A Project Report

on

“Contant Management System”

Submitted By:

Miss.Narode Vaishnavi Narode (UCS22F1095)

Roll No :106

Under the guidance of

Prof.P.M.Dhanrao

Course

Creational Activity Laboratory

Department of Computer Engineering

Sanjivani Rural Education Society’s

SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON-423603

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Academic Year: 2024-2025

Sanjivani Rural Education Society's

SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON-423603

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Department of Computer Engineering

CERTIFICATE

This is to certify that,

Miss.Narode Vaishnavi Narode

(UCS22F1095)

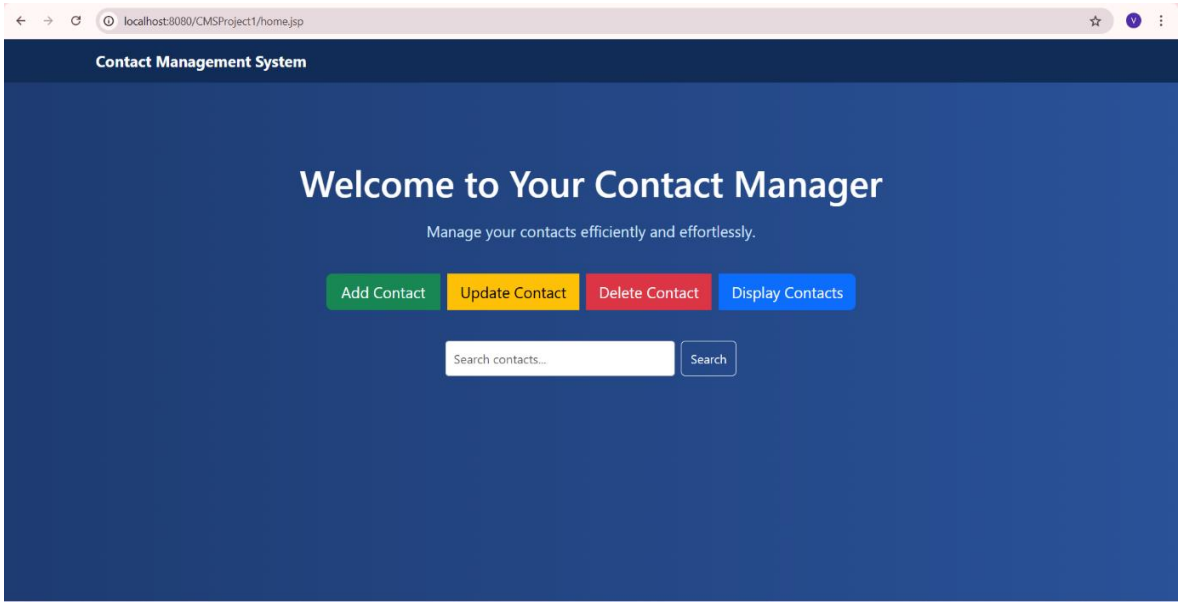
Have successfully completed the Project work entitled "Contact Management System" for the fulfillment of requirements of the course CIA, for T.Y. B.Tech Degree in Computer Engineering, Awarded by Sanjivani College of Engineering, Kopargaon, An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune.

