**FINAL YEAR MAJOR PROJECT REPORT**

**Project Title: Online Job Portal**

**1. Introduction**

The Online Job Portal is a modern web-based application designed to provide job seekers and employers with a centralized digital platform to interact, exchange information, and fulfill recruitment objectives efficiently. Built using technologies like Java for backend processing, MySQL for database management, HTML/CSS (Bootstrap) for front-end interface, and deployed using XAMPP for local server testing, this system serves as a comprehensive solution for managing job applications and job postings. Traditional job application processes are often burdened with inefficiencies, such as delayed communication, limited outreach, and high dependency on physical documentation. This project eradicates those barriers by allowing real-time interaction between employers and job seekers. Job seekers can register, upload resumes, search for job listings based on location, title, and category, and apply directly through the system. Employers, in turn, can create profiles, post job openings, and manage applications. The platform ensures data consistency through normalized database design and secure transactions via role-based access control and password hashing. The system is developed following the Waterfall model of SDLC, which structures development into sequential phases including requirement analysis, design, implementation, testing, and deployment. This methodology has proven ideal due to the project's clearly defined scope and requirements. Moreover, the system ensures that both users and administrators can operate the portal with minimal training thanks to its intuitive interface. The Online Job Portal stands as a functional and scalable prototype that addresses real-world employment challenges and improves user experience for both job seekers and employers.

**2. Feasibility Study**

Before the implementation of the Online Job Portal, a comprehensive feasibility study was conducted to determine the practicality and viability of the system. The feasibility study focused on four critical areas: technical, economic, operational, and legal feasibility. In terms of technical feasibility, the required software and hardware for this project are readily available and widely supported. Technologies like Java, MySQL, HTML, CSS, and Bootstrap have strong community support, extensive documentation, and are ideal for scalable applications. The project was developed and tested using XAMPP, a widely-used local server stack that includes Apache and MySQL, ensuring that all development was conducted in a stable environment. Economically, the project was deemed feasible due to the open-source nature of the tools used, eliminating licensing costs. In addition, the human resource cost was minimized by using academic and in-house development teams. Operational feasibility was ensured through the creation of a user-friendly interface that requires minimal training. Employers and job seekers can navigate the system intuitively, with clear options for job posting, job searching, resume uploads, and user management. Lastly, from a legal standpoint, the project complies with basic data privacy norms such as secure login credentials and limited access to sensitive user data. Given the above factors, the project was considered both practical and cost-effective, setting a strong foundation for development.

**3. Requirement Analysis & Specification**

Requirement analysis is a critical phase in the Software Development Life Cycle (SDLC) that involves gathering functional and non-functional requirements from the stakeholders. For the Online Job Portal, this phase included interactions with students, academic supervisors, and industry professionals to define the core functionalities of the platform. The primary functional requirements included user registration and login for both employers and job seekers, resume uploads, job posting capabilities, job searching features, and an admin dashboard for user and job management. Non-functional requirements encompassed performance, usability, scalability, and security. The system was required to load pages quickly, support multiple concurrent users, and ensure data security through hashed passwords and role-based access control. Additional specifications include browser compatibility, mobile responsiveness using Bootstrap, and server-side validation using Java. The system must also ensure 24/7 availability when deployed to a live server. These requirements were documented in detail and approved before proceeding to the design phase.

**4. SRS / Hardware Description**

The Software Requirement Specification (SRS) document provides a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. For this project, the SRS encompasses user roles (Admin, Employer, Job Seeker), their capabilities (post job, search job, manage users), and system constraints (browser-based, responsive UI, real-time validation). From a hardware perspective, the development system used was a standard laptop with a Core i3 processor, 4GB RAM, and 1TB HDD. The minimal server configuration for hosting the application includes an Apache server (bundled with XAMPP), MySQL database engine, and JVM (Java Virtual Machine). The application is lightweight and can run efficiently on basic hosting services. A strong internet connection is required only for deployment and remote access; otherwise, the system runs seamlessly on a localhost.

