of users
3. Information about Vaccinations
4. Nutrition Information
5. Sending Alerts the users
6. Storing feedback given by customers

2.1.2 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Especially these are the constraints the system must work within. Following are the non-functional requirements:

Performance:

The performance of the developed applications can be calculated by using following methods:
Measuring enables you to identify how the performance of your application stands in relation to your defined performance goals and helps you to identify the bottlenecks that affect your application performance. It helps you identify whether your application is moving toward or away from your performance goals. Defining what you will measure, that is, your metrics, and defining the objectives for each metric is a critical part of your testing plan.
Performance objectives include the following

- Response time orlatency
- Throughput
- Resourceutilization

1. Security for customer details
2. 24*7 availability
3. Giving locations of nearest vaccination centers within a range of 1-2 km
4. Re-usability
5. Reliability
6. The system will allow notifications to be sent to users
7. The platform will protect user data and encrypt passwords
8. The website will be responsive

2.2 SoftwareRequirements

- Eclipse
- MySQL Workbench
- JDBC Connector
- Apache Tomcat Server
- Internet Browser

2.3 HardwareRequirements

- Processor - Intel i5 (Min 2.4 GHz)
- RAM - 8GB
- Disk Space – 1 G

3. LITERATURE SURVEY

SOFTWARE DEVELOPMENT LIFECYCLE (SDLC) is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software that meets customer expectations. The system development should be complete in the pre-defined time frame and cost. SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. Every phase of the SDLC life cycle has its own process and deliverables that feed into the next phase.

The prime reasons why SDLC is important for developing a software system.

- It offers a basis for project planning, scheduling, and estimating
- Provides a framework for a standard set of activities and deliverables
- It is a mechanism for project tracking and control
- Increases visibility of project planning to all involved stakeholders of the development process
- Increased and enhance development speed
- Improved client relations
- Helps you to decrease project risk and project management plan overhead

The entire SDLC process divided into the following stages:

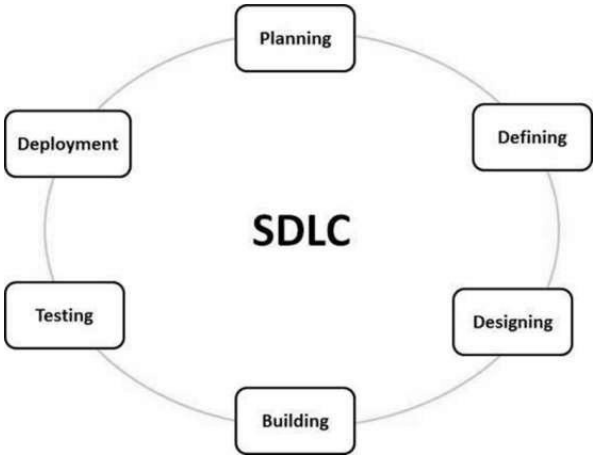


Figure 1. SDLC Phases

Phase 1: Requirement collection and analysis:

The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognization of the risks involved is also done at this stage.

This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project.

Requirements Gathering stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.

Phase 2: Feasibility study:

Once the requirement analysis phase is completed the next step is to define and document software needs. This process conducted with the help of 'Software Requirement Specification' document also known as 'SRS' document. It includes everything which should be designed and developed during the project life cycle.

There are mainly five types of feasibilities checks:

- **Economic:** Can we complete the project within the budget or not?
- **Legal:** Can we handle this project as cyber law and other regulatory framework/compliances.
- **Operation feasibility:** Can we create operations which is expected by the client?
- **Technical:** Need to check whether the current computer system can support the software
- **Schedule:** Decide that the project can be completed within the given schedule or not.

Phase 3: Design:

In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.

This design phase serves as input for the next phase of the model.

There are two kinds of design documents developed in this phase:

High-Level Design (HLD)

- Brief description and name of each module
- An outline about the functionality of every module

- Interface relationship and dependencies between modules
- Database tables identified along with their key elements
- Complete architecture diagrams along with technology details

Low-Level Design(LLD)

- Functional logic of the modules
- Database tables, which include type and size
- Complete detail of the interface
- Addresses all types of dependency issues
- Listing of error messages
- Complete input and outputs for every module

Phase 4: Coding:

Once the system design phase is over, the next phase is coding. In this phase, developers start build the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.

In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

Phase 5: Testing:

Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system. This is done to verify that the entire application works according to the customer requirement.

During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Phase 6: Installation/Deployment:

Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final

software is released and checked for deployment issues if any.

Phase 7: Maintenance:

Once the system is deployed, and customers start using the developed system, following 3 activities occur

- Bug fixing - bugs are reported because of some scenarios which are not tested at all
- Upgrade - Upgrading the application to the newer versions of the Software
- Enhancement - Adding some new features into the existing software

Technologies Used

1. Java Servlets :

A Java servlet is a Java software component that extends the capabilities of a server. Although servlets can respond to many types of requests, they most commonly implement web containers for hosting web applications on web servers and thus qualify as a server-side servlet web API. Such web servlets are the Java counterpart to other dynamic web content technologies such as PHP and ASP.NET.

Servlets are the Java programs that run on the Java-enabled web server or application server.

They are used to handle the request obtained from the web server, process the request, produce the response, then send response back to the web server.

History of Java Servlets

The Java servlets API was first publicly announced at the inaugural JavaOne conference in May 1996. About two months after the announcements at the conference, the first public implementation was made available on the JavaSoft website. This was the first alpha of the Java Web Server (JWS; then known by its codename *Jeeves*) which would eventually be shipped as a product on June 5, 1997.

Importance of Servlets

- Servlet can be described in many ways, depending on the context.
- Servlet is a technology which is used to create a web application.
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page.

Life Cycle of Servlets

Three methods are central to the life cycle of a servlet. These are init(), service() and destroy(). They are implemented by every servlet and are invoked at specific times by the server.

- During initialization stage of the servlet life cycle, the web container initializes the servlet instance by calling the init() method, passing an object implementing the java.servlet.servletConfig interface. This configuration object allows the servlet to access name-value initialization parameters from the web application.
- After initialization, the servlet instance can service client requests. Each request is serviced in its own separate thread. The web container calls the service() method ofthe servlet for every request. The service() method determines the kind of request being made and dispatches it to an appropriate method to handle the request. The developer of the servlet must provide an implementation for these methods. If a request is made for a method that is not implemented by the servlet, the method of the parent class is called, typically resulting in an error being returned to the requester.

- Finally, the web container calls the destroy() method that takes the servlet out of service. The destroy() method, like init(), is called only once in the lifecycle of a servlet.

Servlet Architecture

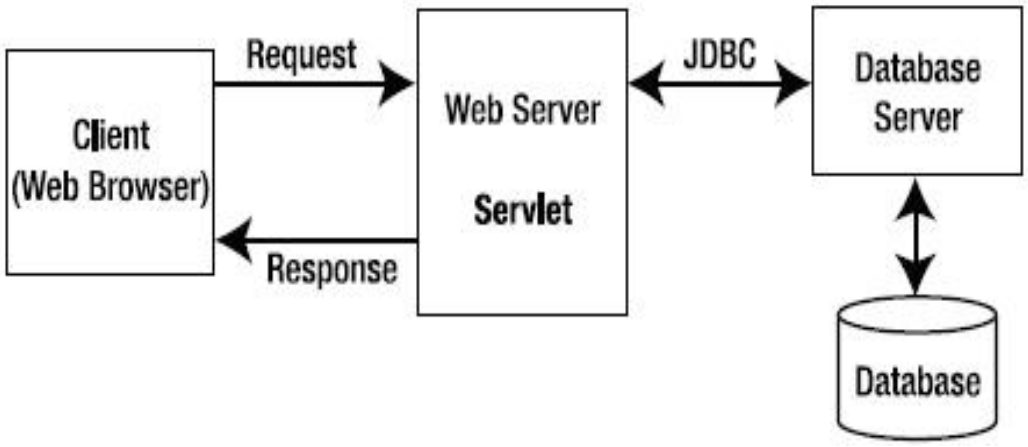


Figure 2. Servlet Architecture

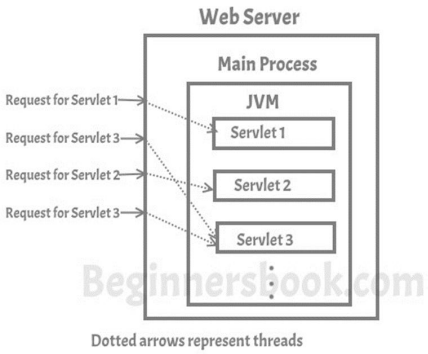
Similar Technology

CGI programs are handled by a new process every time a new request has been made. Unlike CGI, the servlet programs are handled by separate threads that can run concurrently more efficiently.

CGI program can be written in any programming language that makes it mostly platform dependent as not all programming languages are platform independent. Servlet only uses Java as programming language that makes it platform independent and portable. Another benefit of using java is that the servlet can take advantage of the object oriented programming features of java.

How Servlet Works

As I mentioned above that concurrent requests to the server are handled by threads, here is the graphical representation of the same –



Features of Servlets

1. **Portable**:

As I mentioned above that Servlet uses Java as a programming language, Since java is platform independent, the same holds true for servlets. For example, you can create a servlet on Windows operating system that users GlassFish as web server and later run it on any other operating system like Unix, Linux with Apache tomcat web server, this feature makes servlet portable and this is the main advantage servlet has over CGI.

2. **Efficient**:

Once a servlet is deployed and loaded on a web server, it can instantly start fulfilling request of clients. The web server invokes servlet using a lightweight thread so multiple client requests can be fulling by servlet at the same time using the multithreading feature of Java. Compared to CGI where the server has to initiate a new process for every client request, the servlet is truly efficient and scalable.

3. **Robust**:

By inheriting the top features of Java (such as Garbage collection, Exception handling, Java Security Manager etc.) the servlet is less prone to memory management issues and memory leaks. This makes development of web application in servlets secure and less error prone.

Session Tracking in Servlets

Session simply means a particular interval of time. Session Tracking is a way to maintain state (data) of an user. It is also known as session management in servlet.

Http protocol is a stateless so we need to maintain state using session tracking techniques. Each time user requests to the server, server treats the request as the new request. So we need to

maintain the state of an user to recognize to particular user. HTTP is stateless that means each request is considered as the new request.

Session Tracking Techniques

There are four techniques used in Session tracking:

1. Cookies
2. Hidden Form Field
3. URL Rewriting
4. Http Session

Cookies in Servlet

A cookie is a small piece of information that is persisted between the multiple client requests. A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

Types of Cookies

1) **SessionCookies:**

Session cookies do not have expiration time. It lives in the browser memory. As soon as the web browser is closed this cookie gets destroyed.

2) **Persistent Cookies:**

Unlike Session cookies they have expiration time, they are stored in the user hard drive and gets destroyed based on the expiry time.

JDBC Connection in Java

JDBC is an acronym for Java Database Connectivity. It's an advancement for ODBC (Open Database Connectivity). JDBC is an standard API specification developed in order to move data from frontend to backend. This API consists of classes and interfaces written in Java. It basically acts as an interface (not the one we use in Java) or channel between your Java program and databases i.e it establishes a link between the two so that a programmer could send data from Java code and store it in the database for future use.

Steps for connectivity between Java program and database

1. Loading the Driver

To begin with, you first need load the driver or register it before using it in the program . Registration is to be done once in your program.

2. Create the connections

3. Create a statement

Once a connection is established you can interact with the database. The JDBCStatement, CallableStatement, and PreparedStatement interfaces define the methods that enable you to send SQL commands and receive data from your database.

4. Execute the query

Now comes the most important part i.e executing the query. Query here is an SQL Query . Now we know we can have multiple types of queries. Some of them are as follows:

- Query for updating / inserting table in a database.
- Query for retrieving data.

5. Close the connections

So finally we have sent the data to the specified location and now we are at the verge of completion of our task . By closing connection, objects of Statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

2. MySQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

- MySQL is a database management system.

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- MySQL databases are relational.

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

  The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

  SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- MySQL software is Open Source.

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.

- The MySQL Database Server is very fast, reliable, scalable, and easy to use.

  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

  MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- MySQL Server works in client/server or embedded systems.

  The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

  We also provide MySQL Server as an embedded multithreaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

- A large amount of contributed MySQL software is available.

  MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

How MySQL works

MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into seperate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer.

MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

Reasons to choose MySQL:

- **Secure Money Transactions**

  MySQL transactions work as a single unit, which means unless and until every individual operational stage is successfully completed, the transaction is not cleared. So, if an operation fails at any stage, the entire transaction happening within that group fails. MySQL ensures that financial transactions have data integrity, so customers can make worry-free transactions online. The money is not debited until the entire process is completed and in case of failure, every process is reverted to the previous stage.

- **On-Demand Scalability**

  MySQL comes with the advantage of unmatched flexibility that facilitates efficient management of deeply embedded applications, even in gigantic data centers that stack tremendous amounts of mission-critical information. It enables complete customization to cater to the unique requirements of eCommerce businesses with a much smaller footprint. MySQL provides ultimate platform flexibility to enterprises who need additional features and functionalities for their database servers.

- **High Availability**

  Consistent availability is the stalwart feature of MySQL – enterprises that deploy it can enjoy round-the-clock uptime. MySQL comes with a wide variety of cluster servers and master-slave replication configurations that enable instant failover for uninterrupted access. Whether you run an eCommerce website or a high-speed processing system, MySQL is designed to process millions of queries and thousands of transactions while ensuring unique memory caches, full-text indexes and optimum speed.

- **Rock-Solid Reliability**

  Protecting sensitive business information is the primary concern of every enterprise. MySQL ensures data security with exceptional data protection features. Powerful data encryption prevents unauthorized viewing of data and SSH and SSL supports ensure safer connections. It also features a powerful mechanism that restricts server access to authorized users and has the ability to block users even at the man-machine level. Finally, the data backup feature facilitates point-in-time recovery.

- **Quick-Start Capability**

  You can go from software download to complete installation in just 15 minutes. MySQL is exceptionally quick, regardless of the underlying platform. It features self-management capabilities like auto restart, space expansion and automatic configuration changes for ease of management. It also comes with a comprehensive set of migration tools and a fully loaded graphical management suite. MySQL enables real-time performance monitoring for timely troubleshooting of operational issues from a single workstation.

- For all of these reasons, organizations are using MySQL to instantly develop and launch apps. From retail and finance, to healthcare and manufacturing, many industries are capitalizing on the cost-effectiveness, efficiency and reliability of MySQL to deliver seamless services and boost their revenue.

Apache Tomcat Server

Apache Tomcat is an open source Web server tool developed by the Apache Software Foundation (ASF). It is one of many Apache-related open source products used by IT professionals for various tasks and objectives.

Apache Tomcat allows the implementation of Java Servlets and JavaServer Pages (JSP) to promote an effective Java server environment. Users can also access resources for configuration and use extensible markup language (XML) to configure projects. Successive versions of Apache Tomcat have solved different problems by applying software patches and other solutions. Some experts characterize Apache Tomcat as a product offering a runtime shell for Java Servlets. Users can also set up Java virtual machines (JVM) to configure virtual hosting.

Tomcat is an application server from the Apache Software Foundation that executes Java servlets and renders Web pages that include Java Server Page coding. Described as a "reference implementation" of the Java Servlet and the Java Server Page specifications, Tomcat is the result of an open collaboration of developers and is available from the Apache Web site in both binary and source versions. Tomcat can be used as either a standalone product with its own internal Web server or together with other Web servers, including Apache, Netscape Enterprise Server, Microsoft Internet Information Server (IIS), and Microsoft Personal Web Server. Tomcat requires a Java Runtime Enterprise Environment that conforms to JRE 1.1 or later.

Tomcat is one of several open source collaborations that are collectively known as Jakarta. Born out of the Apache Jakarta Project, Tomcat is an application server designed to execute Java servlets and render web pages that use Java Server page coding. Accessible as either a binary or a source code version, Tomcat's been used to power a wide range of applications and websites across the Internet. At the time of writing, it's definitely one of the more popular servlet containers available.

Reasons to choose Tomcat Server

- It's Incredibly Lightweight

  Even with JavaEE certification, Tomcat is an incredibly lightweight application. If offers only the most basic functionality necessary to run a server, meaning it provides relatively quick load and redeploy times compared to many of its peers, which are bogged down with far too many bells and whistles. This lightweight nature also allows it to enjoy a significantly faster development cycle.

- Its an Open Source

  Open-source always counts as a win. Tomcat's free, and the source code for the server is readily available to anyone who'd care to download it. What this means is that – assuming you're willing to tinker with the moving parts of your server – you've got an incredible degree of freedom insofar as what you want to do with a Tomcat installation.

