

## **1. INTRODUCTION**

### **1.1 Background and Context of the Project:**

Blue-collar workers, who are employed in manual labor sectors such as construction, manufacturing, logistics, and maintenance, often face difficulties in accessing reliable job opportunities. Many skilled workers in these fields rely on informal networks or word-of-mouth to find employment, which leads to inefficiencies in the job search and hiring process. Additionally, workers in these industries may have limited access to advanced technology, making it challenging to navigate digital job portals or find opportunities through conventional online platforms. The lack of structured, accessible, and efficient job portals for blue-collar workers has created a gap in employment opportunities, contributing to higher levels of unemployment and underemployment in this sector. This project addresses this gap by creating a user-friendly, accessible, and technology-driven platform specifically designed for blue-collar workers.

### **1.2 Problem Statement and Objectives:**

The primary issue addressed by this project is the lack of accessible and efficient job opportunities for blue-collar workers. The objective of the project is to develop a digital platform that bridges the gap between skilled blue-collar laborers and employers by offering an intuitive, easy-to-use portal where job seekers can find relevant opportunities and employers can efficiently hire the right candidates. The main objectives are:

- To create a user-friendly platform that is easy to navigate, even for individuals with limited digital literacy.
- To facilitate direct communication between employers and potential employees, streamlining the hiring process.
- To provide search filters based on skill set, location, and experience level, ensuring that blue-collar workers find relevant jobs quickly.
- To reduce unemployment and improve economic stability by helping blue-collar workers access reliable job opportunities.

### **1.3 Scope and Significance of the Project:**

The scope of the project involves developing a web-based job portal that caters exclusively to blue-collar workers, providing them with a structured and organized platform to access job opportunities in industries such as construction, manufacturing, logistics, and maintenance. This platform will benefit workers with limited access to technology by ensuring it is mobile-responsive and easy to use. The significance of this project lies in its potential to create a more efficient, accessible, and fair hiring process, reducing unemployment in the blue-collar workforce. Additionally, it will contribute to economic stability by ensuring that employers can efficiently hire skilled workers while providing workers with better access to employment opportunities. This project aligns with broader societal goals of reducing unemployment and promoting economic growth, particularly for those in labor-intensive industries.

### **1.4 Brief Overview of the Methodology/Approach:**

The project will be developed using a combination of HTML, CSS, JavaScript, and PHP. The front-end will be designed with simplicity and accessibility in mind, ensuring that users with varying levels of digital literacy can navigate the platform easily. JavaScript will be used to enhance user interactivity, such as dynamic job filters and application updates. The back-end will be powered by PHP, handling user authentication, job postings, and the management of user profiles. A MySQL database will securely store user and job data. The portal will also feature a direct messaging system to facilitate communication between employers and workers, streamlining the hiring process. Testing and optimization will ensure the platform is mobile-friendly and fully responsive across devices. User feedback will be incorporated into the development process to ensure the platform meets the needs of blue-collar workers and employers alike.

## **2. LITERATURE REVIEW**

### **2.1 EXISTING SYSTEM**

The existing job search platforms primarily cater to white-collar jobs, with complex interfaces that are not user-friendly for blue-collar workers. Many of these platforms require advanced digital skills, limiting accessibility for workers with minimal technological experience. Additionally, blue-collar workers often rely on informal job searches, word-of-mouth, or manual applications, which are time-consuming and inefficient. Employers also struggle to find skilled labor quickly due to the lack of a dedicated platform for blue-collar job recruitment.

### **2.2 PROPOSED SYSTEM**

The proposed system is a dedicated Job Portal for Blue-Collar Workers that simplifies job searching and recruitment. It provides a mobile-friendly, easy-to-use interface designed for workers with limited digital literacy. The system includes real-time job search filters, location-based job recommendations, and direct communication between employers and workers. Employers can efficiently post jobs and manage applications, while workers can track job applications and receive updates. A secure MySQL database ensures safe storage of user profiles, job listings, and application records. The admin panel enables streamlined user and job management.

### **2.3 FEASIBILITY STUDY**

#### **1. Technical Feasibility:**

The system is developed using HTML, CSS, JavaScript, PHP, and MySQL, which are widely used, reliable, and cost-effective technologies. The responsive design ensures accessibility on various devices, including mobile phones, making it feasible for blue-collar workers.

#### **2. Economic Feasibility:**

The platform is cost-effective since it uses open-source technologies, reducing development and maintenance costs. By streamlining recruitment, it saves time and resources for both employers and job seekers, making it a valuable investment.

### **3. Operational Feasibility:**

The system is easy to use, even for workers with limited digital experience. Features like skill-based filtering, location-based job recommendations, and direct communication make it efficient for job seekers and employers, ensuring smooth adoption.

This feasibility study confirms that the proposed Job Portal for Blue-Collar Workers is practical, cost-effective, and beneficial for improving employment opportunities in manual labor industries.

### **3. METHODOLOGIES**

#### **3.1 Tools, and Technologies used in the project:**

##### **1. HTML (Hypertext Markup Language):**

- **Role:** HTML is the backbone of the website and is used to structure the content of the job portal.
- **Usage:**
  - **Page Layout:** HTML defines the structure of the webpage, including sections for job listings, search filters, application forms, and user profiles.
  - **Forms:** It is used to create registration forms, login forms, job application forms, and employer profile submission forms.
  - **Accessibility:** Ensures that content is well-structured for all users, including those using assistive technologies.

##### **2. CSS (Cascading Style Sheets):**

- **Role:** CSS is used for styling and making the platform visually appealing, ensuring it is responsive and accessible on mobile devices.
- **Usage:**
  - **Responsive Design:** CSS ensures the portal is mobile-friendly, which is crucial for blue-collar workers who may only have access to smartphones.
  - **Styling:** It is used to style the overall layout, fonts, colors, buttons, and interactive elements, making the platform visually engaging and easy to navigate.
  - **User Interface (UI):** Through CSS, the platform's design is made simple and intuitive, ensuring that workers with limited digital literacy can still use it effectively.