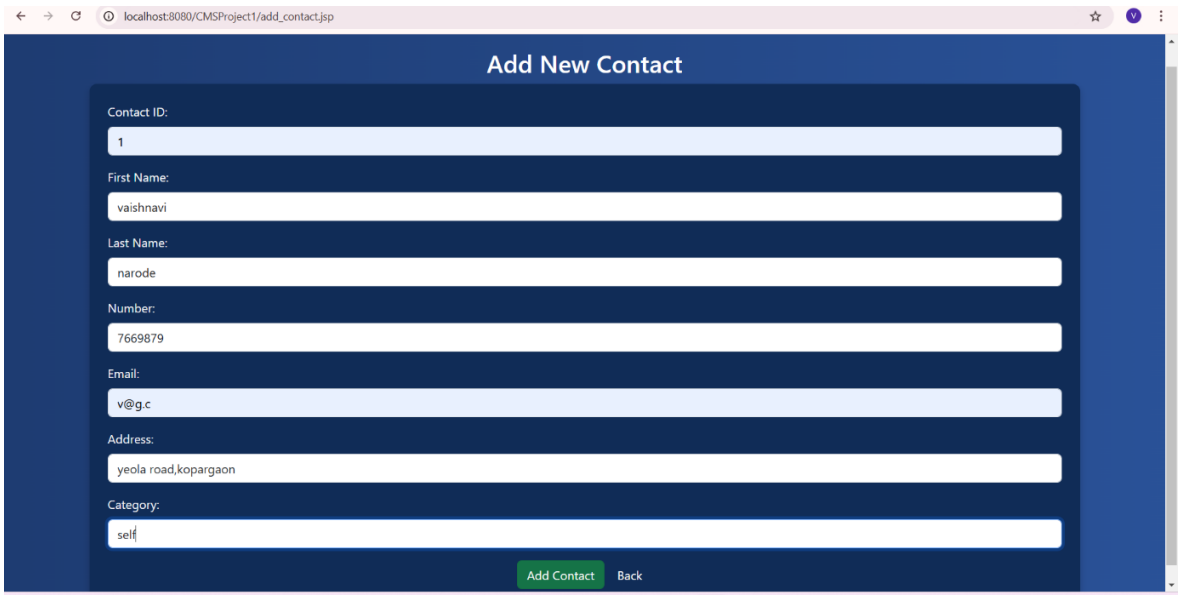
Prof.P.M.Dhanrao
(Course Teacher)

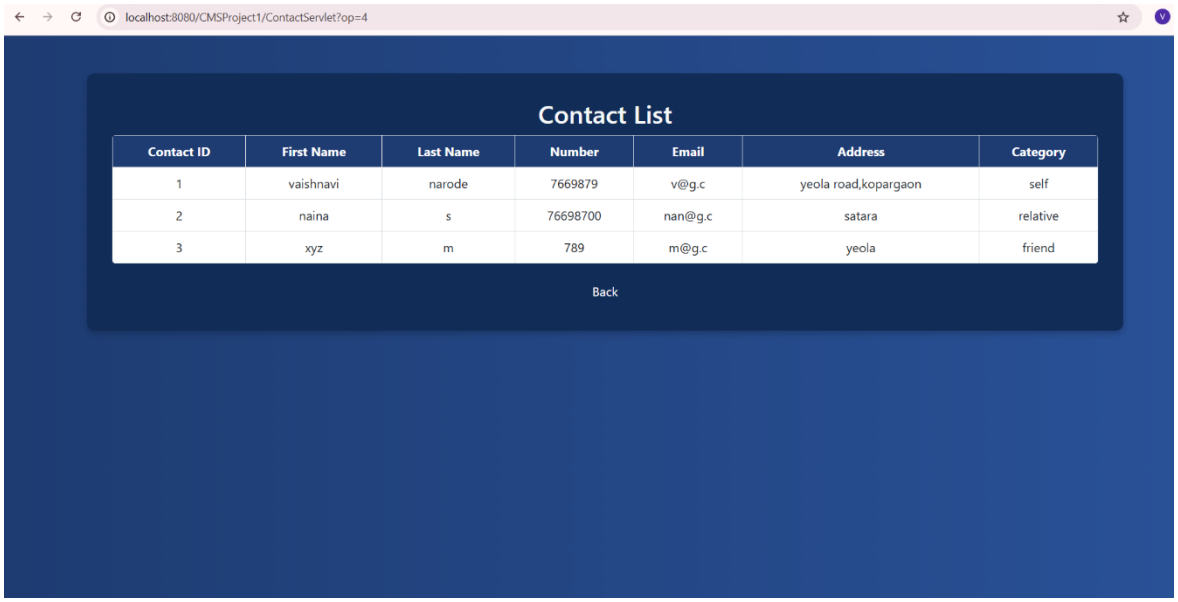
Dr.M.A.Jawale
(HOD)