- It's Highly Flexible

  Thanks to its lightweight nature and a suite of extensive, built-in customization options, Tomcat is quite flexible. You can run it in virtually any fashion you choose, and it'll still work as intended. The fact that it's open-source helps as well, since you can tweak it to fit your needs, provided you've the knowledge to do so.

- Your Server Will Be More Stable

  Tomcat is an extremely stable platform to build on – and using it to run your applications will contribute to your server's stability, as well. This is because Tomcat runs independently of your Apache installation – even if a significant failure in Tomcat caused it to stop working, the rest of your server would run just fine.

- It Offers An Extra Level Of Security

  Many organizations choose to position their Tomcat installation behind an extra firewall, accessible only from the Apache installation. In short, depending on how you implement

your Tomcat installation, it can add an extra layer of security to your server – which is never a bad thing.

Similar Technology

GlassFish is the Open Source reference implementation for a Java EE application server. In addition to being an open source reference implementation of Java EE application server; GlassFish comes packed with core Java EE technologies such as: Servlets, Enterprise Java Beans (EJBs), Java Persistence API (JPA), JavaServer Faces (JSF), Java Message Service (JMS) as well as the default Java EE SDK.

Furthermore, in addition to being a Java EE application server, GlassFish handles EJB requests thus is also an EJB Container. Also, essentially it has its own web container (a derivative of Tomcat) and thus shares the same Catalina servlet container with Tomcat.

3.JSP

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.JSP is a server side technology of programming which is used in order to create the dynamic web pages. Like the servlets, JSP also have an access to different APIs of Java.In a java server page, the java code is inserted inside HTML content. JSP is an extended form of servlet. The page of JSP consists of the HTML tags, JSP tags and scripting elements. The JSP page is able to return the type of content including static and dynamic as the requirement in response to the request.The static content can be in the form of HTML, XML and text.
The dynamic content can be in the form of custom tags, java beans and code of java.There is a

container in the web server that is required in order to process the pages of JSP. The container of JSP basically works with the web server in order to provide the related services which this type of page requires.

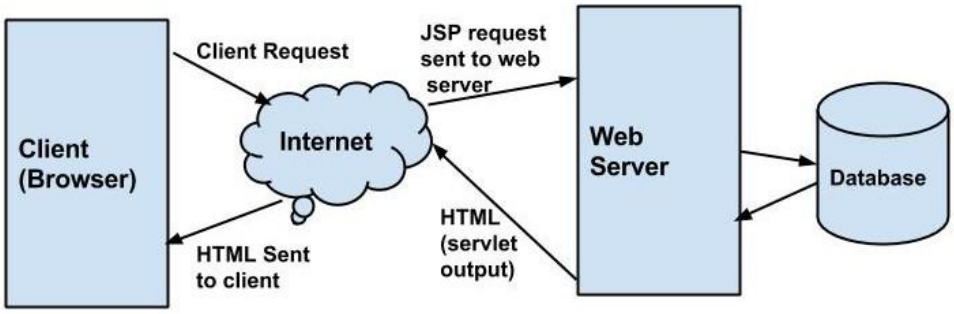


Figure 3. JSP Architecture

Advantages of JSP over Servlet

There are many advantages of JSP over the Servlet. They are as follows:

1) Extension to Servlet

JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

2) Easy to maintain

JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.

3) Fast Development: No need to recompile and redeploy

If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

4) Less code than Servlet

In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

The Lifecycle of a JSP Page

The JSP pages follow these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (the container invokes jspInit() method).
- Request processing (the container invokes _jspService() method).
- Destroy (the container invokes jspDestroy() method).

Importance:

The servlet gets invoked for each request and the full page of JSP can be translated once into a servlet in an easy manner. The pages of JSP are translated to servlets and the servlets runs when the request comes and client is not able to see anything related to JSP. In order to understand the working behind JSP, there is a need to first have a complete understanding about servlets.

The main advantage of JSP is that the static and dynamic content gets separated i.e., the static content includes the design of a webpage and the dynamic content includes the business logic which helps in proper understanding and the page can be maintained in an easy manner.JSP is easy to program and it basically provides the facility in order to develop the dynamic web pages and uses the Java programming language and it consists of scripting and different tags.

How do Java and Servlets work together?

JSP has two phases - Translation Phase and Execution Phase. Technically, a JSP is indeed converted as a Servlet at the backend during the translation phase.

JSP is used as a front end technology for displaying the content/ output to the user with its ease

of use syntax mixed with HTML. A Servlet is mostly used as a controller - which as a central component decides the routing of requests - more like a receptionist picking up the phone for incoming calls and routing to the concerned person whom the actual caller wants to reach out to.

There are ways in which you can mix and match these two. Means, you can produce the HTML output in Servlet and write the business/control logic in JSP itself. However this is discouraged due to the fact the maintenance (or any changes in future) is very difficult. Hence, a better approach is suggested by using MVC - Model View Controller Architecture. It is also a design pattern which helps you to achieve a clear separation between your business logic and the presentation so that any change happened in one layer does not affect much on the other layers.

4. HTML

HTML stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most of markup (e.g. HTML) languages are human readable. Language uses tags to define what manipulation has to be done on the text.
HTML is a markup language which is used by the browser to manipulate text, images and other content to display it in required format. HTML was created by Tim Berners-Lee in 1991. The first ever version of HTML was HTML 1.0 but the first standard version was HTML 2.0 which was published in 1999.

Elements and Tag:
HTML uses predefined tags and elements which tells the browser about content display property. If a tag is not closed then browser applies that effect till end of page.

HTML page structure:

The Basic structure of HTML page is given below. It contain some elements like head, title, body, … etc. These elements are used to build the blocks of web pages.

<DOCTYPE! html>: This tag is used to tells the HTML version. This currently tells that the version is HTML 5.

<html>: This is called HTML root element and used to wrap all the code.

<head>: Head tag contains metadata, title, page CSS etc. All the HTML elements that can be used inside the <head> element are:

- <style>
- <title>
- <base>
- <noscript>
- <script>
- <meta>
- <title>

<body>: Body tag is used to enclosed all the data which a web page has from texts to links. All of the content that you see rendered in the browser is contained within this element.

Features of HTML:

- It is easy to learn and easy to use.
- It is platform independent.
- Images, video and audio can be added to a web page.
- Hypertext can be added to text.
- It is a markup language.

Why HTML?

- It is a simple markup language. Its implementation is easy.
- It is used to create a website.
- Helps in developing fundamentals about web programming.
- Boost professional career.

Advantages:

- HTML is used to build a websites.
- It is supported by all browsers.
- It can be integrated with other languages like CSS, JavaScript etc.

Disadvantages:

- HTML can create only static webpages so for dynamic web page other languages have to be used.
- Large amount of code has to be written to create a simple web page.
- Security feature is not good.

5. CSS

Cascading **S**tyle **S**heets, fondly referred to as **CSS**, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

CSS is easy to learn and understood but it provides powerful control over the presentation of an HTML document.

WHY CSS?

- **CSS saves time :** You can write CSS once and reuse same sheet in multiple HTML pages.

- **Easy Maintainence :** To make a global change simply change the style, and all elements in all the webpages will be updated automatically.

- **Search Engines :** CSS is considered as clean coding technique, which means search engines won't have to struggle to "read" its content.

- **Superior styles to HTML :** CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

- **Offline Browsing :** CSS can store web applications locally with the help of offline catche.Using of this we can view offline websites.

CSS Syntax

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document.

A style rule set consists of a selector and declaration block.

```
Selector => h1
Declaration => {color:blue;font size:12px;}
```

- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.

For Example:

-> color is property and blue is value.

-> font size is property and 12px is value.

- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

6. JavaScript

JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and mobile application development.

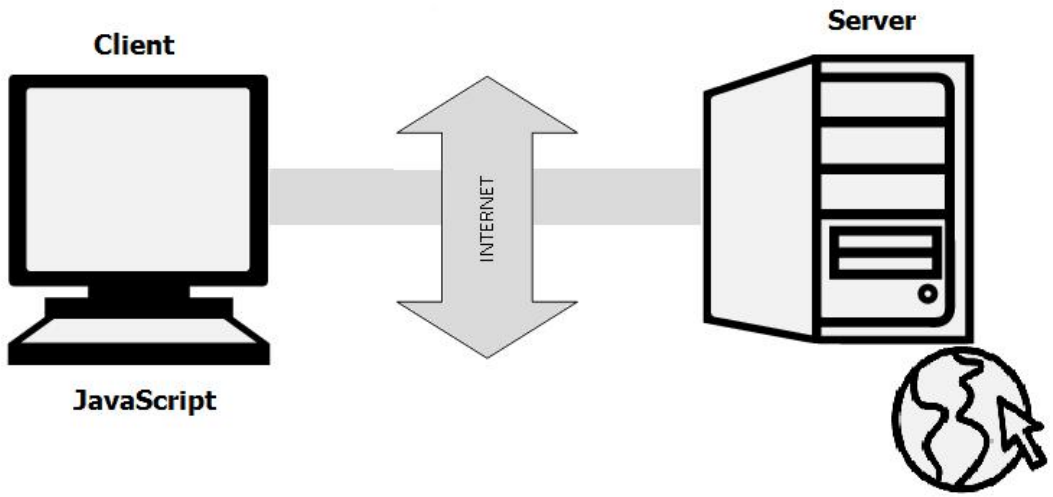


Figure 4 . Javascript Architecture

Javascript History

JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time.

The language was initially called LiveScript and was later renamed JavaScript. There are many programmers who think that JavaScript and Java are the same. In fact, **JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language**. The syntax of JavaScript is mostly influenced by the programming language C.

How to Run JavaScript?

Being a scripting language, **JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code**. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that **all modern web browsers support** JavaScript. So, you do not have to worry about whether your site visitor uses Internet Explorer, Google Chrome, Firefox or any other browser. JavaScript will be supported. Also, JavaScript **runs on any operating system** including Windows, Linux or Mac. Thus, JavaScript overcomes the main disadvantages of VBScript (Now deprecated) which is limited to just IE and Windows.

Tools You Need

To start with, you need a text editor to write your code and a browser to display the web pages you develop. You can use a text editor of your choice including Notepad++, Visual Studio Code, Sublime Text, Atom or any other text editor you are comfortable with. You can use any web browser including Google Chrome, Firefox, Microsoft Edge, Internet Explorer etc.

JavaScript

- JavaScript is a **client-side scripting language** developed by Brendan Eich.
- JavaScript can be **run on any operating systems** and almost all web browsers.
- You need a text editor to write JavaScript code and a browser to display your web page.

7. ChatterBot

ChatterBot is a Python library that makes it easy to generate automated responses to a user's input. ChatterBot uses a selection of machine learning algorithms to produce different types of responses. This makes it easy for developers to create chat bots and automate conversations with users.

Figure 5 . ChatterBot

Language Independence:

The language independent design of ChatterBot allows it to be trained to speak any language. Additionally, the machine-learning nature of ChatterBot allows an agent instance to improve it's own knowledge of possible responses as it interacts with humans and other sources of informative data.

How ChatterBot Works:

ChatterBot is a Python library designed to make it easy to create software that can engage in conversation. An untrained instance of ChatterBot starts off with no knowledge of how to communicate. Each time a user enters a statement, the library saves the text that they entered and the text that the statement was in response to. As ChatterBot receives more input the number of responses that it can reply and the accuracy of each response in relation to the input statement increase. The program selects the closest matching response by searching for the closest matching known statement that matches the input, it then chooses a response from the selection of known responses to that statement.

8. Mail API

- The **JavaMail** is an API that is used to compose, write and read electronic messages (emails).
- The JavaMail API provides protocol-independent and plateform-independent framework for sending and receiving mails.
- The **javax.mail** and **javax.mail.activation** packages contains the core classes of JavaMail API.

The JavaMail facility can be applied to many events. It can be used at the time of registering the user (sending notification such as thanks for your interest to my site), forgot password (sending password to the users email id), sending notifications for important updates etc. So there can be various usage of java mail api.

JavaMail Architecture

The java application uses JavaMail API to compose, send and receive emails. The JavaMail API uses SPI (Service Provider Interfaces) that provides the intermediatory services to the java application to deal with the different protocols.

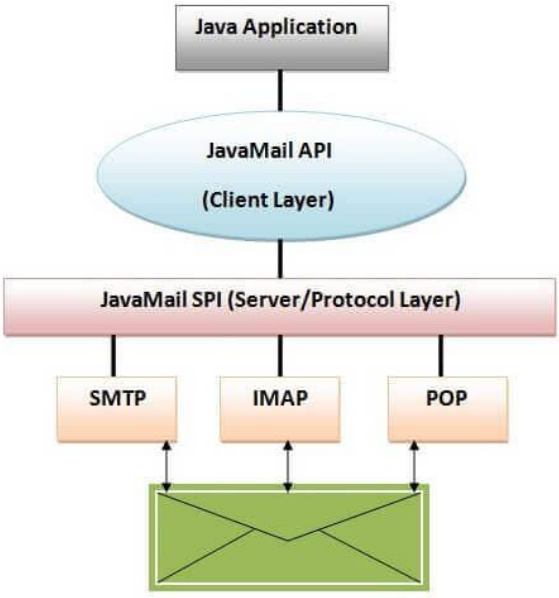


Figure 6 . Java Mail Architecture

The Dependency to be added in POM.XML file (for Maven Project) is:

```
<dependency>
        <groupId>javax.mail</groupId>
        <artifactId>mail</artifactId>
        <version>1.5.0-b01</version>
</dependency>
```

4. SYSTEM DESIGN

4.1 INTRODUCTION TO UML

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

1. User Model View

i. This view represents the system from the users' perspective.

ii. The analysis representation describes a usage scenario from the end-users' perspective.

2. Structural Model View

i. In this model, the data and functionality are arrived from inside the system.

ii. This model view models the static structures.

3. Behavioural Model View It represents the dynamic of behavioural as parts of the system, depicting he interactions of collection between various structural elements described in the user model and structural model view.

4. Implementation Model View In this view, the structural and behavioural as parts of the system are represented as they are to be built.

5. Environmental Model View In this view, the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

4.2 UML Diagrams

4.2.1 Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running /operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application.

A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified.

In brief, the purposes of use case diagrams can be as follows:

a. Used to gather requirements of a system.
b. Used to get an outside view of a system.
c. Identify external and internal factors influencing the system.
d. Show the interacting among the requirements are actors.

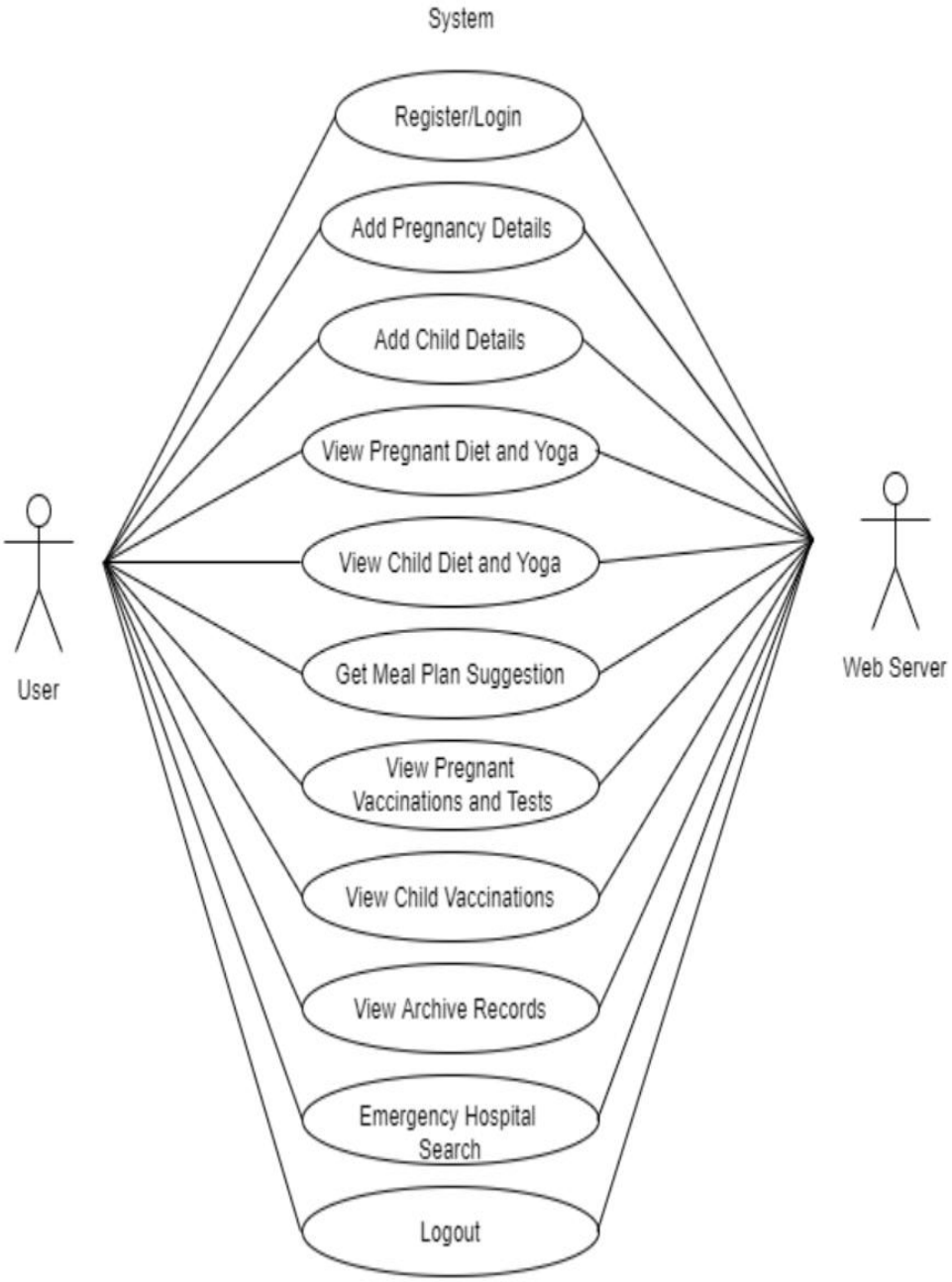


Figure 7 : Use case Diagram

4.2.2 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

The aim of a sequence diagram is to define event sequences, which would have a desired outcome. The focus is more on the order in which messages occur than on the message per se. However, the majority of sequence diagrams will communicate what messages are sent and the order in which they tend to occur.