##### **3. JavaScript:**

- **Role:** JavaScript is used to make the platform interactive and dynamic, improving user experience through real-time features and client-side functionality.

- **Usage:**

- **Interactive Features:** JavaScript enables dynamic search filtering, where workers can filter jobs based on criteria like location, skill set, and experience level without refreshing the page.
- **Job Alerts and Updates:** JavaScript handles real-time updates, such as notifying users of new job openings or the status of their job application.
- **Form Validation:** Before submitting forms (like registration or job applications), JavaScript is used to validate the data entered, ensuring that all fields are filled correctly and reducing user errors.
- **Mobile Optimization:** JavaScript ensures that the platform's features, such as search filters and job applications, function smoothly on mobile devices, which is critical for blue-collar workers.

#### 4. PHP (Hypertext Preprocessor):

- **Role:** PHP is used for server-side processing, managing data flow between the front-end and the database, and handling back-end functionality like user authentication and job posting.
- **Usage:**
  - **User Authentication:** PHP handles user login and registration, ensuring that both blue-collar workers and employers can securely access their profiles.
  - **Job Posting & Application Management:** Employers use PHP to post new job openings, while workers can submit applications. PHP ensures that these processes are securely managed and stored.
  - **Communication:** PHP can be used to facilitate direct messaging between workers and employers, ensuring that they can communicate efficiently about job opportunities.
  - **Data Processing:** PHP processes forms, handles the logic behind the search filters, and manages the database queries for job postings and worker profiles.

## 5. MySQL (Database Management):

- **Role:** MySQL is a relational database management system used to store, retrieve, and manage user data, job listings, and applications.
- **Usage:**
  - **Data Storage:** MySQL stores user data, including profiles for both workers and employers, job listings, applications, and other essential information.
  - **Search Functionality:** MySQL allows for efficient querying of job listings based on various filters like location, skills, and experience. This helps employers find the right candidates quickly and allows workers to find jobs that match their skills.
  - **Security & Integrity:** MySQL ensures that data is stored securely and can be accessed or updated without issues. It is also used to validate the integrity of job applications and other records.
  - **Admin Panel:** PHP and MySQL together support an admin panel where administrators can manage user accounts, job listings, and applications. This ensures the platform remains organized and up-to-date.

## 3.2 Description of the Project's Architecture/Design

The **Job Portal for Blue-Collar Workers** follows a **three-tier architecture** that separates the system into distinct layers: **Presentation Layer**, **Business Logic Layer**, and **Data Layer**. This structure ensures efficient handling of user requests, smooth data processing, and secure storage of information.

### 1. Presentation Layer (Front-End)

- **Technologies Used:** HTML, CSS, JavaScript
- **Role:** The front-end layer is responsible for the user interface (UI) and user experience (UX). It presents the platform to the user in an intuitive, easy-to-use, and visually appealing way.

**Components:**

- **User Interface (UI):**
  - **HTML** is used to structure the content of the webpage (e.g., forms, job listings, user profiles).
  - **CSS** ensures the platform has a clean, mobile-responsive design. It adapts the layout for various screen sizes, ensuring accessibility on both desktop and mobile devices (important for blue-collar workers).
  - **JavaScript** is responsible for enhancing interactivity, such as dynamic job filtering, real-time updates (job alerts, application statuses), and client-side form validation.
- **Responsive Design:**
  - The design is mobile-first, ensuring that blue-collar workers, who may have limited access to computers and mainly use mobile phones, can easily navigate and access all features.
- **User Interaction:**
  - Workers can register, create profiles, search for jobs, apply for opportunities, and track their applications.
  - Employers can post jobs, view applicant profiles, and communicate with potential candidates directly through the platform.

**2. Business Logic Layer (Back-End)**

- **Technologies Used:** PHP
- **Role:** This layer processes the core business logic of the application. It handles user authentication, job postings, search filters, application submissions, and communication between users.

**Components:**

- **User Authentication:**

- PHP manages the registration, login, and session management of both blue-collar workers and employers. It ensures security by verifying user credentials and protecting sensitive data.
- **Job Management:**
  - Employers can post new job opportunities, update existing listings, and track applications.
  - Workers can browse available jobs, filter them by skill set, location, and experience, and apply to suitable jobs.
- **Communication System:**
  - A messaging feature enables direct communication between employers and workers. PHP handles this communication and stores messages in the database for easy retrieval.
- **Search and Filter Logic:**
  - PHP processes the search queries, dynamically fetching job listings that match criteria such as skills, experience level, and location.

### 3. Data Layer (Database)

- **Technologies Used:** MySQL
- **Role:** The data layer is responsible for storing, managing, and retrieving data. It ensures that all user data, job listings, applications, and other related information are securely stored and easily accessible.

### Components:

- **Database Structure:**
  - **Users Table:** Stores details about workers and employers, such as names, contact information, and login credentials.
  - **Jobs Table:** Stores job listings, including job descriptions, required skills, locations, and employer information.

- **Applications Table:** Keeps records of job applications, tracking the worker who applied, the job they applied for, and the status of their application.
- **Messages Table:** Stores messages exchanged between workers and employers for future reference.
- **Data Processing:**
  - The database is queried to fetch job listings based on the search filters (e.g., location, skill, and experience) provided by the user. It also stores user-generated data (profiles, applications) and keeps track of interaction histories (applications, messages).
- **Security:**
  - **MySQL** ensures the security of sensitive data such as passwords and personal information through encryption and proper access control mechanisms.

### System Flow:

#### 1. User Registration/Login:

- Users (workers and employers) register by providing their personal details. PHP validates the registration, stores data in the MySQL database, and allows users to log in securely.