**5. UML Diagram / Hardware Diagram**

UML (Unified Modeling Language) diagrams were created to visually represent the system architecture, interactions, and data flows. These included use case diagrams for Admin, Employer, and User modules; activity diagrams showing the flow of actions like registration and job posting; and class diagrams outlining the system's object-oriented design. The use case diagrams identified the relationships between actors and their tasks, such as an employer posting a job or a user applying for it. Activity diagrams helped model the decision points within job search and application processes. ER diagrams (Entity Relationship Diagrams) depicted how data entities such as Users, Jobs, Applications, and Employers are related within the database schema. A simple hardware diagram was also prepared showing the server-client model, including the role of the XAMPP stack in hosting the web application locally. The hardware architecture follows a standard client-server model where the browser (client) sends requests to the Apache server (hosted via XAMPP), which communicates with the MySQL database for data operations.

**6. Sample Coding**

Below is a simplified code snippet for the login module in Java using JDBC:

try {

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/jobportal", "root", "");

String sql = "SELECT \* FROM users WHERE email=? AND password=?";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, email);

pst.setString(2, password);

ResultSet rs = pst.executeQuery();

if(rs.next()) {

out.println("Login successful");

} else {

out.println("Invalid credentials");

}

} catch(Exception e) {

e.printStackTrace();

}

This code connects to the MySQL database and verifies user credentials against stored data. It uses prepared statements to prevent SQL injection and handles exceptions gracefully.

**7. Tools and Technologies Used**

**7.1 HTML (Hypertext Markup Language)**

HTML stands for HyperText Markup Language. It is the foundational markup language used to structure and present content on the World Wide Web. HTML provides the basic building blocks for any web page, allowing developers to define elements like headers, paragraphs, images, links, forms, and multimedia content. In this project, we have used **HTML5**, the latest version, which offers several new features such as semantic elements (<section>, <article>, <nav>), multimedia tags (<audio>, <video>), and better form controls (<input type="date">, <input type="email">). The choice of HTML5 was deliberate because it ensures compatibility with modern web browsers, enhances accessibility, and improves search engine optimization (SEO). It also supports responsive design when combined with CSS and Bootstrap, making the portal user-friendly on both desktop and mobile platforms.

**7.2 CSS (Cascading Style Sheets)**

CSS is a style sheet language used to describe the visual presentation of a web page written in HTML. It controls the layout, colors, fonts, spacing, and overall look and feel of a website. For this project, CSS was essential in providing a consistent and professional user interface. We specifically used **Bootstrap**, a popular CSS framework, which allowed us to design a responsive and visually appealing layout quickly. Bootstrap comes with a wide range of pre-designed components such as navbars, buttons, forms, and cards. It also uses a powerful grid system that enables responsive design, ensuring that the application looks great on devices of all sizes.

**7.3 Java**

Java is a high-level, object-oriented programming language developed by Sun Microsystems and now maintained by Oracle. It is known for its portability, robustness, and scalability. In our project, Java is used for server-side programming. It handles backend operations such as user authentication, database interaction, job posting logic, and application processing. We used Java Servlets and Java Database Connectivity (JDBC) for interacting with the MySQL database. Java was selected because it supports MVC architecture, has extensive library support, and provides strong security features. Its platform-independent nature also means the application can be deployed on any operating system with a Java Virtual Machine (JVM).

**7.4 MySQL**

MySQL is an open-source relational database management system. It is used to store all the data related to users, employers, jobs, and applications. MySQL supports complex queries, joins, and indexing, which help in retrieving data efficiently. The database in this project is normalized up to 3NF (Third Normal Form) to ensure data integrity and reduce redundancy. The use of MySQL ensures that the data layer is both powerful and secure.