Basic Sequence Diagram Notations

- **Class Roles or Participants**

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

- **Activation or Execution Occurrence**

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.

- **Messages**

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

- **Lifelines**

Lifelines are vertical dashed lines that indicate the object's presence over time.

- **Destroying Objects**

Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

- **Loops**

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [].

- **Guards**

When modelling object interactions, there will be times when a condition must be met for a message to be sent to an object. Guards are conditions that need to be used throughout UML diagrams to control flow.

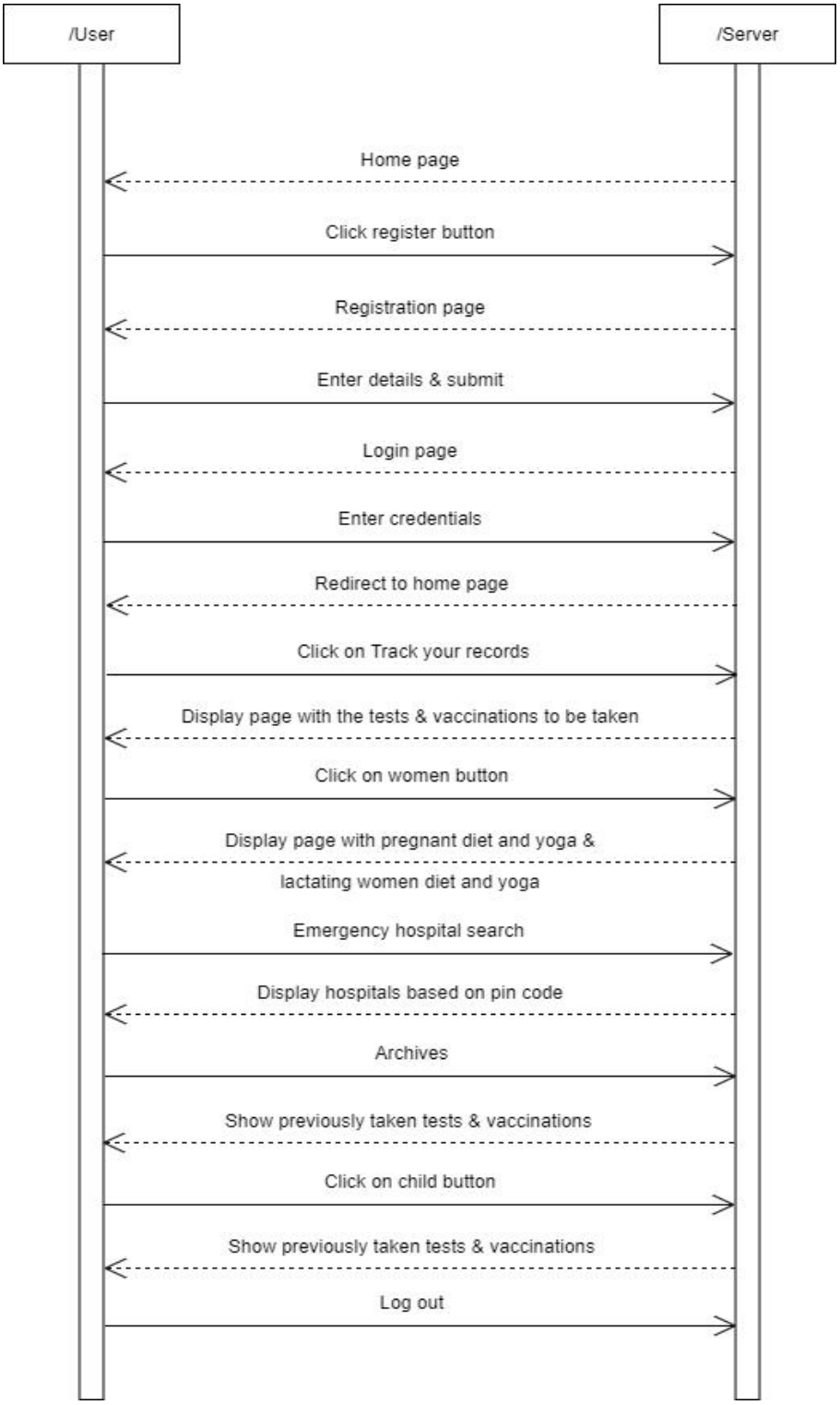


Figure 8 : Sequence Diagram

4.2.3 Class Diagram

Class diagrams are the main building blocks of every object oriented methods. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them.

The main purpose to use class diagrams are:

- This is the only UML which can appropriately depict various aspects of OOPs concept.
- Proper design and analysis of application can be faster and efficient.
- It is base for deployment and component diagram.

Each class is represented by a rectangle having a subdivision of three compartments name, attributes and operation.

There are three types of modifiers which are used to decide the visibility of attributes and operations.

+ is used for public visibility (for everyone)

is used for protected visibility (for friend and derived)

– is used for private visibility (for only me)

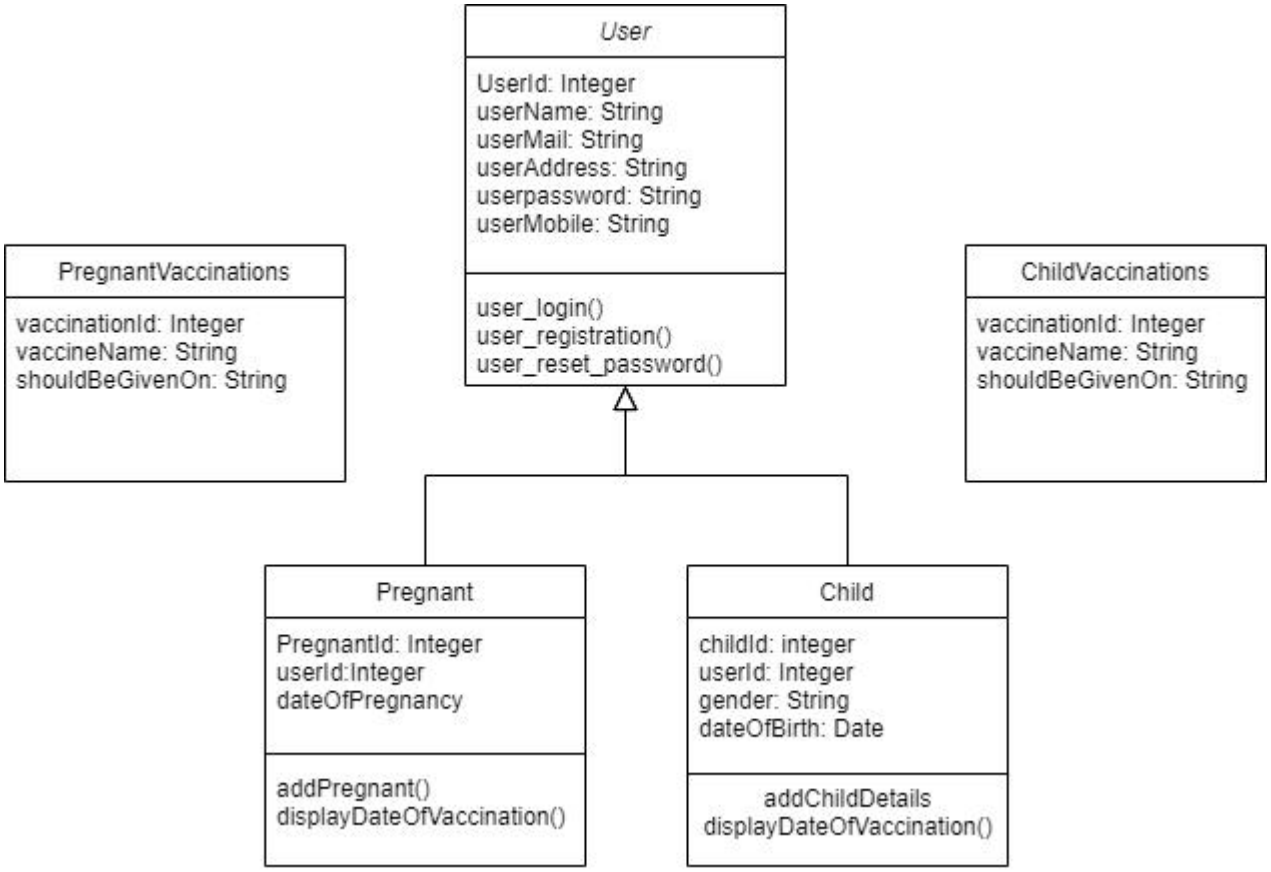


Figure 9 : Class Diagram

5. IMPLEMENTATION

PSEUDO-CODE

Step 1: In the homepage, the user login, registration as well as Information regarding Immunization, women and child health and nutrition are provided. After click on the required Field.

Step 2:

- Suppose If user login is selected

The required details are retrieved and authenticated with the database details, which are stored during registration process.

- Suppose If user Registration is selected

Name, Mobile number, Address, Pincode, Passwords(2 for verifying), Has children? (If yes, their name, gender and DOB), Is Pregnant? (If yes, Start date of Pregnancy). The user gets OTP to verify and after verification he gets redirected to Login Page.

Step 3:

- If the user Mail-Id is not found in the database, we redirect him to the User Registration Page.
- Else if the Password is incorrect we redirect the User to the Login Page.
- Else if user is correctly authenticated then we redirect user to the Home Page where he can choose what services are needed by him.

Step 4:

- In the user Home page he/she can see their details regarding their children – Vaccinations along with their respective dates and their pregnancy- Vaccinations, Visits, Tests along with their respective dates and Daily Meal Plan.
- Information about Women- pregnant and lactating mothers Nutrition diet to be taken and Exercises to be done.
- Information about Child- child and adolescent children's Nutrition diet to be taken and Exercises to be done.
- Archive section in which they can review their previous pregnancy details.

- Emergency Search in which they can search hospitals by giving their PinCode.
- Profile section in which they can add children and Pregnancy details

Step 5:

An E-Mail is sent to the User– Regarding their children and pregnancy vaccinations and check-ups to be done on given date. Thanking him for choosing our services.

Step 6:

Logout option is provided to the user on right most side of Navigation Bar.

After clicking logout button, he is redirected to the Home Page of the Website.

5.1 SAMPLE CODE

MySQL: The Database and Tables are created.

Database creation:

```
create database poshanabhiyaan;
use poshanabhiyaan;
```

User Table is Created :

```
create table user(
userId int auto_increment primary key,
userName varchar(20),
userMail varchar(30),
userPswd varchar(20),
userMobile varchar(10),
userAddress varchar(50),
userPinCode varchar(6)
);
```

Child Table creation:

```
create table child(
cId int auto_increment primary key,
cName varchar(20),
cGender varchar(6),
cDob date,
parent int,
foreign key(parent) references user(userId),
day0 date,
day42 date,
day71 date,
day99 date,
day472 date,
day1780 date,
day3560 date,
day4300 date
);
```

Child Vaccination Table Creation and Insertion:

```
create table childvaccinations(
day0 varchar(100),
day42 varchar(100),
day71 varchar(100),
day99 varchar(100),
day274 varchar(100),
day472 varchar(100),
day1780 varchar(100),
day3560 varchar(100),
day4300 varchar(100)
);

insert into childvaccinations values
("Oral Polio Vaccine,Hepatitis B,Bacille Calmette Guerin-BCG",
"Oral Polio Vaccine-1,Pentavalent-1,Fractional Dose of Inactivated Polio Vaccine-fIPV-1,Rotavirus-RV1",
"Oral Polio Vaccine-2,Pentavalent-2,Rotavirus-RV2",
"Oral Polio Vaccine-3,Pentavalent-3,Fractional Dose of Inactivated Polio Vaccine-fIPV-2",
"measles mumps and rubella-MMR,Typhoid,Vitamin-A",
"Diphtheria Pertussis Tetanus-Booster1,Oral Polio Vaccine-Booster,Hepatits-A",
"Diphtheria Pertussis Tetanus-Booster2",
"tetanus toxoid",
"Human papillomavirus (HPV) vaccines"
);
```

Pregnant Person Table Creation:

```
create table pregnantPerson(
ppid int8 primary key auto_increment,
userid int,
startDateOfPreg date,
day31Date date,
day31Status varchar(20),
day61Date date,
day61Status varchar(20),
day91Date date,
day91Status varchar(20),
day121Date date,
day121Status varchar(20),
day151Date date,
day151Status varchar(20),
day181Date date,
day181Status varchar(20),
day196Date date,
day196Status varchar(20),
day211Date date,
day211Status varchar(20),
day226Date date,
day226Status varchar(20),
day241Date date,
day241Status varchar(20),
day248Date date,
day248Status varchar(20),
day255Date date,
day255Status varchar(20),
day261Date date,
day261Status varchar(20),
foreign key(userid) references user(userId)
);
```

Pregnant Tests Table creation and Insertion:

```
create table pregnantTests(
day varchar(10) primary key,
month varchar(10),
tests varchar(255)
);

insert into pregnantTests values('day31','month2', 'Routine prenatal check up, Blood and urine test (ANC profile), Urine pregnancy test, Pr
insert into pregnantTests values('day61','month3', 'Routine prenatal check up,NT ultrasound scan (sonography) and a double marker test,Firs
insert into pregnantTests values('day91','month4', 'Routine prenatal check up');
insert into pregnantTests values('day121','month5', 'Routine prenatal check up, Anomaly or ultrasound level II scan');
insert into pregnantTests values('day151','month6', 'Routine prenatal check up, First dose of Tetanus Toxoid (TT) injection');
insert into pregnantTests values('day181','month7', 'Routine prenatal check up, Second dose of Tetanus Toxoid (TT) injection, Growth and fe
insert into pregnantTests values('day196','month7', 'Routine prenatal check up');
insert into pregnantTests values('day211','month8', 'Routine prenatal check up');
insert into pregnantTests values('day226','month8', 'Routine prenatal check up');
insert into pregnantTests values('day241','month9', 'Routine prenatal check up');
insert into pregnantTests values('day248','month9', 'Routine prenatal check up');
insert into pregnantTests values('day255','month9', 'Routine prenatal check up');
insert into pregnantTests values('day261','month9', 'Routine prenatal check up');
```

Sending Mail using Mail - API dependency:

```
public class SendingMail {
    String otp;
    Properties emailProperties;
    Session mailSession;
    MimeMessage emailMessage;
    public void setMailServerProperties() {
        String emailPort = "587";// gmail's smtp port
        emailProperties = System.getProperties();
        emailProperties.put("mail.smtp.port", emailPort);
        emailProperties.put("mail.smtp.auth", "true");
        emailProperties.put("mail.smtp.starttls.enable", "true")
    }
    public void createEmailMessage(String toEmail, String otp) throws AddressException, MessagingException {
        String emailSubject = "Poshan Abhiyaan";
        String emailBody = otp+" Thank you for choosing us. This is an email sent by poshanabhiyaan.";
        mailSession = Session.getDefaultInstance(emailProperties, null);
        emailMessage = new MimeMessage(mailSession);
        emailMessage.addRecipient(Message.RecipientType.TO, new InternetAddress(toEmail));
        emailMessage.setSubject(emailSubject);
        emailMessage.setContent(emailBody, "text/html");
    }
    public void sendEmail() throws AddressException, MessagingException {
        String emailHost = "smtp.gmail.com";
        String fromUser = "minimajorp";// just the id alone without @gmail.com
        String fromUserEmailPassword = "Kmit123$";
        Transport transport = mailSession.getTransport("smtp");
        transport.connect(emailHost, fromUser, fromUserEmailPassword);
        transport.sendMessage(emailMessage, emailMessage.getAllRecipients());
        transport.close();
        System.out.println("Email sent successfully.");
    }
```

Background Job Manager:

```
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;
import javax.servlet.annotation.WebListener;
@WebListener
public class BackgroundJobManager implements ServletContextListener {
    private ScheduledExecutorService scheduler;
    @Override
    public void contextInitialized(ServletContextEvent event) {
        scheduler = Executors.newSingleThreadScheduledExecutor();
        scheduler.scheduleAtFixedRate(new SomeDailyJob(), 0, 1, TimeUnit.DAYS);
    }
    @Override
    public void contextDestroyed(ServletContextEvent event) {
        scheduler.shutdownNow();
    }
}
```

In SomeDailyJob() , we are writing code to send emails regarding child vaccinations and pregnant women vaccinations and check-ups. This job happens daily, it checks whether their vaccination date is today, tomorrow or the day after tomorrow and reminds them.