#### 2. Job Search and Application:

- Blue-collar workers use the search functionality to filter jobs based on criteria (location, skills, experience level). The back-end (PHP) queries the database and returns results.
- Workers can apply for jobs, and their applications are stored in the database, with PHP handling the submission process.

#### 3. Employer Job Posting:

- Employers create job listings by providing job descriptions and requirements. This data is processed by PHP and stored in the MySQL database, making it visible to workers.

**4. Employer-Worker Communication:**

- Both parties can message each other through an in-platform messaging system. PHP processes the communication and stores the messages in the database.

**5. Application Status Updates:**

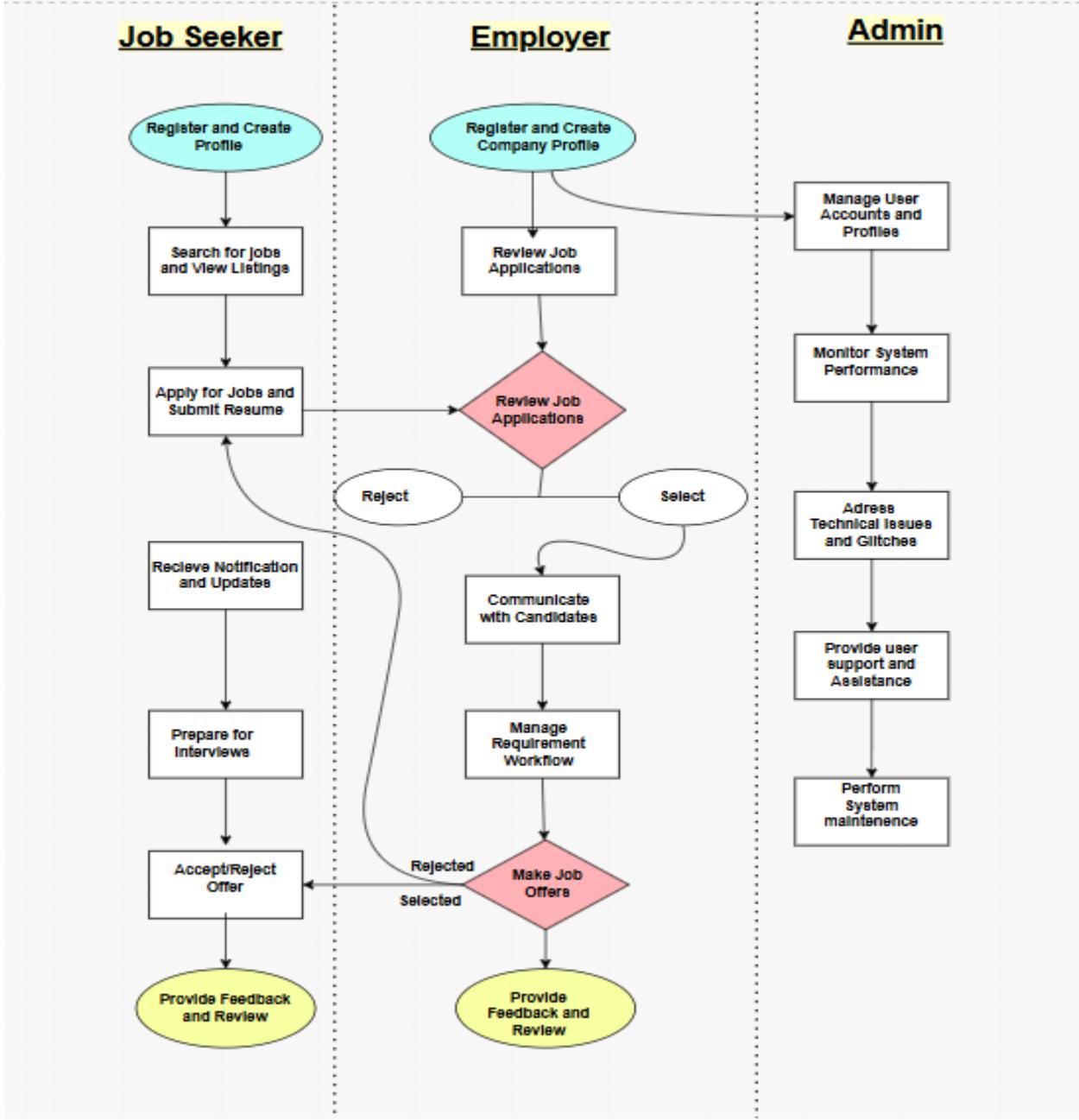
- Workers receive updates on their application statuses (e.g., "Under Review," "Accepted," or "Rejected"). PHP manages the status transitions and sends real-time notifications.