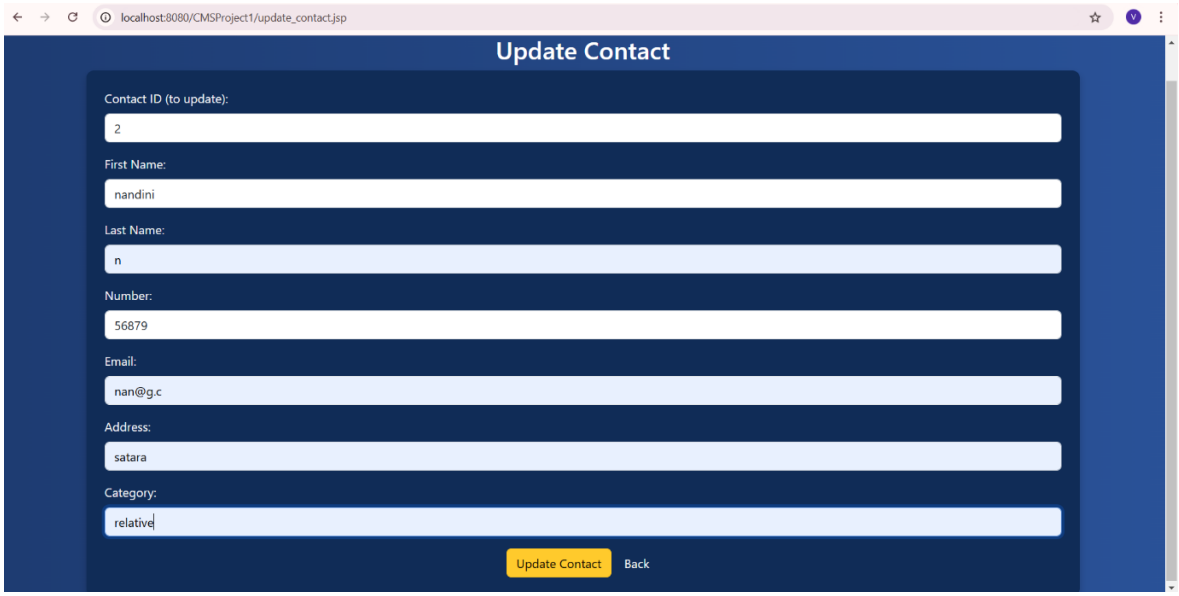
PROJECT TOTLE:CONTACT MANAGEMENT SYSTEM

FRONTEND OF PROJECT






localhost:8080/CMSProject1/ContactServlet?op=4
Contact List
Contact ID
First Name
Last Name
Number
Email
Address
Category
1
vaishnavi
narode
7669879
v@g.c
yeola road,kopargaon
self
2
naina
s
76698700
nan@g.c
satara
relative
3
xyz
m
789
m@g.c
yeola
friend
Back


localhost:8080/CMSProject1/update_contact.jsp
Update Contact
Contact ID (to update):
2
First Name:
nandini
Last Name:
n
Number:
56879
Email:
nan@g.c
Address:
satara
Category:
relative
Update Contact
Back

CODES

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                        http://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>
   <groupId>com.example</groupId>
   <artifactId>contact-management-system</artifactId>
   <version>1.0.0</version>
   <packaging>war</packaging>

   <properties>
      <java.version>17</java.version>
   </properties>

   <dependencies>
      <!-- Spring Web -->
      <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-web</artifactId>
      </dependency>

      <!-- Spring Data JPA -->
      <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-data-jpa</artifactId>
      </dependency>

      <!-- JSP & JSTL -->
      <dependency>
         <groupId>jakarta.servlet.jsp.jstl</groupId>
         <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
      </dependency>
```

```
    <dependency>
      <groupId>org.apache.tomcat.embed</groupId>
      <artifactId>tomcat-embed-jasper</artifactId>
    </dependency>