User Registration HTML:

```
<form action="UserRegistration" method="get">
    <table>
        <tr>
            <td>Name:</td>
            <td><input type="text" name="userName" required></td>
        </tr>
        <tr>
            <td>Mail ID:</td>
            <td><input type="text" name="userMail" id="userMail"
                onkeyup="validateEmail(document.getElementById('userMail').value)"
                required>  <span
                style="color: red; font-size: smaller" id="mailValidation"></span>
            </td>
        </tr>
        <tr>
            <td>Phone Number:</td>
            <td><input type="text" name="userPhone" required></td>
        </tr>
        <tr>
            <td>Address:</td>
            <td><textarea type="text" name="userAddress" placeholder="Enter Address" style="height:60px;width:200px" required></textarea></td>
        </tr>
        <tr>
            <td>PinCode:</td>
            <td><input type="text" name="userPinCode" placeholder="Enter Pin Code" id="userPinCode" onkeyup="validatePinCode(document.getElementById('userPinCode').
             <span style="color:red;font-size:smaller" id="pinCodeValidation"></span> <br></td>
        </tr>
        <tr>
            <td>Set Password:</td>
            <td><input type="password" name="userPswd"
                placeholder="Enter Password" id="userPswd"
                onkeyup="validatePassword(document.getElementById('userPswd').value)"
                required>  <span
                style="color: red; font-size: smaller" id="pswdValidation"></span>
                <br></td>
        </tr>
    <tr>
        <td>Confirm Password:</td>
        <td><input type="password" name="userPswd"
            placeholder="Re-Enter Password" id="userPswd2" required><br>
        </td>
    </tr>
    <tr>
        <td><br /><br />Do you have children?:</td>
        <td>Yes:<input type="radio" name="children" value="yes"
            id="yesCheck" onclick="javascript:yesnoCheck();" required>
            <div id="ifYes" style="display: none">
                No.of Children: <input type="text" name="noofchildren"
                    id="noofchildren"
                    placeholder="maximum 3 children are only allowed"
                    onkeyup="javascript:giveChildren();">
                <div id="childInfo">

                </div>
            </div>
        </td>
    </tr>
    <tr>
        <td></td>
        <td>No:<input type="radio" name="children" value="no"
            id="noCheck" onclick="javascript:yesnoCheck();" required>
            <div id="ifNo" style="display: none"></div>
        </td>
    </tr>
    <tr>
        <td><br /><br />Are you Pregnant?:</td>
        <td>Yes:<input type="radio" name="pregnant" value="yes"
            id="yesPregCheck" onclick="javascript:pregnantCheck();" required>
            <div id="ifYesPreg" style="display: none">
                Start Date of Pregnancy: <input type="date" name="pregnancyDate"
                    id="pregnancyDate">
            </div>
        </td>
    </tr>
```

```
        <tr>
            <td></td>
            <td><br /> <input
                style="color: blanchedalmond; font-size: large; height: 40px; background-color: #0099cc; align-content: center"
                type="submit" name="submit" value="Register"
                onmouseover="validatePasswords(document.getElementById('userPswd').value, document.getElementById('userPswd2').value),validateForm()"><br>
                <br></td>
        </tr>
    </table>
</form>
```

User Registration Servlet:

```
String userName = request.getParameter("userName");
String userMail = request.getParameter("userMail");
String userPhone = request.getParameter("userPhone");
String userAddress = request.getParameter("userAddress");
String userPinCode = request.getParameter("userPinCode");
String userPswd = request.getParameter("userPswd");

SendingMail sm = new SendingMail();
String generatedotp = sm.getRandom();
try {
    sm.setMailServerProperties();
    sm.createEmailMessage(userMail, "Your OTP is:" + generatedotp);
    sm.sendEmail();
} catch (AddressException e) {
    System.out.println("Enter valid address mail id");
    e.printStackTrace();
    RequestDispatcher rd = request.getRequestDispatcher("UserRegistration.html");
    rd.forward(request, response);
    return;
} catch (MessagingException e) {
    e.printStackTrace();
}
RequestDispatcher rd = request.getRequestDispatcher("EnterOTP.html");
rd.forward(request, response);
```

- **User entry into database**

```
if (enteredotp.equals(generatedotp)) {
            try {
                Class.forName("com.mysql.jdbc.Driver");
                Connection con = DriverManager.getConnection(
                        "jdbc:mysql://localhost:3306/poshanabhiyaan?autoReconnect=true&useSSL=false", "root", "root");
                PreparedStatement stmt = con.prepareStatement("select * from user where userMail=?");
                stmt.setString(1, userMail);
                if (stmt.executeQuery().next()) {
                    out.print("<html><body><h3>Already Registered with Provided Mail ID</h3></body></html>");
                    RequestDispatcher rd = request.getRequestDispatcher("UserLogin.html");
                    rd.forward(request, response);
                    return;
                }
                PreparedStatement pstmt = con.prepareStatement(
                        "insert into user(userName,userMail,userPswd,userMobile,userAddress,userPinCode) values(?,?,?,?,?,?);");
```

- **Children entry into database**

```
if (children != null && children.equalsIgnoreCase("yes")) {
    String noOfChildren = (String) session.getAttribute("noofchildren");
    for (int i = 1; i <= Integer.parseInt(noOfChildren); i++) {
        PreparedStatement pstmt3 = con
                .prepareStatement("insert into child(cName,cGender,cDob,parent,day0,day42,day71,day99,day472,day1780,day3560,day4300) value
        int rs3 = pstmt3.executeUpdate();
        if (rs3 > 0) {
            System.out.println("Child entry entered into db");
        }
```

- **Pregnancy entry into database**

```
if(pregnancyDate!=null && pregnancyDate != "") {
    PreparedStatement pstmtPreg = con
            .prepareStatement("insert into pregnantPerson(userid, startDateOfPreg,day31Date, day61Date, day91Date, day121Date, day151Date, day181Date, day1
    pstmtPreg.setInt(1, userId);
    pstmtPreg.setString(2, pregnancyDate);
    int i=0;
    for(i = 0;i< alertDates.length ; i++) {
        pstmtPreg.setString(i+3, alertDates[i]);
        System.out.println(i+3 +" "+alertDates[i]);
    }
    pstmtPreg.setString(16, status);
    int rs3 = pstmtPreg.executeUpdate();
    if (rs3 > 0) {
        System.out.println("Pregnant entry entered into db");
    }
```

- **To check whether it is an archive entry**

```
LocalDate SystemDate = java.time.LocalDate.now();
LocalDate lastDate = LocalDate.parse(alertDates[alertDates.length-1]);
long lastDateToToday = ChronoUnit.DAYS.between(SystemDate, lastDate);
String status="false";
if(lastDateToToday<0) {
    status="true";
}
```

User Login HTML:

```
<form method="get" action="UserLogin"><br><br>
        <div class="imgcontainer">
                <img src="login.jpg"    >
            </div><br><br>
            <div class="container">
 <center> <b style="font-size: 16pt;"> Mail ID :  </b> <input style="font-size: 16pt; height: 40px; width: 280px;" type="text" name="userMail" placeholder="Enter Mail"><
 <center> <b style="font-size: 16pt;"> Password : </b> <input style="font-size: 16pt; height: 40px; width: 280px;" type="password" name="userPswd" placeholder="Enter Pas
  <center><input style="color: blanchedalmond; font-size: large; height: 40px; width:90px; background-color: #0099cc; align-content: center"
    type="submit" value="LOGIN"></center><br><br>
</div>
</form>
<div>
 <center> <a href="ForgotPassword.html">Forgot Password?</a></center>
</div><br/>
<div>
 <center>Not Registered Yet!!
  <a href="UserRegistration.html">Register Here</a></center>
</div></div></div>
```

User Login Servlet:

```
String userMail = request.getParameter("userMail");
String userPswd = request.getParameter("userPswd");
System.out.println(userMail+" "+userPswd);
    try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/poshanabhiyaan?autoReconnect=true&useSSL=false", "root", "root");
    PreparedStatement pstmt = con.prepareStatement("select * from user where userMail=?");
    pstmt.setString(1, userMail);
    ResultSet rs = pstmt.executeQuery();
    if (rs.next() == false) {
        System.out.println("First Register");
        RequestDispatcher rd = request.getRequestDispatcher("UserRegistration.html");
        rd.include(request, response);
    } else {
        do {
            pswd = rs.getString("userPswd");
            userMobile = rs.getString("userMobile");
            userName=rs.getString("userName");
            userId=rs.getInt("userId");
            userPinCode=rs.getString("userPinCode");
        } while (rs.next());
        if (userPswd.equals(pswd)) {
            RequestDispatcher rd = request.getRequestDispatcher("UserHomePage");
            rd.forward(request, response);
        } else {
            System.out.println("Wrong Password");
            RequestDispatcher rd = request.getRequestDispatcher("UserLogin.html");
            rd.include(request, response);
        }
```

Forgot Password Servlet:

```
PreparedStatement pstmt = con.prepareStatement("select * from user where userMail=?");
            pstmt.setString(1, userMail);
            ResultSet rs = pstmt.executeQuery();
            if (rs.next() == false) {
                RequestDispatcher rd = request.getRequestDispatcher("UserRegistration.html");
                rd.include(request, response);
            } else {
                SendingMail sm=new SendingMail();
                String userPswd=rs.getString("userPswd");
                try {
                    sm.setMailServerProperties();
                    sm.createEmailMessage(userMail, "Your Password is:"+userPswd);
                    sm.sendEmail();
                } catch (AddressException e) {
                    e.printStackTrace();
                } catch (MessagingException e) {
                    e.printStackTrace();
                }
                RequestDispatcher rd = request.getRequestDispatcher("UserLogin.html");
                rd.forward(request, response);
            }
```

User Home-Page/ Track Your Records JSP:

- **If user doesn't have children and is not pregnant**

```
<%
if (noOfChildren == 0 && !isNowPreg.equals("true") && daysBetween < 0) {
            System.out.println("Inside isNowPreg: " + isNowPreg + " daysBetween: " + daysBetween + " noOfChildren: "
                    + noOfChildren);
    %>
    <center>
        <h2>
            Add your pregnancy detials or Add Children in <a href="profile.jsp">Profile
                Section</a>
        </h2>
        <br />
        <h2>To Know about Pregnancy and child care navigate</h2>
        <form action="women.html" method='get'>
            <input type="submit"
                style="margin-left: 100px; color: blanchedalmond; font-size: large; height: 100px; width: 250px; background-color: #0088cc; align-content: center"
                name='PregnancyDetails' value="Know About Pregnancy">
        </form>
        <br /> <br />
        <form action="child.html" method='get'>
            <input type="submit"
                style="margin-left: 100px; color: blanchedalmond; font-size: large; height: 100px; width: 250px; background-color: #99003d; align-content: center"
                name='ChildDetails' value="Know About Child care">
        </form>
    </center>
    <%
        }
```

- **If user is pregnant and is not an archived entry**

```
    if (noOfTimesPreg > 0 && daysBetween >= 0) {
%>
<center>
    <h3>
        <b> Pregnancy: Vaccination & Nutritions </b>
    </h3>
</center>
<%
    String pregName = (String) session.getAttribute("pregName");
        String startDateOfPreg = (String) session.getAttribute("startDateOfPreg");
%>
<center>

    <form action='PregnantDetails.jsp' method='get'>

        <input type="submit"
            style="color: blanchedalmond; font-size: 30px; height: 120px; width: 300px; background-color: #009999; align-content: center"
            name="pregName" value=" <%out.print(pregName);%> ">
    </form>
```

```
    if (session.getAttribute("noOfChildren") != null && noOfChildren > 0) {
%>
<center><br />
    <h3>
        <b>Children: Vaccination & Nutritions</b>
    </h3>
</center>
<%
    ArrayList<String> childNames = (ArrayList<String>) session.getAttribute("childNames");
        ArrayList<Integer> childIds = (ArrayList<Integer>) session.getAttribute("childIds");
        if (childNames != null && childIds != null)
            for (int i = 0; i < childNames.size(); i++) {
%>
<center>
    <br /> <br />
    <form action='ChildDetails.jsp' method='get'>
        <input type="submit"
            style="color: blanchedalmond; font-size: 30px; height: 120px; width: 300px; background-color: #cc0066; align-content: center"
            name="childName" value=" <%out.print(childNames.get(i));%> ">
    </form>
```

children

```
ArrayList<String> dates = new ArrayList<>();
ArrayList<String> vaccinations = new ArrayList<>();
PreparedStatement pstmt = con.prepareStatement("select * from child where cId=?");
pstmt.setInt(1, childId);
ResultSet rs = pstmt.executeQuery();
while (rs.next()) {
    dates.add(rs.getString("day0"));
```

Child Details JSP page to return vaccinations and their respective dates

```
PreparedStatement pstmt2 = con.prepareStatement("select * from childvaccinations;");
ResultSet rs2 = pstmt2.executeQuery();
while (rs2.next()) {
    vaccinations.add(rs2.getString("day0"));
```

```
<table border=1 align="center"  bordercolor = "black" bgcolor = "#e6ffe6" width = "900" height = "600">
    <col width="180">
    <col width="500">
    <caption align=top>
        <h3>Dates on which Vaccinations to be taken</h3>
    </caption>
    <th>Date:</th>
    <th>Vaccinations:</th>
    <%
        for (int i = 0; i < dates.size(); i++) {
    %>
    <tr>
        <td align="center">
            <%
                out.println(dates.get(i));
            %>
        </td>
        <td align="center">
            <%
            String vaccines[]=vaccinations.get(i).split(",");
            for(int j=0;j<vaccines.length;j++){
                out.println(vaccines[j]);%><br/><%
            }
            %>
        </td>
    </tr>
```

```
ArrayList<String> dates = new ArrayList<>();
ArrayList<String> tests = new ArrayList<>();
PreparedStatement pstmt = con.prepareStatement("select * from pregnantPerson where userid=? and day261Status!='true'");
pstmt.setInt(1, userId);
ResultSet rs = pstmt.executeQuery();
while (rs.next()) {
    dates.add(rs.getString("day31Date"));
```

Pregnancy Details JSP page to return vaccinations, visits and their respective dates

```
PreparedStatement pstmt2 = con.prepareStatement("select * from pregnantTests;");
ResultSet rs2 = pstmt2.executeQuery();
while (rs2.next()) {
    tests.add(rs2.getString("tests"));
}
```

```
<table border=1 align="center" bordercolor="black" bgcolor="#e6f2ff"
    width="1000" height="500">
    <col width="180">
    <col width="500">
    <caption align=top>
        <h3>Dates on which Tests to be taken</h3>
    </caption>
    <th>Date:</th>
    <th>Tests:</th>
    <%
        for (int i = 0; i < dates.size(); i++) {
    %>
    <tr>
        <td align="center">
            <%
                out.println(dates.get(i));
            %>
        </td>
        <td align="center">
            <%
                String vaccines[] = tests.get(i).split(",");
                        for (int j = 0; j < vaccines.length; j++) {
                            out.println(vaccines[j]);
            %><br />
            <%
                }
            %>
        </td>
    </tr>
    <%
        }
    %>
</table>
```

Profile Section

- **Add Child JSP**

HTML code

```
<center>
    <h3>Want to add CHILD? Fill in the details.</h3>
    <form action="addChild.jsp">
        <div id="ifYes" style="display: block">
            No.of Children: <input type="text" name="noofchildren"
                id="noofchildren" placeholder="maximum 3 children are only allowed"
                onkeyup="javascript:giveChildren();">
            <div id="childInfo"></div>
        </div>
        <div id="button"></div>
        </div>
    </form>
</center>
```