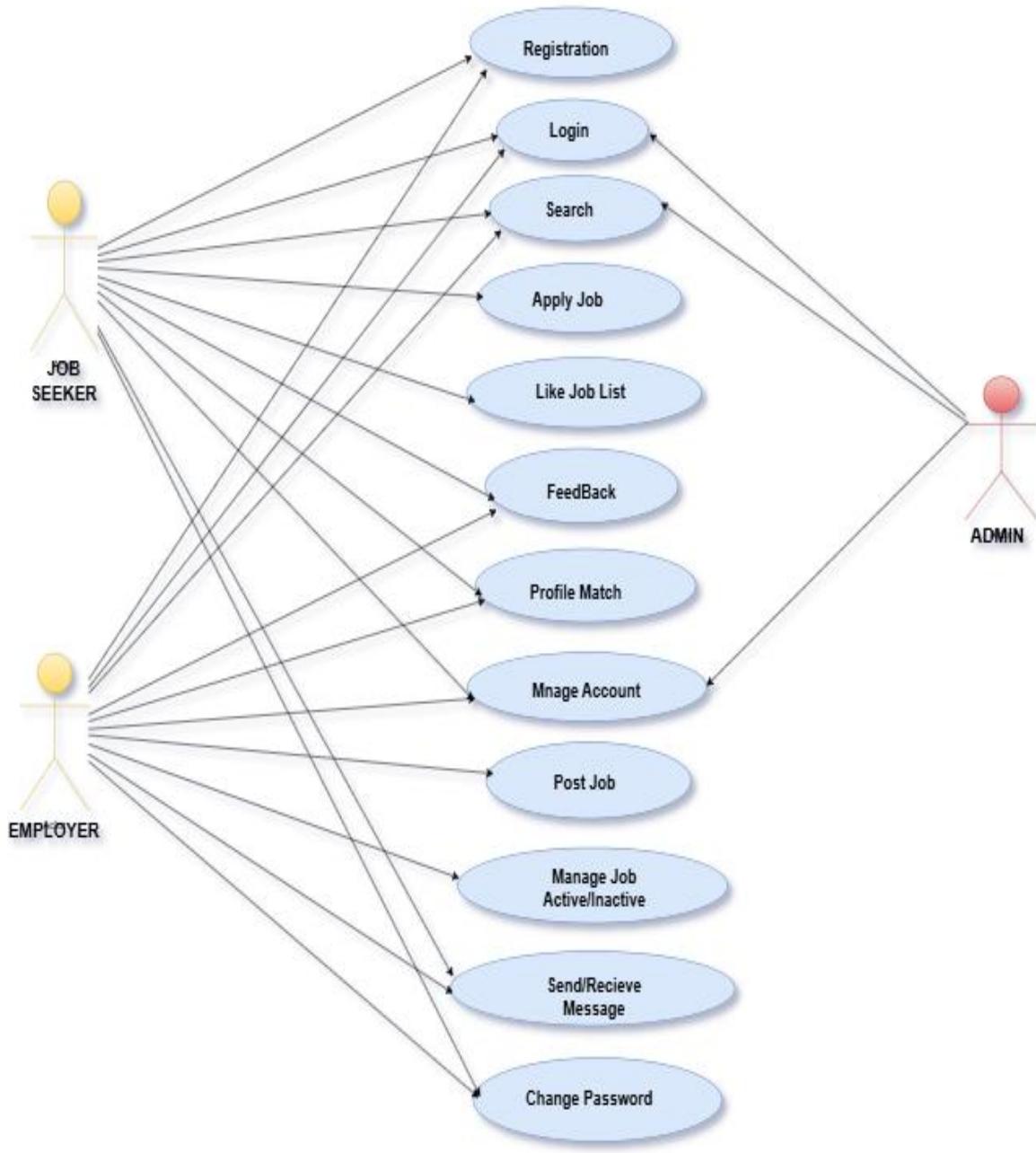
**Security & Scalability Considerations:**

- **Security:** The platform ensures secure handling of personal information by encrypting passwords and using HTTPS for secure communication. PHP sessions handle user authentication securely.
- **Scalability:** The database is designed to handle a growing number of users, job listings, and applications. Indexing is implemented to optimize database queries for better performance.