    <!-- MySQL Driver -->
    <dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
    </dependency>

    <!-- Servlet -->
    <dependency>
      <groupId>jakarta.servlet</groupId>
      <artifactId>jakarta.servlet-api</artifactId>
      <scope>provided</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.3.2</version>
        <configuration>
          <failOnMissingWebXml>false</failOnMissingWebXml>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

 application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/contactdb
spring.datasource.username=root
spring.datasource.password=yourpassword

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
```

Entity: Contact.java

```
java
CopyEdit
package com.example.contactmanagement.model;

import jakarta.persistence.*;

@Entity
@Table(name = "contacts")
public class Contact {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String email;
    private String phone;

}
```

Service: ContactService.java

```
java
CopyEdit
package com.example.contactmanagement.service;
```

```
import com.example.contactmanagement.dao.ContactDAO;
import com.example.contactmanagement.model.Contact;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class ContactService {

    @Autowired
    private ContactDAO contactDAO;

    public List<Contact> getAllContacts() {
        return contactDAO.findAll();
    }

    public void addContact(Contact contact) {
        contactDAO.save(contact);
    }
}
```

Controller: ContactController.java

```
java
CopyEdit
package com.example.contactmanagement.controller;

import com.example.contactmanagement.model.Contact;
import com.example.contactmanagement.service.ContactService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

@Controller
public class ContactController {

    @Autowired
    private ContactService contactService;
```

```
    @GetMapping("/")
    public String index(Model model) {
        model.addAttribute("contacts", contactService.getAllContacts());
        return "index";
    }

    @GetMapping("/add")
    public String addForm(Model model) {
        model.addAttribute("contact", new Contact());
        return "add";
    }

    @PostMapping("/save")
    public String save(@ModelAttribute Contact contact) {
        contactService.addContact(contact);
        return "redirect:/";
    }
}
```

Servlet: LogServlet.java

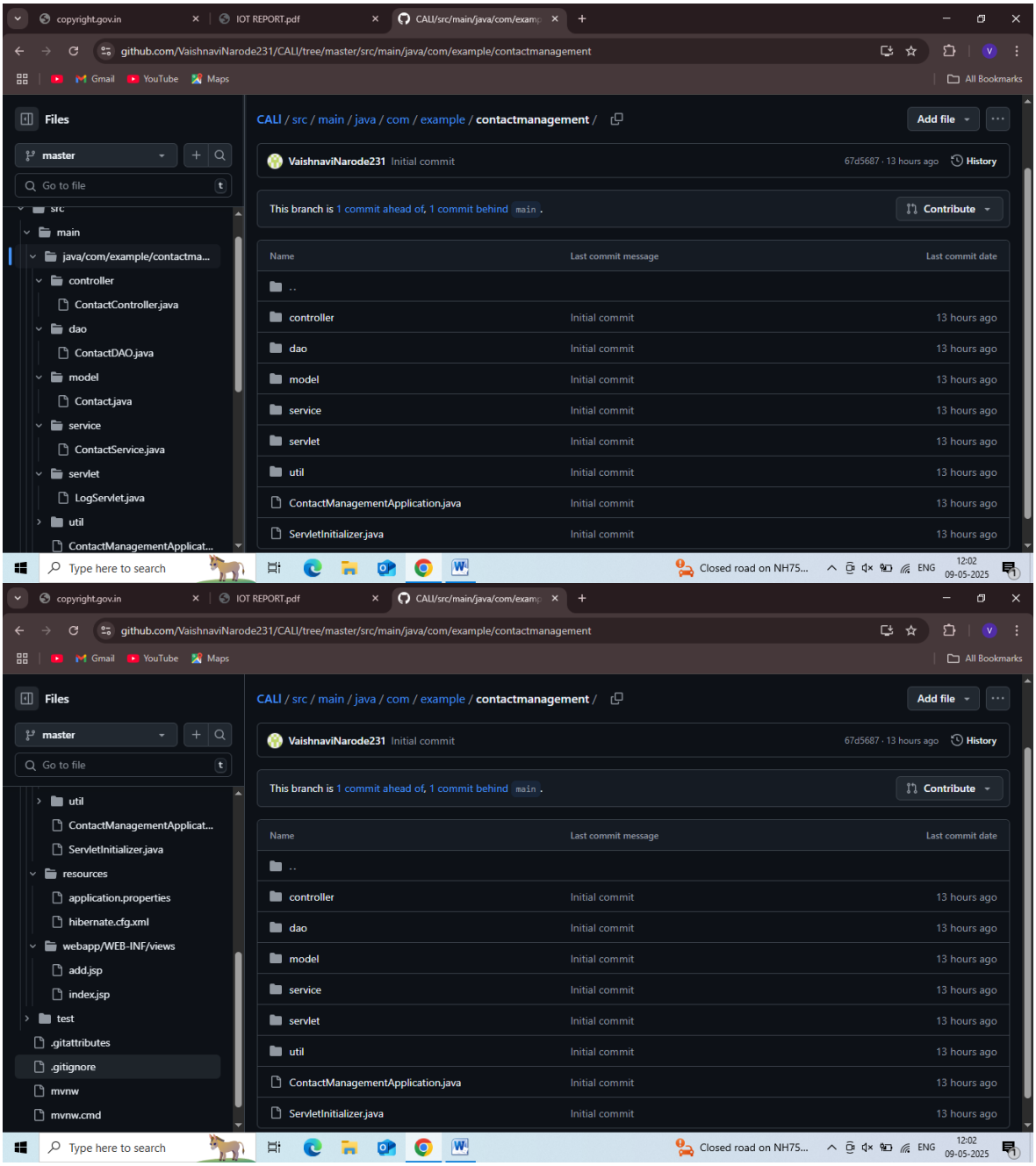
```
java
CopyEdit
package com.example.contactmanagement.servlet;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;