Java Code

```
if(request.getParameter("noofchildren")!=null){
int noOfChildren=Integer.parseInt(request.getParameter("noofchildren"));
int parent=(Integer)session.getAttribute("userId");
if(noOfChildren>0){
    String cNames[] = request.getParameterValues("cName");
    String cDobs[] = request.getParameterValues("cDob");
    String cGenders[] = request.getParameterValues("cGender");
    for(int i=1;i<=noOfChildren;i++){
        PreparedStatement pstmt3 = con.prepareStatement(
                "insert into child(cName,cGender,cDob,parent,day0,day42,day71,day99,day472,
                    day1780,day3560,day4300) values(?,?,?,?,?,?,?,?,?,?,?,?,?);");
        int rs3 = pstmt3.executeUpdate();
        if (rs3 > 0) {
            System.out.println("Child entry entered into db");
        %>
            <jsp:forward page="profile.jsp" ></jsp:forward>
        <%
        }
```

- **Add Pregnancy JSP**

HTML Code

```
<h2>Want to add Pregnancy Details? Fill in the details.</h2>
<form action="addPregnancy.jsp" method = "get">
        <h3>Start Date of Pregnancy:</h3> <input style="font-size: 16pt; height: 40px; width:280px;"  type="date" name="pregnancyDate" id="pre
    <br/><br/><input class ="button" type = "submit" name = "Add Pregnancy Details" value="AddPregnancyDetails">
</form>
```

Java Code

```
if (request.getParameter("pregnancyDate") != null) {
    String pregnancyDate = (String) request.getParameter("pregnancyDate");
    int[] days = { 31, 61, 91, 121, 151, 181, 196, 211, 226, 241, 248, 255, 261 };
    String[] alertDates = new String[days.length];
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Calendar c = Calendar.getInstance();
    try {
        for (int i = 0; i < days.length; i++) {
            c.setTime(sdf.parse(pregnancyDate));
            c.add(Calendar.DAY_OF_MONTH, days[i]);
            alertDates[i] = sdf.format(c.getTime());
        }
        LocalDate SystemDate = java.time.LocalDate.now();
        LocalDate lastDate = LocalDate.parse(alertDates[alertDates.length-1]);
        long lastDateToToday = ChronoUnit.DAYS.between(SystemDate, lastDate);
        String status="false";
        if(lastDateToToday<0) {
            status="true";
        }
```

```
        PreparedStatement pstmtPreg = con.prepareStatement(
                "insert into pregnantPerson(userid, startDateOfPreg,day31Date, day61Date, day91Date, day121Date, day151Date,
                 day181Date, day196Date, day211Date, day226Date, day241Date, day248Date, day255Date, day261Date, day261Status )
                  values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
                  pstmtPreg.setInt(1, userId);
            pstmtPreg.setString(2, pregnancyDate);
            for (int i = 0; i < alertDates.length; i++) {
                pstmtPreg.setString(i + 3, alertDates[i]);
                System.out.println(i + 3 + " " + alertDates[i]);
            }
            pstmtPreg.setString(16, status);
            int rs3 = pstmtPreg.executeUpdate();
            if (rs3 > 0) {
                System.out.println("Pregnant entry entered into db");
    %>
        <jsp:forward page="profile.jsp"></jsp:forward>
    <%
}
```

Daily Meal Suggestion

```
Meal[] weeklyMeals = new Meal[8];
        // 1 - Sunday
        weeklyMeals[1] = new Meal("Low-Gi muesli,Natural yoghurt,Few slices of your favourite fruit",
                "Soya and potato (aloo) curry,\n Ladies Finger,\n Curd,\n Chapati/ Rice",
                "Lemonade, Mixed nuts and raisins", "Mung bean dal, Mixed Veg Curry, Chapati/ Rice");
        weeklyMeals[2] = new Meal("Sprouts paratha , Buttermilk", "Spinach and Corn Curry, Chapati/ Rice", "Banana",
                "Peas and mint soup, Pasta with tomato sauce");
        weeklyMeals[3] = new Meal("Sautéed mushrooms, Wholewheat toast, Banana milkshake",
                "Mixed dal, Carrot and peas sabzi, Curd (dahi), Pearl millet (bajra) roti", "Roasted Peanuts",
                "Cumin and potato curry, Chapati/ Rice");
        weeklyMeals[4] = new Meal("Tomato and cucumber sandwich with mint chutney, Pineapple juice",
                "Black gram dal, Brinjal bharta, Curd, Chapati/rice", "Coconut water, Roasted chickpeas",
                "Paalak Paneer, Jowar Roti");
        weeklyMeals[5] = new Meal("Low-Gi muesli,Natural yoghurt,Few slices of your favourite fruit",
                "Soya and potato (aloo) curry,\n Ladies Finger,\n Curd,\n Chapati/ Rice",
                "Lemonade, Mixed nuts and raisins", "Mung bean dal, Mixed Veg Curry, Chapati/ Rice");
        weeklyMeals[6] = new Meal("Sprouts paratha , Buttermilk", "Spinach and Corn Curry, Chapati/ Rice", "Banana",
                "Peas and mint soup, Pasta with tomato sauce");
        weeklyMeals[7] = new Meal("Sautéed mushrooms, Wholewheat toast, Banana milkshake",
                "Mixed dal, Carrot and peas sabzi, Curd (dahi), Pearl millet (bajra) roti", "Roasted Peanuts",
                "Cumin and potato curry, Chapati/ Rice");
        Calendar calendar = Calendar.getInstance(TimeZone.getDefault());
        int dayOfWeek = calendar.get(Calendar.DAY_OF_WEEK);
        Meal todaysMeal = weeklyMeals[dayOfWeek];
```

```
    <table border=1 align="center"  bordercolor = "black" bgcolor = "#f2e6ff" width = "1000" height = "300">
        <col width="180">
        <col width="500">
        <caption align=top>
            <h3>Today's Meal</h3>
        </caption>
        <tr>
            <td align="center">BreakFast</td>
            <td align="center">
                <%
                    String[] food = todaysMeal.getBreakfast().split(",");
                    for (int i = 0; i < food.length; i++) {
                        out.println(food[i]);
                    %> <br> <%
    }
%>
            </td>
        </tr>
        <tr>
            <td align="center">Lunch</td>
            <td align="center">
                <%
                    food = todaysMeal.getLunch().split(",");
                    for (int i = 0; i < food.length; i++) {
                        out.println(food[i]);
                    %> <br> <%
    }
%>
            </td>
        </tr>

        <tr>
            <td align="center">Snack</td>
            <td align="center">
                <%
                    food = todaysMeal.getSnack_option().split(",");
                    for (int i = 0; i < food.length; i++) {
                        out.println(food[i]);
                    %> <br> <%
    }
%>
            </td>
        </tr>
        <tr>
            <td align="center">Dinner</td>
            <td align="center">
                <%
                    food = todaysMeal.getDinner().split(",");
                    for (int i = 0; i < food.length; i++) {
                        out.println(food[i]);
                    %> <br> <%
    }
%>
            </td>
        </tr>

    </table>
```

Emergency Hospital Search JSP

- **HTML Code**

```
<form action="emergencySearch.jsp">
    <h2>Enter PINCODE:</h2>
    <input style="font-size: 16pt; height: 40px; width: 280px;"
        type="text" name="pincode" placeholder="Enter valid pincode"
        onkeyup="validatePinCode(document.getElementById('userPinCode').value)"
        required> <br /> <br />
    <input style="color: blanchedalmond; font-size: large; height: 40px; width: 100px;
        background-color: rgb(80, 4, 80); align-content: center"
        type="submit" value="SEARCH">
</form>
```

- **Java Code**

```
String pinCode = request.getParameter("pincode");
if(pinCode!=null) {
    int noOfRecords=0;
    PreparedStatement pstmt = con.prepareStatement("select * from hospital where hPinCode=?");
    pstmt.setString(1, pinCode);
    ResultSet rs = pstmt.executeQuery();
    while (rs.next()) {
        noOfRecords++;
        String hName= rs.getString("hName");
        String hAddress=rs.getString("hAddress");
        String hPhone=rs.getString("hPhone");
    %>
    <h2><%out.print(hName);%></h2>
    <h3><%out.print(hAddress);%></h3>
    <h3><%out.print(hPhone);%></h3><br/><br/>
    <%
    }}catch(Exception ex) { ex.printStackTrace();}
    if(noOfRecords==0){
        %>
        <h3>Sorry.. No Hospitals available</h3>
        <%
    }
}else{
    %>
    <jsp:forward page="EmergencySearch.html"></jsp:forward>
    <%
}
%>
```

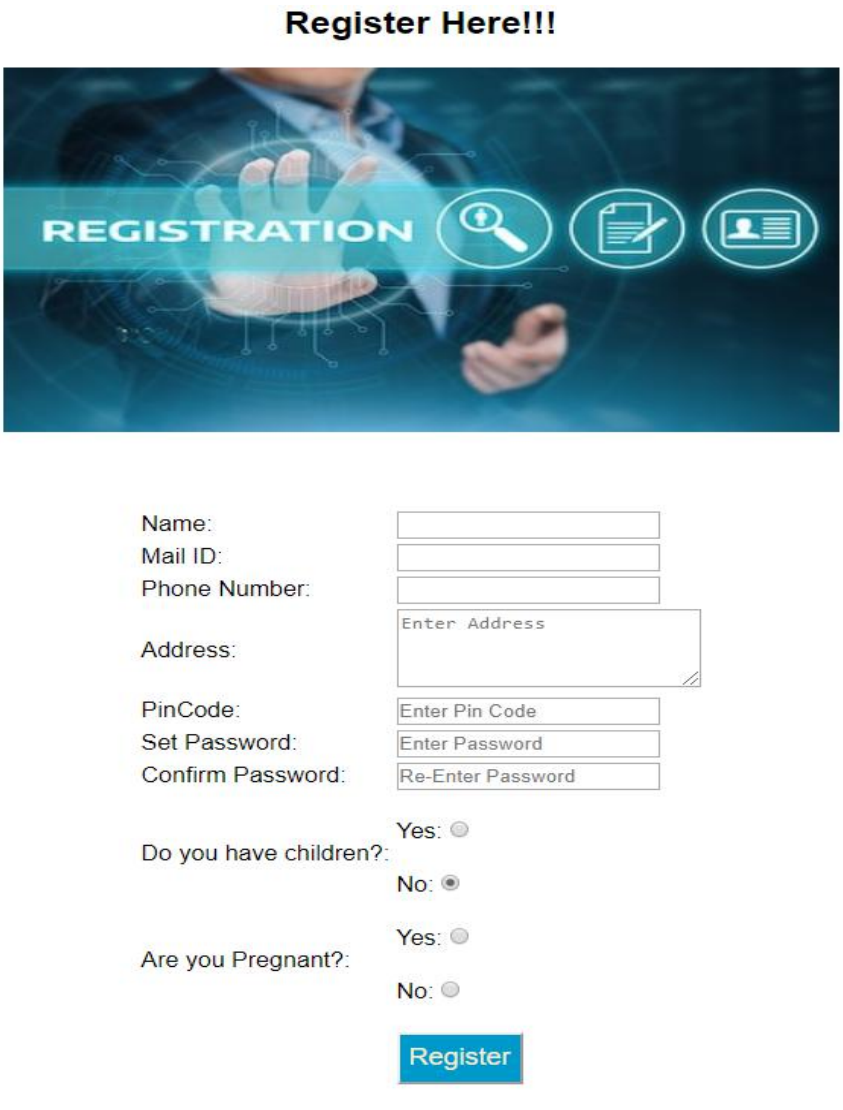
Archive Section (Pregnancy)

```
if((isNowPreg.equals("true") && noOfTimesPreg>1) || (noOfTimesPreg>0 && isNowPreg.equals("false")) ){
    %>
    <h3>Know About Your Archived Pregnancy Details here:</h3>
    PreparedStatement pstmt = con.prepareStatement("select * from pregnantPerson where userid=?");
                pstmt.setInt(1, userId);
                ResultSet rs = pstmt.executeQuery();
                while (rs.next()) {
                    startDate=rs.getString("startDateOfPreg");
                    endDate=rs.getString("day261Date");
                    LocalDate SystemDate = java.time.LocalDate.now();
                    LocalDate date = LocalDate.parse(endDate);
                    long daysBetween = ChronoUnit.DAYS.between(SystemDate, date);
                    if(daysBetween<0){
                    %>
        <center>
            <h3>
                <%out.print(startDate);%>
                to
                <%out.print(endDate); %>
            </h3>
        </center>
        <center>
            <form action='archivePregnantDetails.jsp' method='get'>
             <input type="hidden" name="lastDate" value="<%out.print(endDate); %>">
                <input type="submit" style="color: blanchedalmond; font-size: large; height: 100px;
                 width: 250px; background-color: #3973ac; align-content: center"
                    name="pregName" value=" <%out.print(pregName);%> ">
            </form>
        </center>
        <%
        }
        }
        }else{
            %>
            <center><h2>No Archives Available</h2></center>
            <%
        }
```