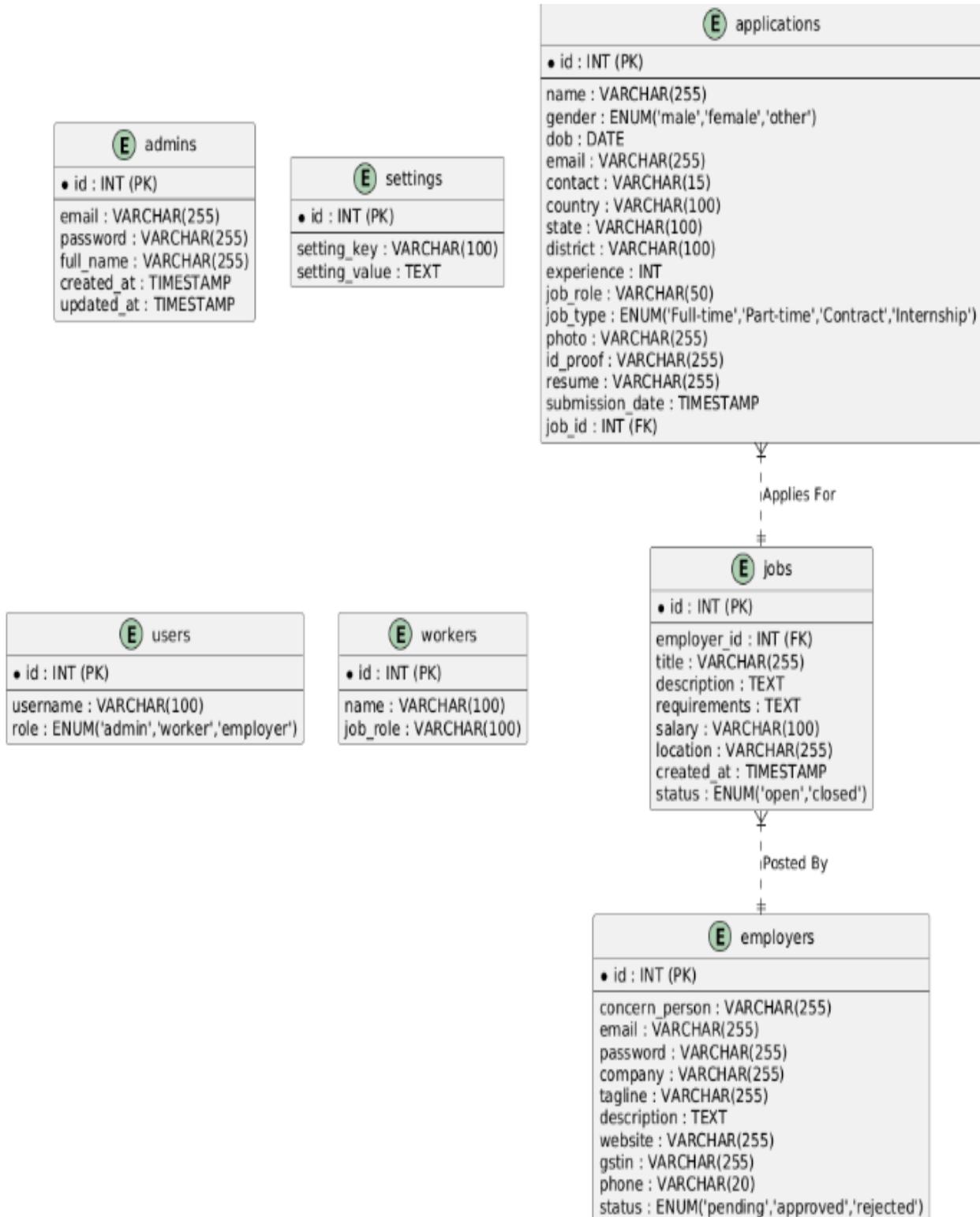
## 4. SYSTEM DESIGN

### 4.1 Data Flow Diagram:

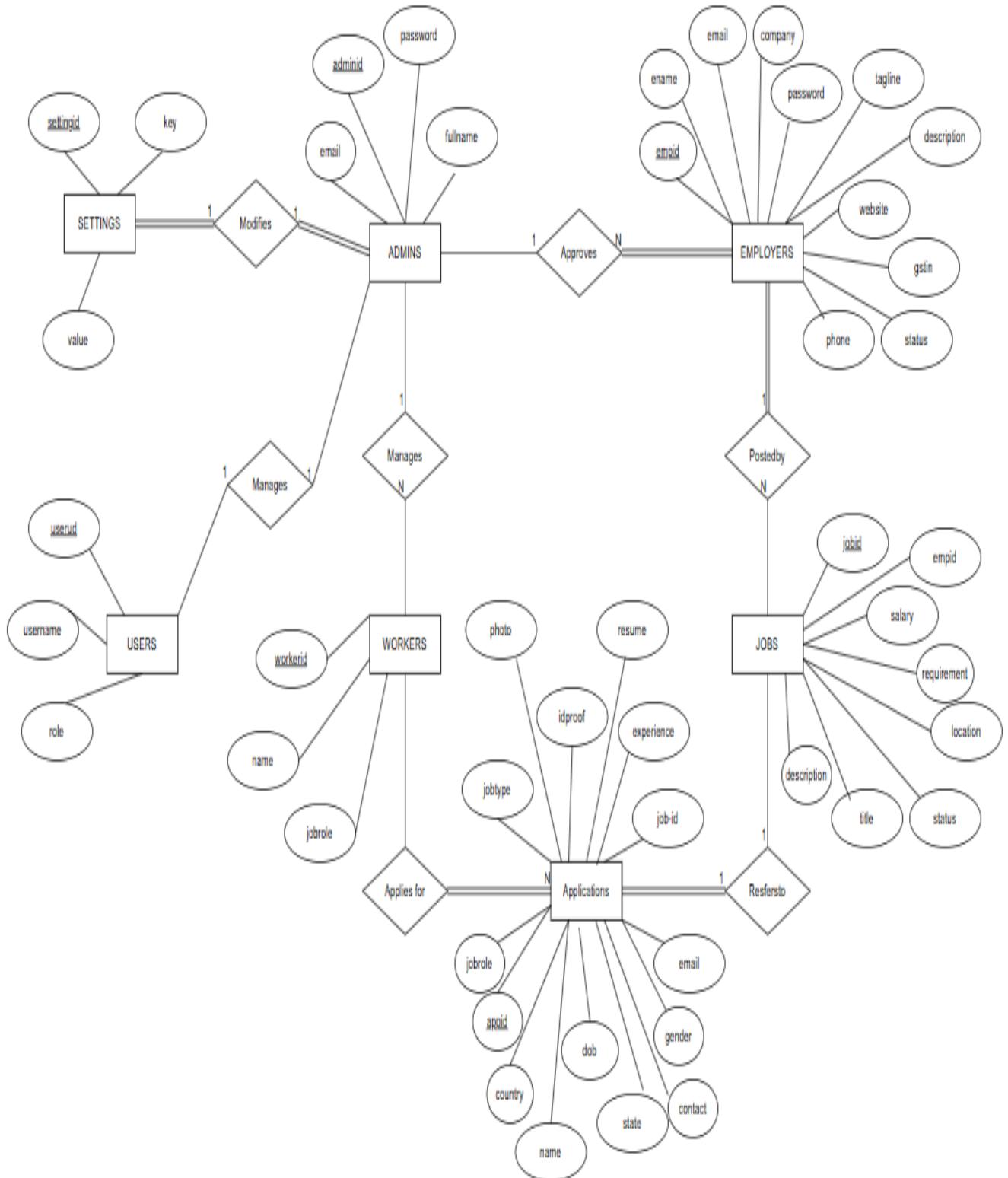


**4.2 Use-case Diagram:**

### 4.3 Schema Diagram:



#### 4.4 Entity Relationship Diagram:



## **5. IMPLEMENTATION**

### **5.1 Description about modules**

#### **1. Admin Module**

The Admin Module is responsible for managing the overall platform operations. It provides a user-friendly interface for administrators to monitor and control job postings, user accounts, and applications.

Key Features:

- User Management: View, verify, and manage employer and jobseeker accounts.
- Job Posting Management: Approve or reject job listings posted by employers.
- Application Tracking: Monitor job applications submitted by jobseekers.
- Reports & Analytics: Generate reports on job postings, user activity, and hiring trends.
- Security & Maintenance: Ensure data security, prevent unauthorized access, and maintain smooth platform operations.

#### **2. Employer Module**

The Employer Module allows businesses and recruiters to post job opportunities, manage applications, and communicate with potential candidates.

Key Features:

- Employer Registration & Profile Management: Employers can register, log in, and manage their company profiles.
- Job Posting: Employers can post job openings with details such as job description, skills required, salary, and location.
- Application Management: View and manage applications received for job postings.