@WebServlet(name = "LogServlet", value = "/log")
public class LogServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.getWriter().println("Log Servlet Accessed");
    }
}
```

JSP: index.jsp

```
jsp
CopyEdit
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head><title>Contact List</title></head>
<body>
<h2>Contacts</h2>
<a href="add">Add New Contact</a>
<table border="1">
   <tr><th>Name</th><th>Email</th><th>Phone</th></tr>
   <c:forEach var="contact" items="${contacts}">
      <tr>
         <td>${contact.name}</td>
         <td>${contact.email}</td>
         <td>${contact.phone}</td>
      </tr>
   </c:forEach>
</table>
</body>
</html>
```

JSP: add.jsp

```
jsp
CopyEdit
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head><title>Add Contact</title></head>
<body>
<h2>Add New Contact</h2>
<form action="save" method="post">
   Name: <input type="text" name="name" required/><br/>
   Email: <input type="email" name="email" required/><br/>
   Phone: <input type="text" name="phone" required/><br/>
   <input type="submit" value="Save"/>
</form>
</body>
</html>
```


copyright.gov.in
IOT REPORT.pdf
CALI/src/main/java/com/exam
github.com/VaishnaviNarode231/CALI/tree/master/src/main/java/com/example/contactmanagement
Gmail
YouTube
Maps
All Bookmarks
Files
master
Go to file
main
java/com/example/contactma...
controller
ContactController.java
dao
ContactDAO.java
model
Contact.java
service
ContactService.java
servlet
LogServlet.java
util
ContactManagementApplicat...
CALI / src / main / java / com / example / contactmanagement /
Add file
VaishnaviNarode231 Initial commit
67d5687 · 13 hours ago
History
This branch is 1 commit ahead of, 1 commit behind main .
Contribute
Name
Last commit message
Last commit date
..
controller Initial commit 13 hours ago
dao Initial commit 13 hours ago
model Initial commit 13 hours ago
service Initial commit 13 hours ago
servlet Initial commit 13 hours ago
util Initial commit 13 hours ago
ContactManagementApplication.java Initial commit 13 hours ago
ServletInitializer.java Initial commit 13 hours ago
Type here to search
Closed road on NH75...
ENG
12:02
09-05-2025
util
ContactManagementApplicat...
ServletInitializer.java
resources
application.properties
hibernate.cfg.xml
webapp/WEB-INF/views
add.jsp
index.jsp
test
.gitattributes
.gitignore
mvnw
mvnw.cmd

CONCLUSION

The Contact Management System has been successfully designed and implemented to meet the essential requirements of a modern contact management application. Key functionalities such as Create, Read, Update, and Delete (CRUD) operations, search and filter capabilities, and basic user authentication have been effectively integrated. The system provides a secure, responsive, and user-friendly interface that ensures efficient management of contact information.

The application is developed using Java Servlets and JSP for server-side logic, along with HTML, CSS, and JavaScript for the frontend. The backend uses MySQL for structured data storage, and the entire system is deployed on an Apache Tomcat server, ensuring platform independence and scalability. The project has also been integrated with Spring Boot, enabling modular configuration, faster development, and future extensibility.