5.2 SCREENSHOTS

5.1.1 Home Page

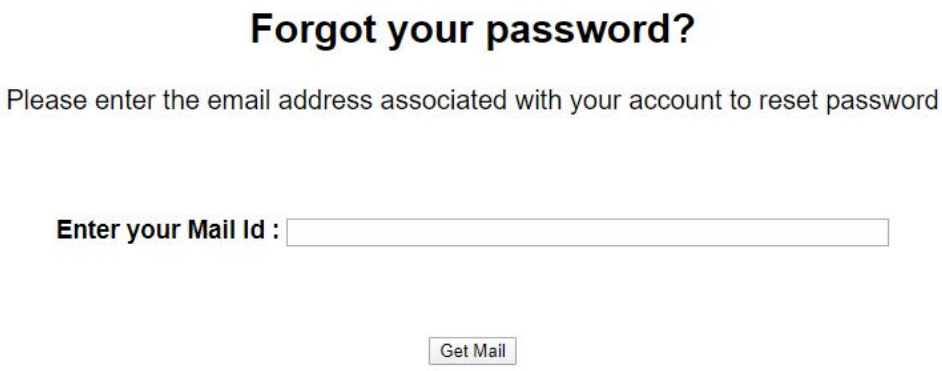
5.2.2 User Registration Page



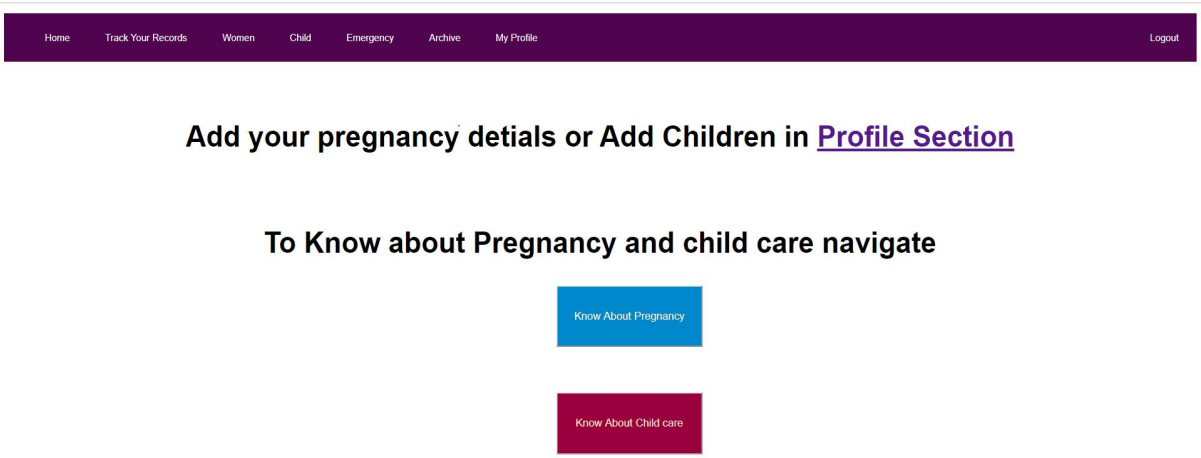
5.2.3 OTP

5.2.4 User Login Page

5.2.5 Forgot Password Page

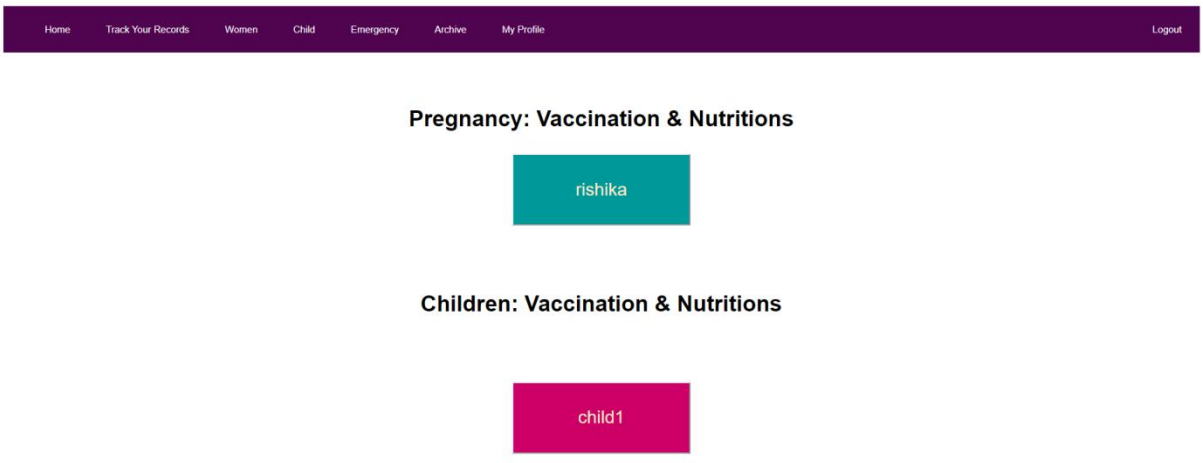


5.2.6 Track your Records Page of User without Children and Pregnancy

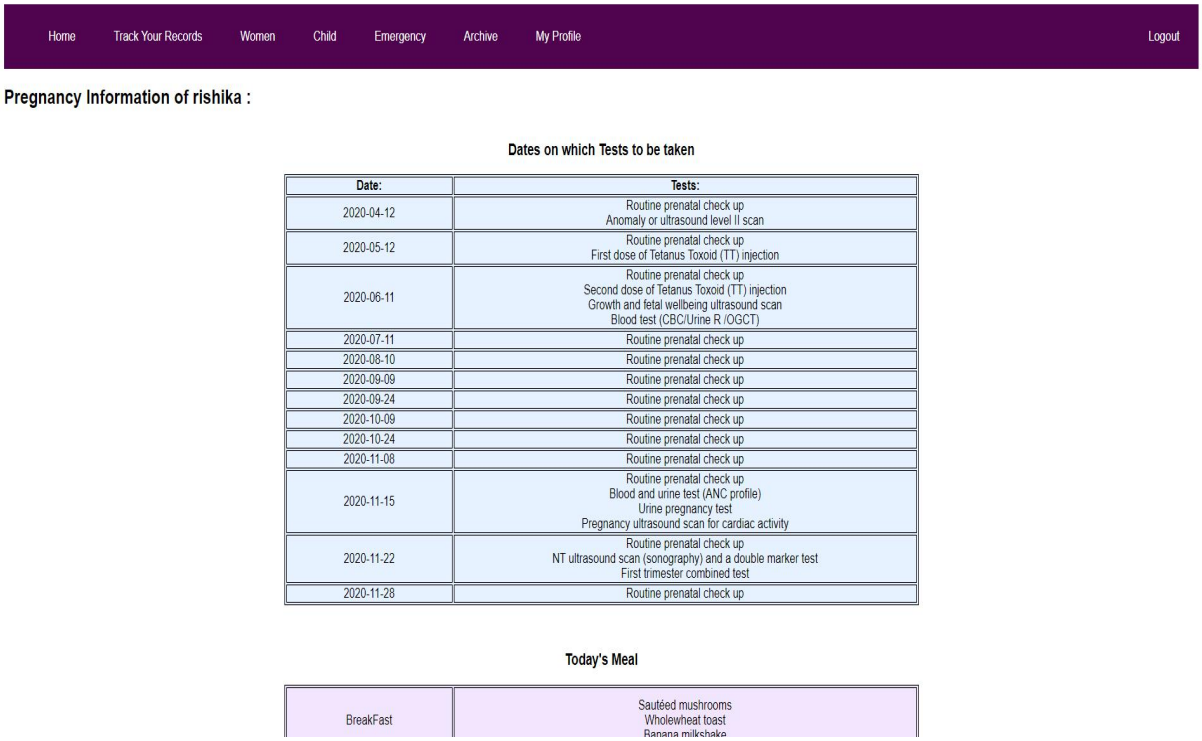


5.2.7 Track your Records page with only Pregnancy

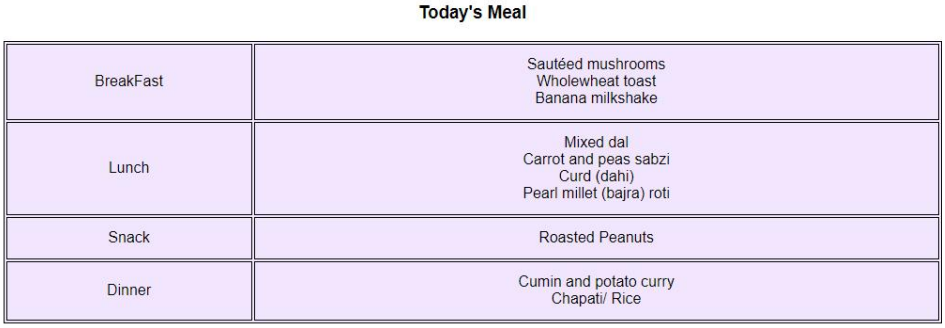
5.2.8 Track your Records page with pregnancy and child



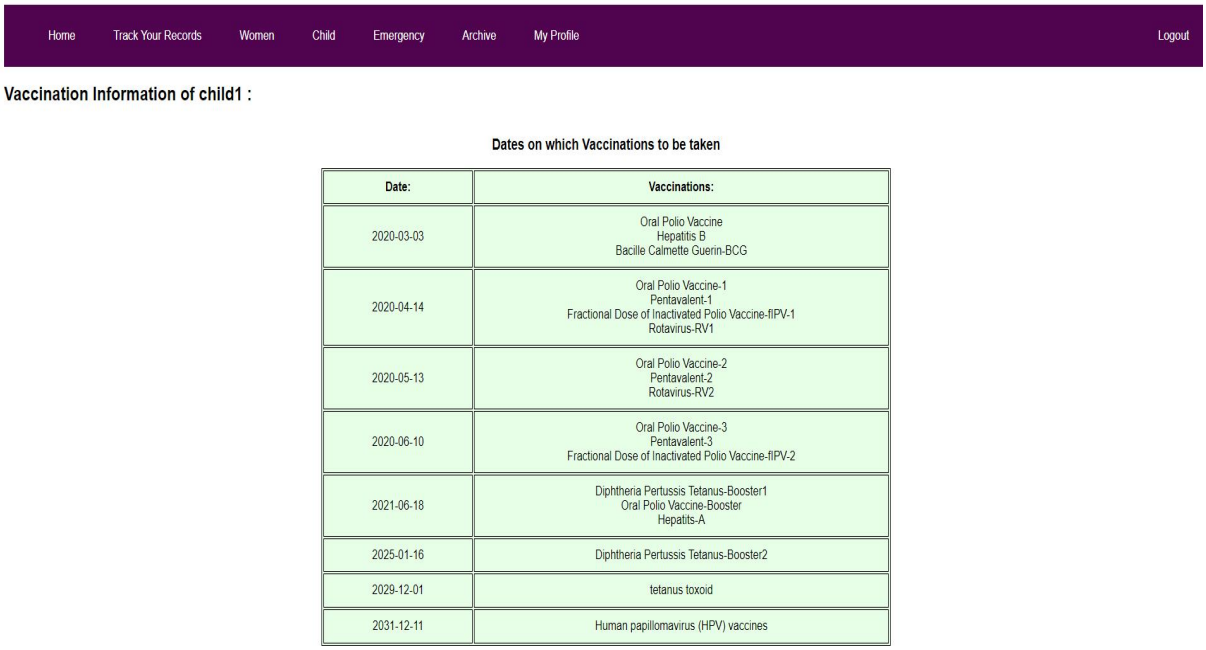
5.2.9 Pregnancy Records



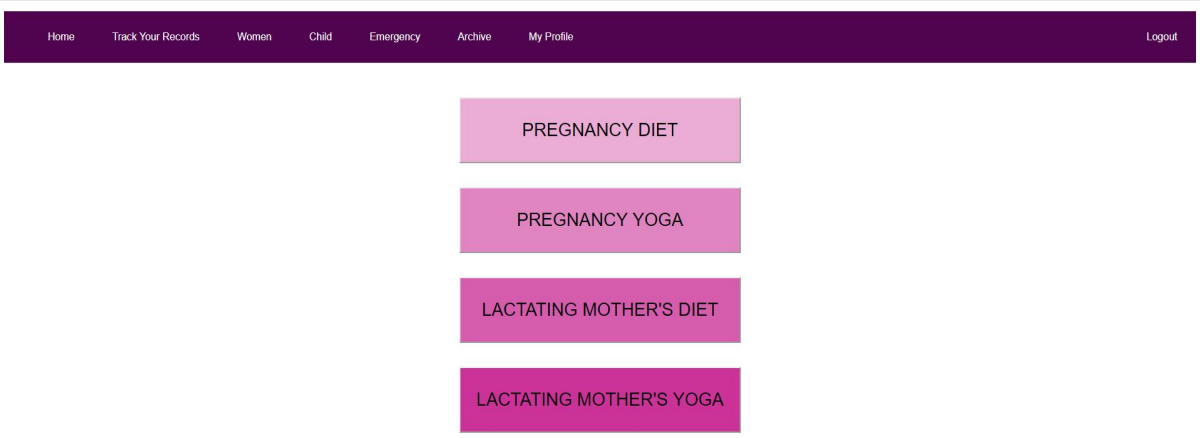
5.2.10 Daily Meal Plan



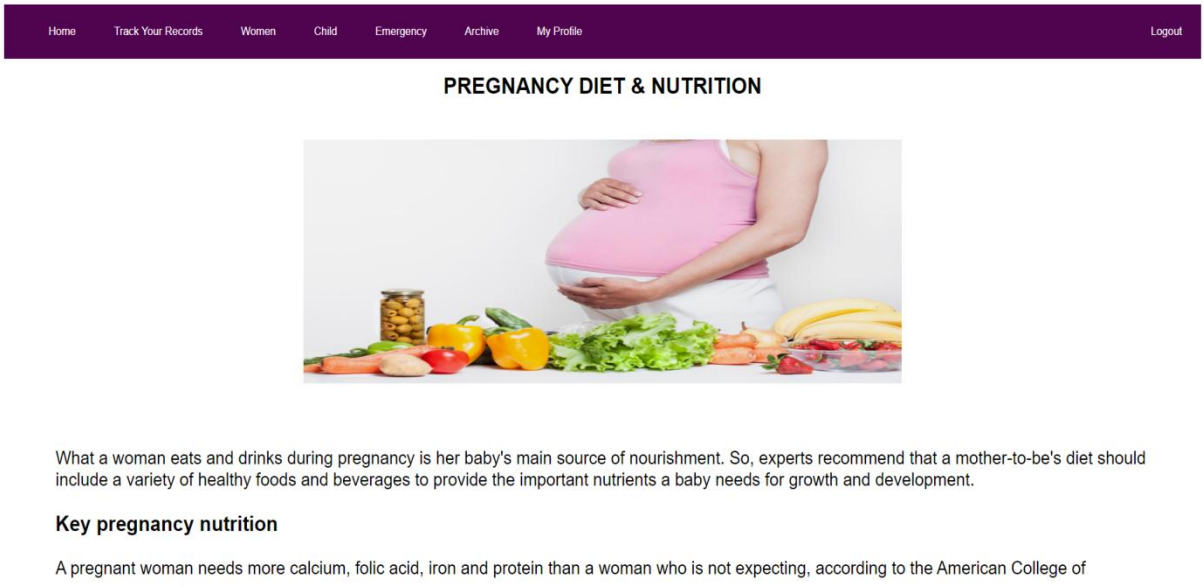
5.2.11 Child Details



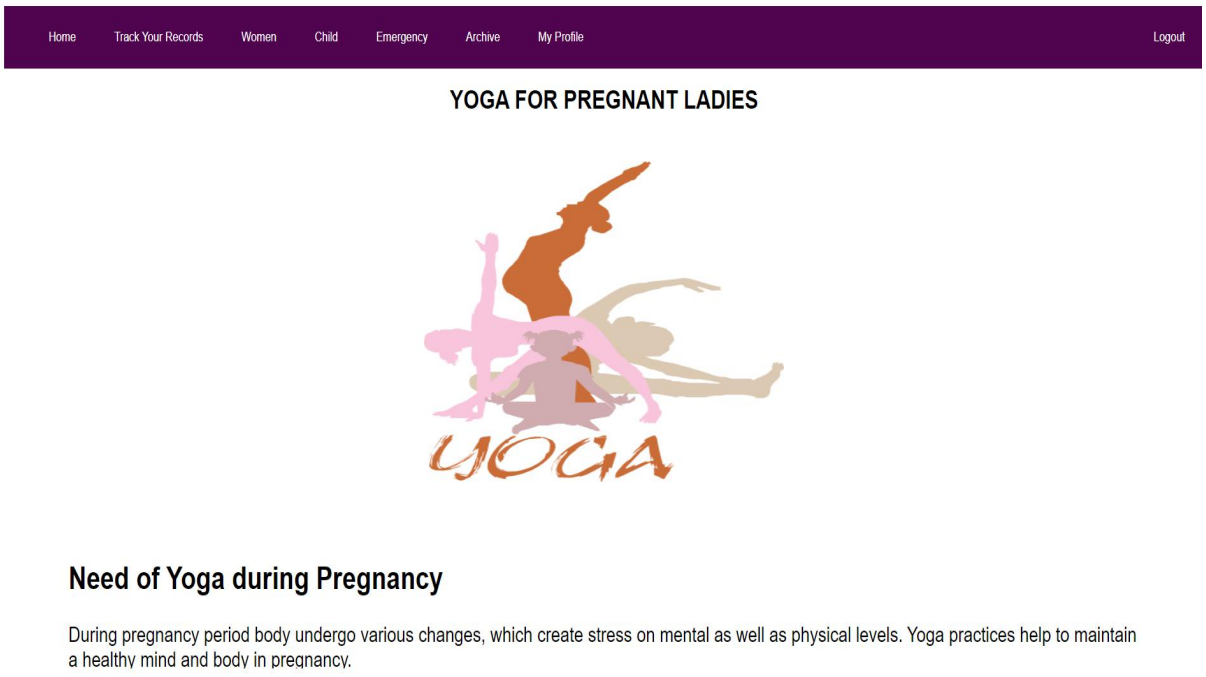
5.2.12 Women



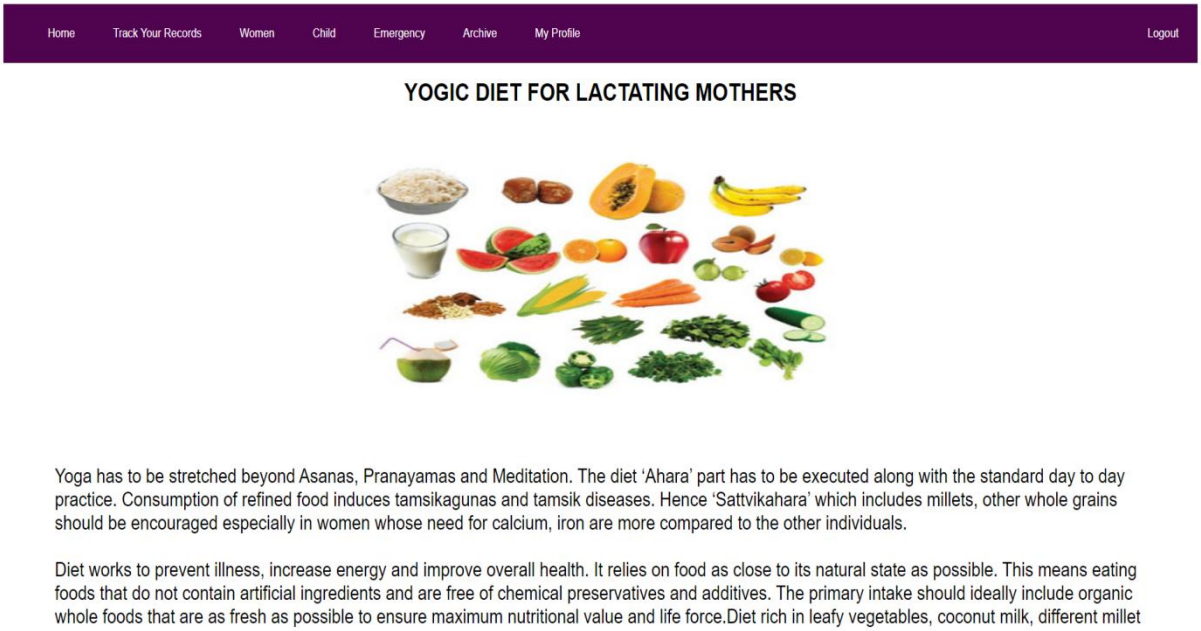
5.2.13 Pregnant Women Diet



5.2.14 Pregnant Women Yoga



5.2.15 Lactating Mothers Diet



Yoga has to be stretched beyond Asanas, Pranayamas and Meditation. The diet 'Ahara' part has to be executed along with the standard day to day practice. Consumption of refined food induces tamsikagunas and tamsik diseases. Hence 'Sattvikahara' which includes millets, other whole grains should be encouraged especially in women whose need for calcium, iron are more compared to the other individuals.