- Candidate Search & Communication: Employers can filter jobseekers based on skills, experience, and location, and directly communicate with them.

### 3. Jobseeker Module

The Jobseeker Module is designed to help blue-collar workers easily search for jobs, apply for positions, and track their applications.

Key Features:

- Jobseeker Registration & Profile Creation: Users can sign up, create profiles, and upload their resumes.
- Job Search & Filters: Search for jobs based on location, skills, industry, and salary range.
- Application Submission: Apply for jobs directly through the platform.
- Application Tracking: View the status of submitted applications in real-time.
- Employer Communication: Connect directly with employers for job-related queries.

## 5.2 Code snippets & Screenshots of modules

### 1.Front page:

The homepage introduces HomeCare Experts, a platform connecting blue-collar job seekers with employers. It highlights various job categories and explains its role in bridging employment gaps.

## FrontPage

```
<nav class="navbar navbar-expand-sm">  
  <div class="container-fluid">  
    <a class="navbar-brand fs-3 font-sans-serif" href="main.html"></a>  
    <button  
      class="navbar-toggler"  
      type="button"  
      data-bs-toggle="collapse"  
      data-bs-target="#navbarSupportedContent"  
      aria-controls="navbarSupportedContent"  
      aria-expanded="false"  
      aria-label="Toggle navigation">  
      <span class="navbar-toggler-icon"></span>  
    </button>  
  </div>  
  
  <div class="collapse navbar-collapse" id="navbarSupportedContent">  
    <ul class="navbar-nav me-auto mb-2 mb-lg-0" id="nav-content"></ul>  
    <a class="nav-link fs-5 fst-italic" href="main.html">Home</a>  
    <a class="nav-link fs-5 fst-italic" href="jobs.php">Jobs</a>  
    <a class="nav-link fs-5 fst-italic" id="scrollButton">About</a>  
    <a class="nav-link fs-5 fst-italic" id="scrollButton1">Contact</a>  
    <a class="nav-link fs-5 fst-italic" href="emp_login.html"><i class="bx bxs-user">Login/Signup</i></a>
```

```
</div>
</nav>
<div class="header-content">
<h1>HomeCare Experts</h1>
<p>Connecting hardworking individuals with the best opportunities</p>
</div>
</header>
<section id="aboutSection" class="p-4">
<br />
<h2>Connect People with the</h2>
<h2>Right Jobs for Them</h2>
<p>
We are specialists in bridging the gaps between blue collar job
vacancies and seekers. We nurture a vast database of competent blue
collar job seekers bestowing the best info to credible recruiters. Our
application will assist candidates in searching for the best paying blue
collar jobs. It will also assist employers in hiring the best workforce
for a multitude of sectors.
</p>
<br />
</section>
```

## 2. Admin Module:

The screenshot shows the Admin Panel dashboard with a sidebar on the left containing links for Admin Panel, Dashboard, Manage Admins, and Logout. The main area has a red header bar with the word "Dashboard". Below it, a message says "Welcome to the HomeCare Experts Admin Panel." A section titled "Registered Employers" displays a table with columns: Company Name, Email, Contact, Status, and Actions. The table contains six rows of employer data, each with a "View" and "Delete" button under the Actions column.

Company Name	Email	Contact	Status	Actions
Aradhy Engineering	allayyanavarprajwal@gmail.com	9008341720	Approved	<a href="#">View</a> <a href="#">Delete</a>
M/s Rohit Industries	sushantpatil.ssp11@gmail.com	8073495689	Approved	<a href="#">View</a> <a href="#">Delete</a>
Ashok Iron	2g23mc102@students.git.edu	9611748034	Approved	<a href="#">View</a> <a href="#">Delete</a>
Laxmi Traders	2g23mc082@students.git.edu	7892621416	Approved	<a href="#">View</a> <a href="#">Delete</a>
Shri Hanuman Traders	santoshsanniganavar@gmail.com	9916062923	Pending	<a href="#">Approve</a> <a href="#">Reject</a> <a href="#">View</a> <a href="#">Delete</a>
Laxmi Traders	2g23mc063@students.git.edu	7829621416	Approved	<a href="#">View</a> <a href="#">Delete</a>

This page allows admins to manage registered employers by approving, rejecting, or deleting them. It provides details like company name, email, contact, and status.

### Admin

```
<?php foreach ($employers as $employer): ?>

<tr>

<td><?= htmlspecialchars($employer['company']) ?></td>

<td><?= htmlspecialchars($employer['email']) ?></td>

<td><?= htmlspecialchars($employer['phone']) ?></td>

<td class="status-<?= $employer['id'] ?>">

<?php

$status = !empty($employer['status']) ? ucfirst(strtolower($employer['status'])) : 'Pending';

?>
```

```
<span class="badge <?= $status ?>"><?= $status ?></span>

</td>

<td>

<?php if ($status === 'Pending'): ?>

<button class="approve-btn" data-id="<?= $employer['id'] ?>" data-email="<?= $employer['email'] ?>" data-name="<?= $employer['company'] ?>" data-status="Approved">Approve</button>

<button class="reject-btn" data-id="<?= $employer['id'] ?>" data-email="<?= $employer['email'] ?>" data-name="<?= $employer['company'] ?>" data-status="Rejected">Reject</button>

<?php endif; ?>

<a href="view_employer.php?id=<?= $employer['id']; ?>" class="btn btn-info btn-sm">View</a>

<a href="delete_employer.php?id=<?= $employer['id']; ?>" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to delete this employer?');">Delete</a>

</td>

</tr>

<?php endforeach; ?>
```

### 3. Employer Module:

Job Title	Applicants	Status	Actions
Plumber	3	open	<a href="#">View</a> <a href="#">Close Job</a> <a href="#">Delete</a>
Electrician	6	open	<a href="#">View</a> <a href="#">Close Job</a> <a href="#">Delete</a>
Geyser Repair	0	open	<a href="#">View</a> <a href="#">Close Job</a> <a href="#">Delete</a>

This page enables employers to manage job postings and view applicants. Employers can view, close, or delete job listings based on hiring needs.