**7.5 XAMPP**

XAMPP stands for **Cross-Platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P)**. It is an open-source local web server package developed by Apache Friends. XAMPP simplifies the installation of Apache, MySQL, and PHP/PERL to create a local server environment. For our project, we used XAMPP primarily for two components: **Apache**, which acts as the HTTP server to host the portal locally, and **MySQL**, which handles database operations. It provides an easy-to-use control panel to start or stop services, configure ports, and monitor performance. XAMPP allows for testing and development without needing an active internet connection or a live server, making it ideal for academic and development purposes.

**7.6 Waterfall Model**

The Waterfall Model is a linear sequential approach to software development. It divides the software development life cycle into distinct phases: **Requirement Gathering, System Design, Implementation, Integration and Testing, Deployment, and Maintenance**. Each phase must be completed before the next begins. This model was chosen for the Online Job Portal because the project requirements were well understood and unlikely to change. The advantages of using the Waterfall model include clear documentation, structured progress, and easy management. Since our team followed academic timelines and predefined requirements, Waterfall provided the discipline and clarity needed to complete each phase effectively and on schedule.

**7. Testing Cases**

Testing ensures that all modules work as expected and that the application is free from bugs. Manual testing was conducted on all core modules, including registration, login, job posting, and application submission. Test cases were written for each scenario:

* **Test Case 1**: Valid user login — Expected Result: Redirect to user dashboard
* **Test Case 2**: Invalid login — Expected Result: Show error message
* **Test Case 3**: Job posting by admin — Expected Result: Job listed on job board
* **Test Case 4**: Resume upload — Expected Result: File saved to user profile  
  Each test case passed during functional testing. Cross-browser testing was also performed to ensure UI consistency across Chrome, Firefox, and Edge.

**8. Advantages**

The Online Job Portal offers multiple advantages:

* **Convenience**: Users can apply for jobs from anywhere at any time.
* **Efficiency**: Admins can manage listings and user data in real time.
* **Scalability**: The architecture supports adding more users, features, and modules.
* **Security**: Password hashing and session management ensure secure operations.
* **Responsiveness**: With Bootstrap, the UI adapts to mobile, tablet, and desktop views.
* **Modularity**: Each feature is modularly designed for easy updates and debugging.

**9. Limitations**

Despite its robust functionality, the system has certain limitations:

* Limited to local deployment unless hosted on a cloud or server.
* No multi-language support — interface only available in English.
* No integration with third-party APIs (e.g., LinkedIn or email services).
* Lacks real-time notification system.
* No built-in analytics dashboard for job trends and user statistics.

**10. Future Prospects**

Several enhancements can be made to improve the system:

* Deploy the portal to a public server or cloud platform.
* Introduce mobile apps using React Native or Kotlin.
* Integrate with email systems for notifications.
* Use AI to suggest jobs based on resume data.
* Add employer-user chat and interview scheduling.
* Enable third-party login options (Google, Facebook).

**11. Conclusions**

The Online Job Portal successfully delivers a platform where employers and job seekers can interact efficiently. Through the application of Java and MySQL under the Waterfall model, the system is structured, secure, and functionally rich. The portal solves real-world problems in recruitment and demonstrates the effectiveness of academic software development projects. While it has some limitations, its extendable design ensures that future improvements can be seamlessly integrated. This project not only fulfills the academic criteria for a major final year submission but also has real-life applicability that makes it relevant beyond the classroom.

**12. Bibliography**

1. Java Documentation - <https://docs.oracle.com/javase/>
2. MySQL Reference Manual - <https://dev.mysql.com/doc/>
3. Bootstrap Official Docs - <https://getbootstrap.com/>
4. XAMPP Control Panel - <https://www.apachefriends.org/>
5. Waterfall Model - <https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>
6. HTML & CSS Basics - [https://www.w3schools.com](https://www.w3schools.com/)
7. UML Diagrams - [https://www.visual-paradigm.com](https://www.visual-paradigm.com/)
8. JDBC Guide - <https://docs.oracle.com/javase/tutorial/jdbc>