Diet works to prevent illness, increase energy and improve overall health. It relies on food as close to its natural state as possible. This means eating foods that do not contain artificial ingredients and are free of chemical preservatives and additives. The primary intake should ideally include organic whole foods that are as fresh as possible to ensure maximum nutritional value and life force.Diet rich in leafy vegetables, coconut milk, different millet

5.2.16 Lactating Mothers Yoga

Target Group:

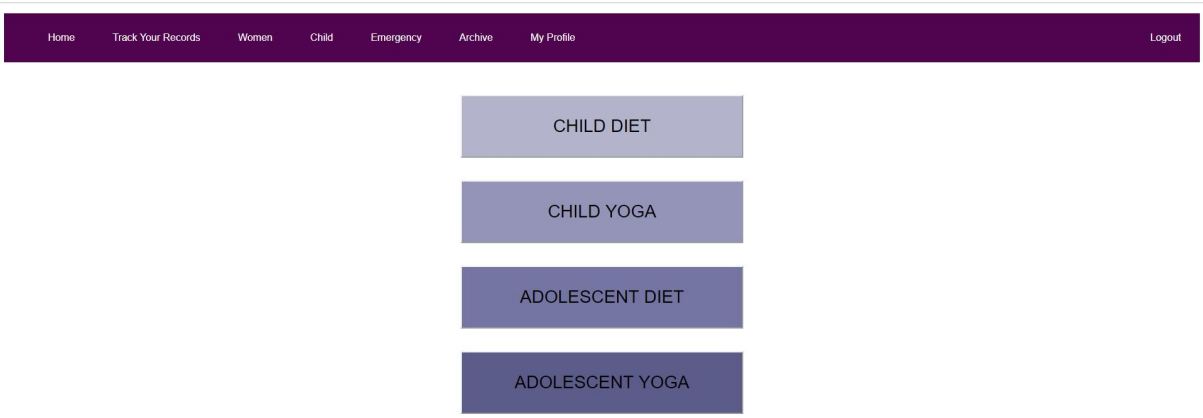
There are several ways to increase milk production, one of which is by doing Yoga exercise and having yogic diet. For the normal delivery Yoga practices are recommended after one month but for a caesarian cases Yoga practices are recommended after three months. Breathing practices shall be started immediately after delivery. Keeping in the mind all the concerns mothers face after child birth, yogic practices, diet, cautions and precautions etc.

General Guidelines Before the Practice

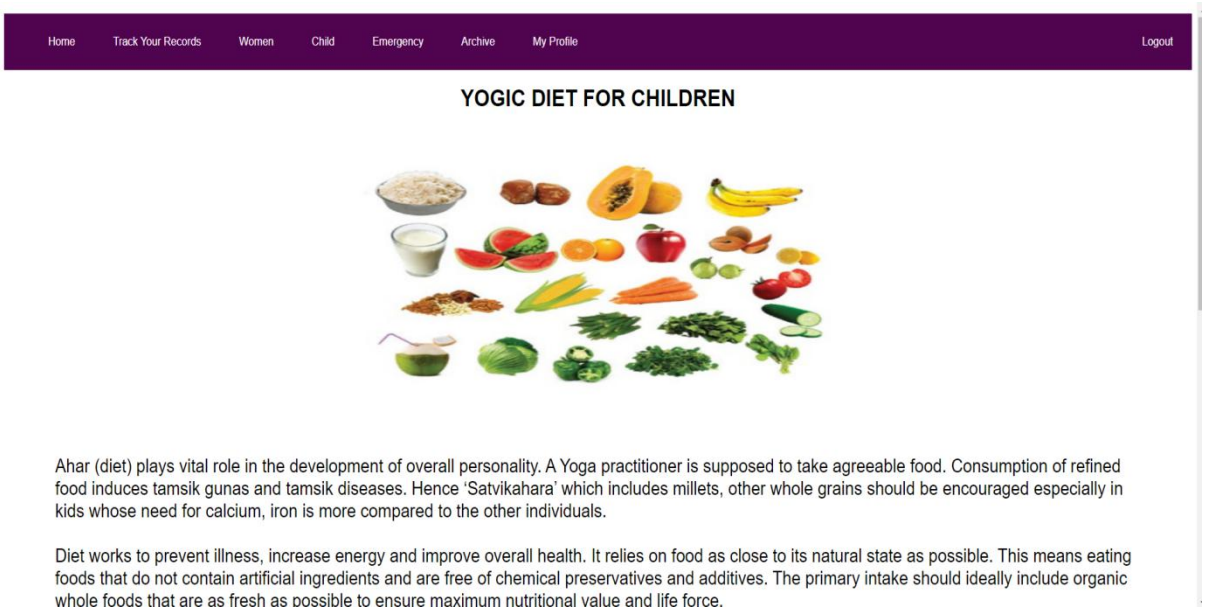
Śauca means cleanliness - an important prerequisite for Yogic practice. It includes cleanliness of surroundings, body and mind

Yogic practice should be performed in a calm and quiet atmosphere with a relaxed body and mind

5.2.17 Child



5.2.18 Child Diet



5.2.19 Child Yoga

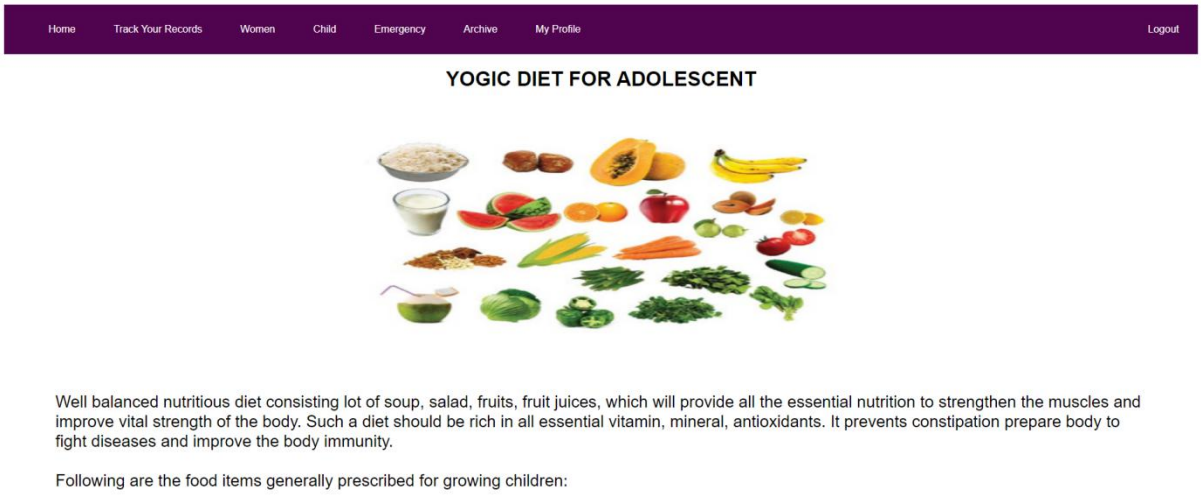
II . HAND MOVEMENTS

Hands up and down

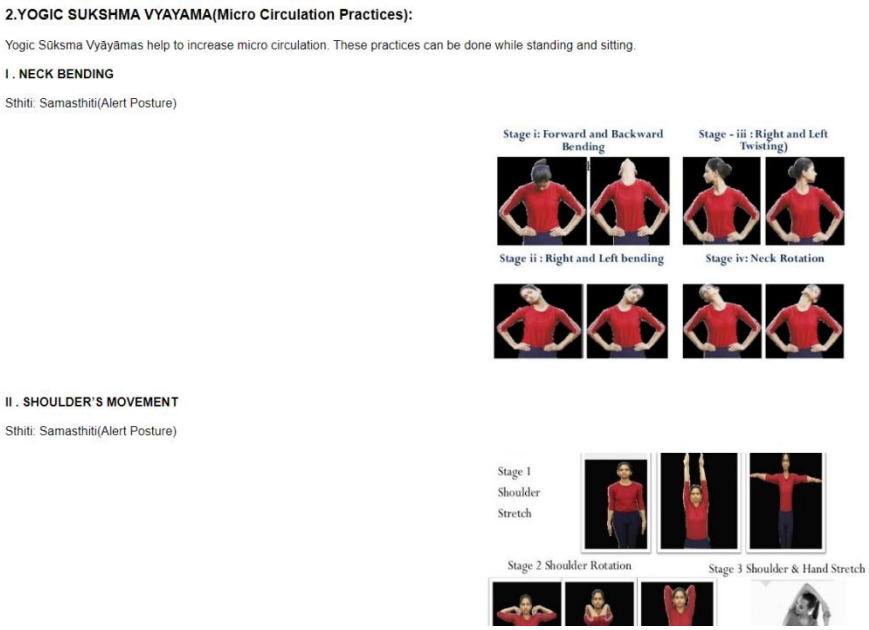
Hands rotation

III . SHOULDER MOVEMENTS

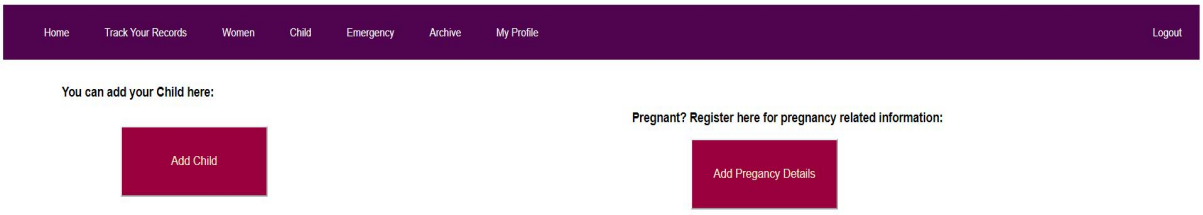
5.2.20 Adolescents Diet



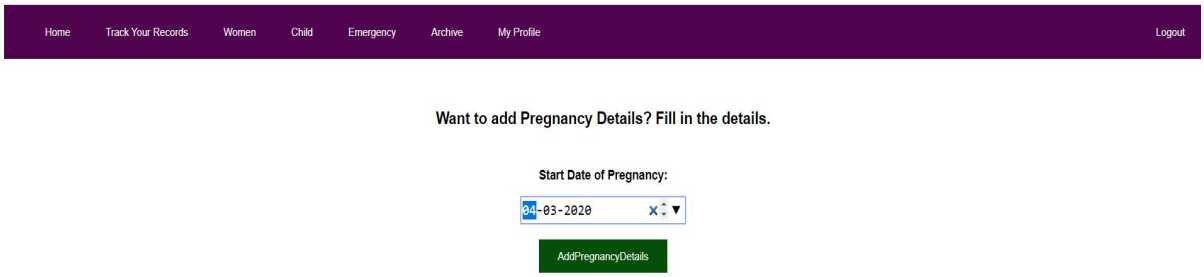
5.2.21 Adolescents Yoga



5.2.22 Profile without Children and Pregnancy

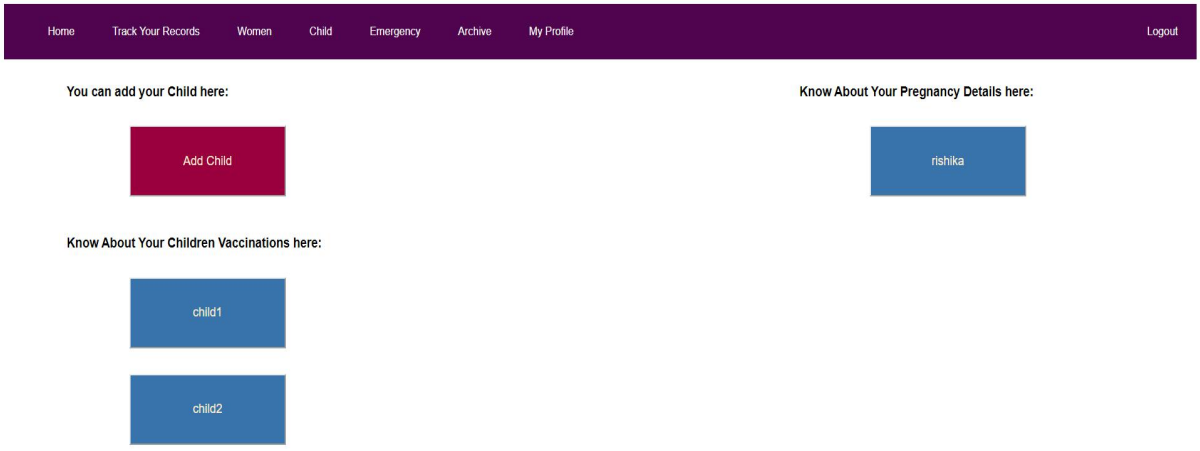


5.2.23 Add Pregnancy Details

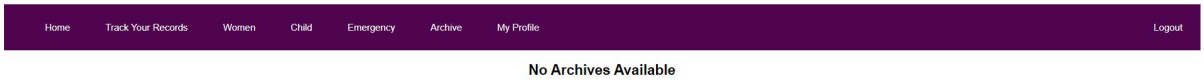


5.2.24 Add Child Details

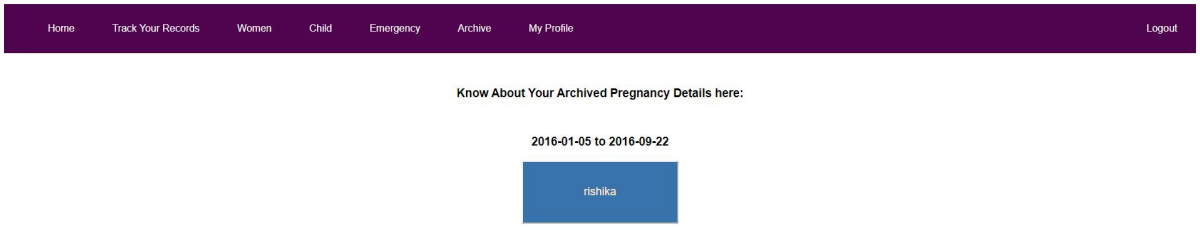
5.2.25 Profile with Children and Pregnancy