#### Employer

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Employer Panel</a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item"><a class="nav-link" href="update_profile.php">Update Profile</a></li>
```

```
<li class="nav-item"><a class="nav-link" href="post_job.php">Post a Job</a></li>

<li class="nav-item"><a class="nav-link" href="view_applicants.php">View
Applicants</a></li>

<li class="nav-item"><a class="nav-link" href="logout.php">Logout</a></li>

</ul>

</div>

</nav>

<div class="container mt-4">

<h2>Welcome, <?php echo htmlspecialchars($employer['name']); ?></h2>

<p>Email: <?php echo htmlspecialchars($employer['email']); ?></p>

<p>Status: <?php echo htmlspecialchars($employer['status']); ?></p>

</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
```

#### 4. Jobseeker Module:

The screenshot shows the 'Application Form' page for job seekers. The form is divided into several sections: Personal Information, Verify OTP, Required Documents, and a central input area. The Personal Information section includes fields for Full Name (Shreedhar Patilashetti), Gender (Male), Date of Birth (15-02-2000), Email (shreedharpatilashetti@gmail.com), and Mobile Number (953235888). The Verify OTP section includes a field for Enter OTP (656742) and a dropdown for Country (India), State (Karnataka), and City (Belagavi). The Required Documents section contains three file upload fields: Passport Size Photo (Choose File: carmen\_knowles\_CDMGVRj01.jpg), Government ID Proof (Choose File: Screenshot (13).png), and Resume (Choose File: project resume1.pdf). A large blue button at the bottom right is labeled 'Submit Application'.

This page allows job seekers to apply for positions by entering personal details, verifying OTP, selecting a job type, and uploading required documents. It ensures a secure and streamlined application process.

**Employee**

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
  
    $name = htmlspecialchars(trim($_POST['name']));  
  
    $gender = htmlspecialchars(trim($_POST['gender']));  
  
    $dob = htmlspecialchars(trim($_POST['dob']));  
  
    $email = htmlspecialchars(trim($_POST['email']));  
  
    $contact = htmlspecialchars(trim($_POST['contact']));  
  
    $country = htmlspecialchars(trim($_POST['country']));  
  
    $state = htmlspecialchars(trim($_POST['state']));  
  
    $district = htmlspecialchars(trim($_POST['district']));  
  
    $experience = htmlspecialchars(trim($_POST['experience']));  
  
    $job_role = htmlspecialchars(trim($_POST['job-role'])); // Corrected variable name  
  
    $job_type = htmlspecialchars(trim($_POST['job-type']));  
  
    // ◆ Fetch job_id from jobs table based on job role  
  
    $job_id = null;  
  
    $job_query = "SELECT id FROM jobs WHERE title = ? LIMIT 1"; // Adjust column name if  
needed  
  
    $stmt = $conn->prepare($job_query);  
  
    $stmt->bind_param("s", $job_role);  
  
    $stmt->execute();  
  
    $stmt->bind_result($job_id);
```

```
$stmt->fetch();

$stmt->close();

if (!$job_id) {

die("No matching job found for the selected role. $job_role");

}

// File Upload Handling

$upload_dir = "../Employer/uploads/";

if (!is_dir($upload_dir)) {

mkdir($upload_dir, 0777, true);

}

function uploadFile($file, $upload_dir, $allowed_types, $max_size)

{

$filename = basename($file['name']);

$file_ext = strtolower(pathinfo($filename, PATHINFO_EXTENSION));

$new_filename = time() . '_' . $filename;

$target_path = $upload_dir . $new_filename;

if (!in_array($file_ext, $allowed_types) || $file['size'] > $max_size) {

return false;

}

return move_uploaded_file($file['tmp_name'], $target_path) ? $new_filename : false;

}
```

```
$allowed_types = ['jpg', 'jpeg', 'png', 'pdf'];

$max_file_size = 2 * 1024 * 1024; // 2MB

$photo = uploadFile($_FILES['photo'], $upload_dir, $allowed_types, $max_file_size);

$id_proof = uploadFile($_FILES['id-proof'], $upload_dir, $allowed_types, $max_file_size);

$resume = uploadFile($_FILES['resume'], $upload_dir, ['pdf', 'doc', 'docx'], $max_file_size);

if (!$photo) {

die("Invalid file type or photo size too large.");

} elseif (!$id_proof) {

die("Invalid file type or id_proof size too large.");

} elseif (!$resume) {

die("Invalid file type or resume size too large.");

}
```

## **6. TESTING AND EVALUTION**

### **6.1 UNIT TESTING**

Unit testing is conducted at the module level to verify that individual components function correctly. In this job portal, unit tests are performed on key functions such as:

- **User Registration & Login:** Ensuring users can register, log in, and log out securely.
- **Job Posting:** Verifying that employers can successfully post jobs with required details.
- Application Submission: Checking if jobseekers can apply for jobs and track their applications.
- **Database Operations:** Testing CRUD (Create, Read, Update, Delete) operations on the MySQL database.
- Each unit test ensures that every small function works correctly before moving to higher-level testing.
- \_\_\_\_\_

### **6.2 FUNCTIONALITY TESTING**

- Functionality testing ensures that all features of the job portal work as expected. This includes:
- **Job Search & Filters:** Testing location-based, skill-based, and keyword search functionalities.
- **Employer & Jobseeker Dashboards:** Verifying that each user can access and manage their profile, job postings, or applications.
- **Admin Panel:** Ensuring admins can approve/reject job postings and manage users.
- **Notifications & Updates:** Checking if real-time updates (like application status changes) work correctly.
- This type of testing ensures that the platform meets functional requirements.