This system is accessible across multiple devices, including desktops and mobile platforms, providing a seamless user experience. Its modular design and use of open-source technologies make it highly maintainable and adaptable for future enhancements.

Looking ahead, the system can be extended with additional features such as cloud-based contact storage, multi-user support, email/SMS integration, tag-based contact categorization, and role-based access control (RBAC). These features would significantly enhance the utility, accessibility, and scalability of the system, making it suitable for both personal and enterprise use.

This project serves as a strong foundation for understanding full-stack development using Java EE technologies, and it demonstrates practical skills in software design, implementation, and deployment.

FUTURE ENHANCEMENTS

 Cloud Integration :Integrate cloud services (such as AWS, Google Cloud, or Azure) to store contacts, ensuring users can access and synchronize their contact list across multiple devices and platforms securely.

 Email Notifications :Implement an automated email notification system to inform users whenever a contact is added, updated, or deleted. This adds transparency and keeps users informed of changes in real-time.

 Multi-User Support:Introduce a login and registration module to allow multiple users to use the application with personalized dashboards. Each user will have their own contact repository and settings.

 Contact Categorization and Tagging:Provide options for users to assign tags or categories (e.g., "Work," "Family," "Friends") to contacts, making it easier to search, group, and manage contacts based on context or purpose.

 Search and Filter Functionality :Add advanced search and filter options (by name, tag, date added, etc.) to help users quickly find specific contacts from large datasets.

 Data Export and Import:Enable users to export their contacts in various formats (CSV, Excel, PDF) and import contacts from external sources (such as Google Contacts or phone backups).

 Mobile-Friendly Responsive Design:Enhance the user interface to be fully responsive and mobile-friendly, ensuring seamless access and usability on smartphones and tablets.

REFERENCES

[1] Oracle Corporation, “Java Servlet Technology,” Oracle, 2024. [Online]. Available: https://www.oracle.com/java/technologies/java-servlet-tec.html.

[2] Oracle Corporation, “JavaServer Pages Technology,” Oracle, 2024. [Online]. Available: https://www.oracle.com/java/technologies/jspt.html.

[3] VMware, Inc., “Spring Boot,” Spring.io, 2024. [Online]. Available: https://spring.io/projects/spring-boot.

[4] Oracle Corporation, “Java JDBC API,” Oracle Java SE Documentation, 2023. [Online]. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/.

[5] Oracle Corporation, “MySQL 8.0 Reference Manual,” Oracle MySQL Developer Zone, 2024. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/.

[6] Apache Software Foundation, “Apache Tomcat 8.5 Documentation,” Apache, 2024. [Online]. Available: https://tomcat.apache.org/tomcat-8.5-doc/.

[7] MDN Web Docs, “HTML: HyperText Markup Language,” 2025. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML.

[8] MDN Web Docs, “CSS: Cascading Style Sheets,” 2025. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS.

[9] GitHub, Inc., “GitHub Docs,” GitHub, 2024. [Online]. Available: https://docs.github.com/.