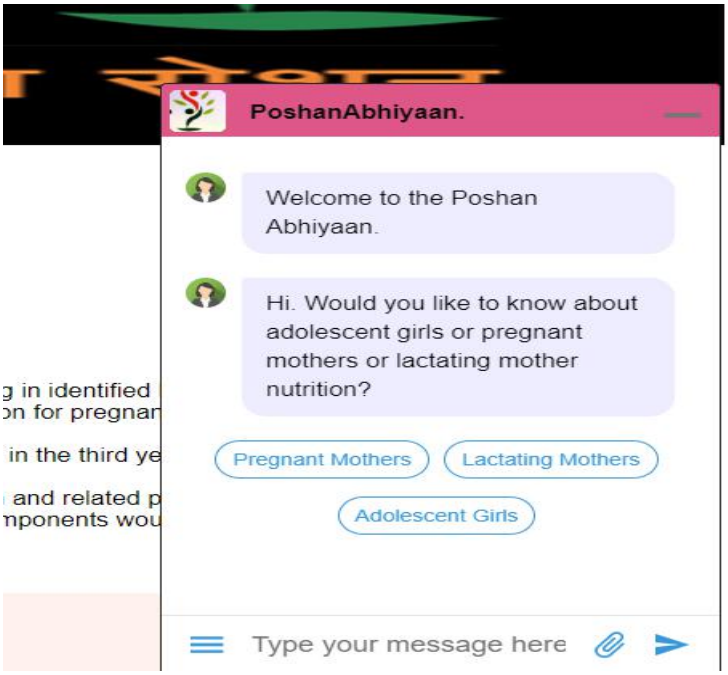
5.2.26 Archive with No Entries

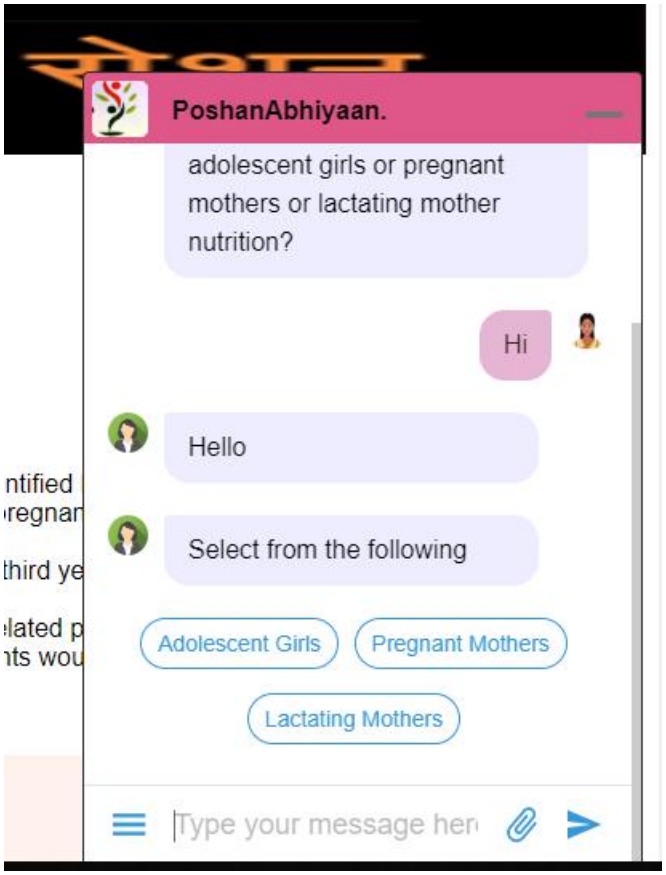


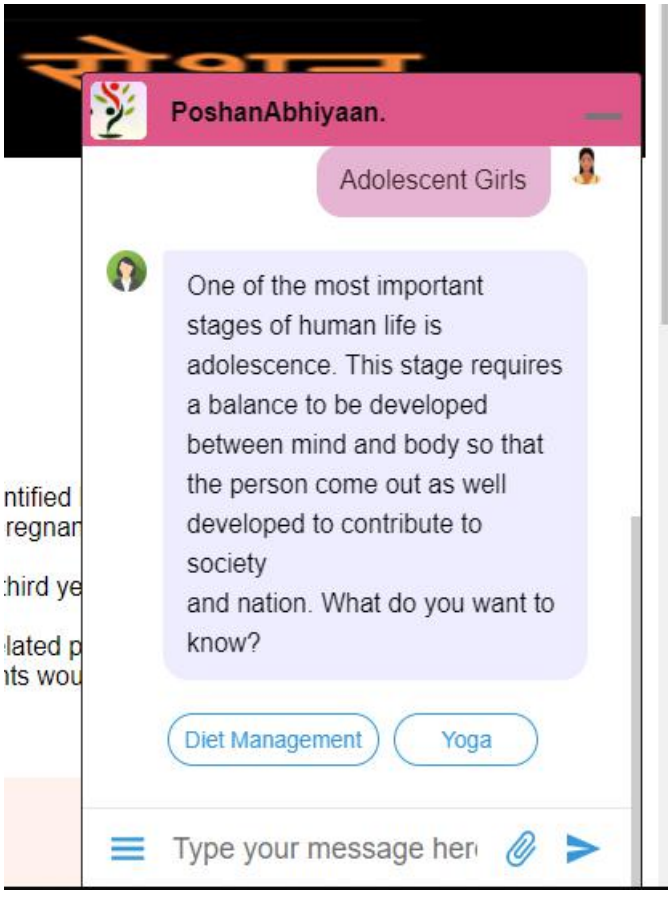
5.2.27 Archive



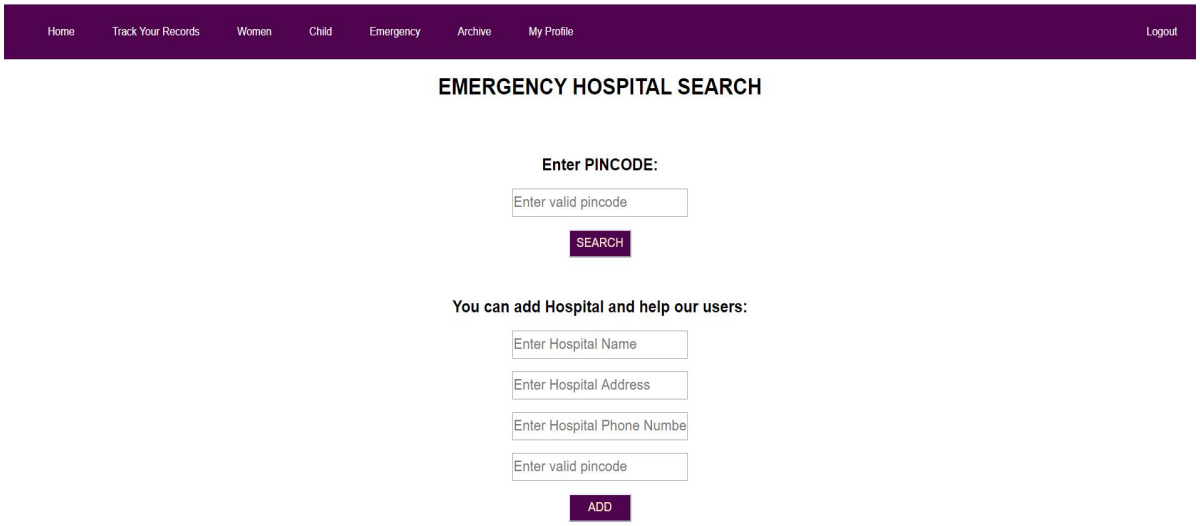
5.2.28 Chatbot




PoshanAbhiyaan.
adolescent girls or pregnant mothers or lactating mother nutrition?
Hi
Hello
Select from the following
Adolescent Girls
Pregnant Mothers
Lactating Mothers
Type your message here


PoshanAbhiyaan.
Adolescent Girls
One of the most important stages of human life is adolescence. This stage requires a balance to be developed between mind and body so that the person come out as well developed to contribute to society
and nation. What do you want to know?
Diet Management
Yoga
Type your message here

5.2.29 Emergency Hospital Search



6. SYSTEM TESTING

6.1 Introduction to testing

The purpose of testing is to detect error. Testing is the system of seeking to observe each possible fault or weak point in a piece product. It presents a technique to determine the functionality of accessories, sub-assemblies, assemblies and/or a finished product.

It is the approach of exercising program with the intent of ensuring that the application system meets its standards and consumer expectations and does now not fail in an unacceptable method. There are quite a lot of types of test. Each experiment style addresses a precise trying out requirement.

Testing is one of the principal phases within the application development pastime. In program development lifestyles cycle (SDLC), the predominant aim of Testing process is the satisfactory; the progress application is demonstrated against achieving the specified performance and efficiency.

For the duration of the trying out procedure the program is worked with some detailed scan cases and the output of the test circumstances are analyzed whether the application is working in step with the expectations or not. The success of the Testing system in settling on the errors in general relies on the experiment case standards, for trying out any program we have to have an outline of the anticipated habits of the method and process of settling on whether or not the found habits confirmed to the expected conduct.

LEVELS OF TESTING

When you consider that the mistakes within the application can be injured at any stage. So, we have got to carry out the testing process at specific stages for the duration of the development. The elemental stages of trying out are unit, Integration, method and acceptance trying out.

The unit testing is carried out on coding. Here exclusive modules are proven against the specification produced during design for modules.

In case of Integration Testing exceptional verified modules are combined into sub systems and established in case of the procedure Testing the entire application is demonstrated and within the subsequent level of trying out the method is established with person requirement file prepared for the duration of SRS.

There are two basic approaches for testing. They are,

Functional Testing

In functional testing circumstances are decided solely and the groundwork of necessities of the application or module and the Internals of the software or modules should not regarded for determination of experiment cases. That is often known as black box testing.

Structural Testing

In structural Testing scan circumstances are generated on actual code of the application or module to be demonstrated. This is called white field Testing.

6.2 TESTING ACTIVITIES

Different levels of testing are used in the testing process,each level of testing aims to test different aspects of the system.The basic levels are:

I.Unit Testing:

Unit trying out entails the design of scan circumstances that validate that the internal application common sense is functioning correctly, and that application inputs produce legitimate outputs. All decision branches and glide of inner code should be validated. It is the Testing of individual software items of the applying its carried out after the completion of an character unit earlier than integration.

It is a kind of structural Testing which depends on advantage of its construction and is invasive. Unit assessments perform normal checks at element stage and experiment a certain industry process, utility, and/or procedure configuration. Unit assessments ensure that every certain direction of a industry system performs properly to the documented requisites and contains certainly outlined inputs and expected outcome.

II. Integration Testing:

Integration exams are designed to experiment integrated application components to assess if they in reality run as one software. Testing is occasion driven and is more worried with the elemental outcome of screens or fields. Integration exams show that even though the accessories have been individually delight, as proven by means of efficaciously unit testing, the blend of accessories is right and consistent.

Integration trying out is chiefly aimed at exposing the problems that arise from the combination of add-ons.

III. System Testing:

In system testing the entire software is tested.The reference document for this process is the requirement document and the goal is to see whether the software meets its requirements.The system was tested for various test cases with various inputs.

IV .Acceptance Testing:

User Acceptance Testing out is a primary section of any undertaking and requires big participation with the aid of the end consumer. It also ensures that the approach meets the functional specifications.

6.3 TYPES OF TESTING

Black Box Testing

A software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

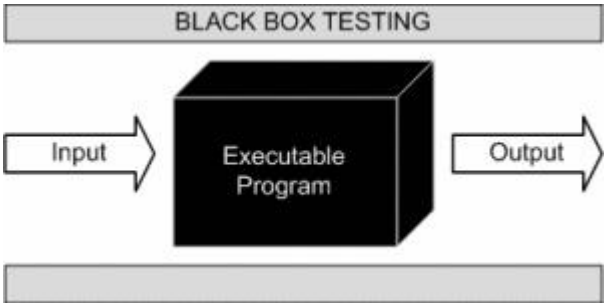


Figure 10 : Black Box Testing

White Box Testing

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees.

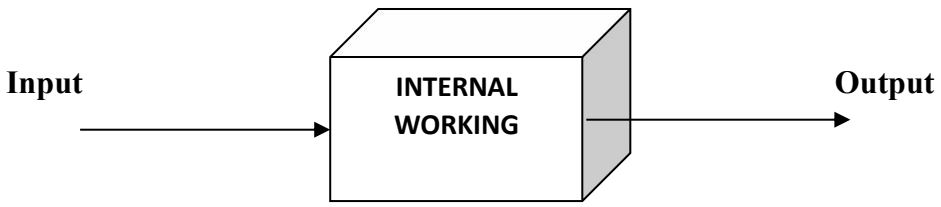


Figure 11 : White Box Testing

6.4 TEST PLAN

Experiment plan is common file for entire task, which defines the scope, method to be taken and the individual accountable for extraordinary routine of testing. The inputs for forming scan aircraft are

Challenge plan

Requirement file

Process design

Test case specification: although there is one scan plan for whole mission scan cases have to be exact separately for every test case. Scan case specification gives for each item to be confirmed all scan instances and outputs expected for those scan cases.

Testing Process

A quantity of routine ought to be carried out for Testing software. Trying out begins with experiment plan. Experiment plan identifies all testing associated hobbies that needed to be performed together with the agenda and consultant traces for testing.

The plan also specifies the phases of testing that have got to be completed, with the aid of deciding upon the exceptional trying out units. For each unit specified within the plan first the scan instances and studies are produced.

Test strategy and approach

Field testing can be performed manually and sensible assessments will likely be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

6.5 TEST CASES

A **TEST CASE** is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Types of test cases

There are two types of test cases as mentioned below:

1. Formal test cases: Formal test cases are those test cases which are authored as per the test case format. It has all the information like preconditions, input data, output data, post conditions, etc. It has a defined set of inputs which will provide the expected output.
2. Informal test cases: Informal test cases are authored for such requirements where the exact input and output are not known. In order to test them the formal test cases are not authored but the activities done and the outcomes are reported once the tests are run.

Test Cases Examples:

- **Module: User Login**
- **File Name: Login.java**

Test Case	Input	Expected Output	Actual Output	Description
Valid login	username, password, type	Success	Success	Login successful.
Invalid Login	username, password, type	Failed	Failed	Login unsuccessful. Try again.

- **Module: User registration**
- **Filename:Register.java**

Test Case	Input	Expected Output	Actual Output	Description
Register new User	User Info.	Register successful.	Register successful.	User registered.
Register new User	User Info.	Failed.	Failed.	Invalid Data.

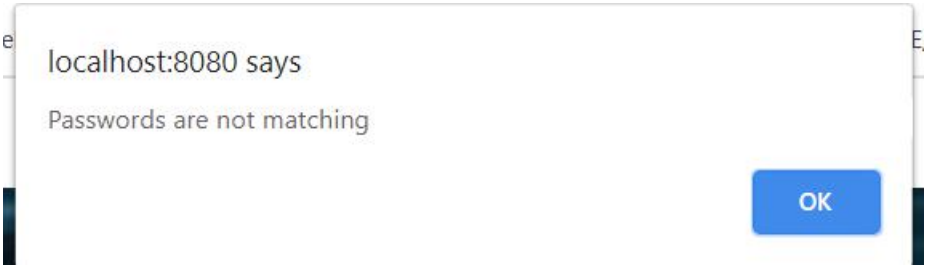
VALIDATING THE FIELDS WHILE REGISTERING

- Validating mail ID

- Validating if all fields are filled – if no, this type of warning occurs

- Validating pincode

- Validating if the Password and Re-Enter password are matching



CONCLUSION

There are multiple websites which provide information on pregnancy nutrition, pregnancy vaccines and child nutrition. But there is no website that provides information on all of them. More often than not, the users are thrown into a confusion as to which information can be trusted, due to multiple sources of information.

The website Poshan Abhiyaan provides one stop for all needs where the user can get diet plan suggestions, nutrition information, yogic diet and yoga, vaccination alerts for both children and pregnant woman. The registered user can enter the details of pregnancy or their child and can get suggestions accordingly. The vaccination alerts are sent 1 week, 3 days and 1 day prior to the day on which vaccination is to be taken, via an email. The website also provides a list of hospitals based on the user's location in case of an emergency. Also, a daily meal plan suggestion is given to the pregnant woman based on the trimester in which she is.

The website provides the user an ease of access to information and also helps them ensure that they never miss a vaccination or test. Thus, Poshan Abhiyaan helps in contributing to the health of women and children.

FUTURE ENHANCEMENTS

The web application is currently capable of generating the dates on which vaccinations and tests are to be taken, given the date of pregnancy and date of birth respectively. It sends reminders to the users to ensure that vaccination is taken at the right time. Also, it is capable of providing the hospital based on the pin code provided by the user. It provides the diet plan based on trimesters for pregnant.

In the future, we would like to make the web application capable of taking the location of the user with the help of a GPS to provide the list of hospitals nearby. We also intend to make communication stronger by enabling an SMS system where the reminders for vaccination will be messaged to the user.

We also intend to develop an android application with the above mentioned improvements for easier usage and convenience of the user.

BIBLIOGRAPHY

1. https://www.javatpoint.com/servlet-tutorial
2. https://www.geeksforgeeks.org/introduction-java-servlets/
3. https://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html
4. https://www.w3schools.com/html/html_css.asp
5. https://www.w3.org/Style/Examples/011/firstcss.en.html
6. https://www.tutorialspoint.com/jdbc/jdbc-db-connections.html
7. https://www.codejava.net/java-ee/jsp/sending-e-mail-with-jsp-servlet-and java mail
8. https://stackoverflow.com/questions/29818042/how-to-schedule-a-java-program-to-run-daily-in-windows
9. https://www.mysql.com/
10. https://www.tutorialspoint.com/mysql/index.html
11. https://stackoverflow.com/
12. https://www.india.gov.in/spotlight/poshan-abhiyaan-pms-overarching-scheme-holistic-nourishment
13. https://www.parent24.com/Pregnant/Pregnancy_health/7-DAY-meal-plan-20150826
14. https://www.drvarshaliclinic.com/learn/pregnancy/pregnancy-checkup-and-baby-ultrasound-scan-schedule
15. https://www.tutorialspoint.com/servlets/index.html
16. https://www.gupshup.io/developer/docs/bot-platform/guide/add-a-chatbot-to-your-website
17. https://chatterbot.readthedocs.io/en/stable/
18. https://www.tutorialspoint.com/maven/index.htm
19. https://mvnrepository.com/artifact/javax.mail/javax.mail-api/1.6.0
20. https://stackoverflow.com/questions/2962612/how-do-i-link-2-jsp-pages-together