## 6.3 INTEGRATION TESTING

- Integration testing ensures that different modules work together seamlessly. The focus areas include:
- User Authentication & Authorization: Ensuring users can log in and access their respective dashboards without conflicts.
- Job Posting & Application Linking: Testing if jobseekers can view job listings, apply, and employers can see the received applications.
- Database & Frontend Communication: Checking if data updates correctly when users perform actions (e.g., applying for a job updates both user and employer dashboards).
- Third-Party Integrations: If external services (like email notifications) are used, they are tested for proper integration.
- This testing phase ensures smooth interaction between different system components.

## 6.4 VERIFICATION AND VALIDATION TESTING

### 1.Verification Testing:

- Ensures the system is developed correctly according to specifications.
- Conducted through reviews, walkthroughs, and inspections of code and design.
- Example: Checking if the job search filters return accurate results.

### 2.Validation Testing:

- Ensures the system meets user requirements and works in real-world conditions.
- Conducted through user acceptance testing (UAT) and real-world scenario testing.
- Example: Confirming that a jobseeker with low digital literacy can easily search for and apply to a job.

## **7. RESULTS AND DISCUSSION**

### **Comparison of Project Results with Initial Objectives**

The Job Portal for Blue-Collar Workers successfully met its initial objectives by providing a user-friendly, accessible platform for job seekers and employers. The system was designed with a simple and mobile-responsive interface, making job searches easy for users with varying levels of digital literacy. Skill-based and location-based job search filters were implemented, allowing workers to find suitable opportunities efficiently. Employers can post job listings, manage applications, and communicate directly with candidates, streamlining the hiring process. The platform uses JavaScript for real-time updates, ensuring dynamic job search filters and application status tracking. A secure MySQL database was implemented for storing user profiles, job listings, and applications safely. Additionally, an admin panel was developed to manage users and job postings, ensuring a structured and well-maintained system. Overall, the project successfully enhances employment opportunities for blue-collar workers, providing a structured and efficient job-searching experience.

### **Key Achievements:**

- User-friendly and mobile-responsive interface for accessibility.
- Skill-based and location-based job search filters implemented.
- Employers can post jobs, manage applications, and communicate with candidates.
- Real-time updates using JavaScript for job search and application tracking.
- Secure MySQL database for safe storage of user data and job listings.
- Admin panel for user and job management to maintain platform integrity.
- Overall improvement in job opportunities for blue-collar workers.

## **8. CONCLUSION**

The **Job Portal for Blue-Collar Workers** successfully bridges the gap between skilled laborers and employers by offering an intuitive, accessible, and efficient job-matching platform. With a user-friendly interface and mobile responsiveness, the system ensures that workers with varying levels of digital literacy can navigate the portal with ease.

By leveraging HTML, CSS, JavaScript, PHP, and MySQL, the platform enables essential functionalities such as job posting, application tracking, real-time updates, and employer-worker communication. The incorporation of location-based recommendations and skill-based filtering further enhances job search efficiency, ensuring that workers find opportunities suited to their expertise.

Additionally, the admin panel provides seamless management of users, job listings, and applications, making the system scalable and maintainable. Overall, this project significantly improves the job search experience for blue-collar workers, empowering them with better employment opportunities while streamlining recruitment for employers. By addressing the employment challenges in manual labor industries, the portal contributes to a more inclusive and efficient job market, fostering economic growth and workforce empowerment.

## 9. FUTURE ENHANCEMENT

- **Mobile Application Development** – Create a dedicated mobile app for Android and iOS to improve accessibility for workers with limited access to computers.
- **Multilingual Support** – Integrate multiple language options to help workers from diverse backgrounds use the platform easily.
- **AI-Powered Job Matching** – Implement AI algorithms to recommend jobs based on skills, location, and previous job applications.
- **Skill Verification & Certification** – Partner with training institutes to provide skill validation and certification for workers.
- **Video Resume & Profile Enhancement** – Allow workers to upload video resumes showcasing their skills and experience.
- **Real-Time Job Alerts & Notifications** – Enable SMS and push notifications for job updates and employer responses.
- **E-Wallet & Digital Payments** – Introduce secure payment gateways for salary transactions, ensuring timely and transparent payments.
- **Employer Reviews & Ratings** – Allow workers to rate and review employers, improving trust and transparency.
- **Freelance & Contract-Based Work Support** – Expand the platform to include short-term and freelance job opportunities.

## **10. REFERENCES**

### **1. Web Development References**

- MDN Web Docs – <https://developer.mozilla.org/en-US/curriculum/core/web-standards/> (For HTML, CSS, JavaScript)
- W3Schools – <https://www.w3schools.com/php/default.asp> (For front-end and PHP development)
- PHP Official Documentation – <https://www.php.net/manual/en/>
- MySQL Documentation – <https://dev.mysql.com/doc/>
- Email Js - <https://wsww.emailjs.com/>
- 2Factor SMS Gateway – <https://2factor.in>

### **2. Web-Based Job Portals & Platforms:**

- Indeed: <https://in.indeed.com/?r=us>
- LinkedIn Jobs: <https://www.linkedin.com/jobs>
- Apna (Job Platform for Blue-Collar Workers): <https://apna.co>
- WorkIndia: <https://www.workindia.in>

### **3. Technical Documentation & Guides:**

- W3Schools. (2023). *PHP & MySQL Web Development*. Retrieved from [https://www.w3schools.com/php/php\\_mysql\\_intro.asp](https://www.w3schools.com/php/php_mysql_intro.asp)
- Mozilla Developer Network (MDN). *JavaScript and Web Technologies Documentation*. Retrieved from <https://developer.mozilla.org/en-US/curriculum/core/javascript-fundamentals/>
- Bootstrap Official Documentation for Responsive UI Design: <https://getbootstrap.com/docs/5.3/getting-started/contribute/#tooling-setup>