SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

A PROJECT REPORT ON

"LABOUR PROBLEM FACE BY FARMER AND ITS SOLUTION"

SUBMITTED TOWARDS THE

PARTIAL FULFILLMENT OF THE REQUIREMENTS

OF CIA of SY BACHELOR OF TECHNOLOGY IN

COMPUTER ENGINEERING

BY

Mr. Manoj Rambhau Jadhav	[UCS23M1050]
Mr. Aditya Narayan Jadhav	[UCS23M1048]
Mr. Abhay Suresh Dyavane	[UCS23M1036]

(SY A.Tech Computer Engineering)

UNDER THE GUIDANCE OF

Dr. A. B. Pawar

DEPARTMENT OF COMPUTER ENGINEERING

SANJIVANI RURAL EDUCATION SOCIETY'S

SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON

(An Autonomous Institute)

SANJIVANI RURAL EDUCATION SOCIETY'S

SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON

(AN AUTONOMOUS INSTITUTE)

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

THIS IS TO CERTIFY THAT THE PROJECT ENTITLED

"LABOUR PROBLEM FACE BY FARMER AND ITS SOLUTION"

SUBMITTED BY

Mr. Manoj Rambhau Jadhav	[UCS23M1050]
Mr. Aditya Narayan Jadhav	[UCS23M1048]
Mr. Abhay Suresh Dyavane	[UCS23M1036]

IS A BONAFIDE WORK CARRIED OUT BY STUDENTS UNDER THE SUPERVISION OF DR. A.B.PAWAR AND IT IS SUBMITTED TOWARDS THE PARTIAL FULFILLMENT OF THE REQUIREMENT OF CIA OF SY BACHELOR OF TECHNOLOGY (COMPUTER ENGINEERING). DURING THE ACADEMIC YEAR 2024-25

DR. A. B. PAWAR

[SUBJECT TEACHER]

DR. D. B. KSHIRSAGAR

[H.O.D]

DR. A.G. THAKUR

[DIRECTOR]

CIA APPROVAL SHEET

A CIA Project Report On

“LABOUR PROBLEM FACE BY FARMER AND ITS SOLUTION”

is successfully completed by

Mr. Manoj Rambhau Jadhav [UCS23M1050]

Mr. Aditya Narayan Jadhav [UCS23M1048]

Mr. Abhay Suresh Dyavane [UCS23M1036]

DEPARTMENT OF COMPUTER ENGINEERING

SANJIVANI RURAL EDUCATION SOCIETY’S

SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON

(An Autonomous Institute)

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

ACADEMIC YEAR 2024-25

Dr. A. B. Pawar

[Subject Teacher]

Dr. D. B. Kshirsagar	**Dr. A.G. Thakur**
[H.O.D]	**[Director]**

Sanjivani College of Engineering,

Kopargaon

Project Proposal

“LABOUR PROBLEM FACE BY FARMER AND ITS SOLUTION”

Submitted by

Students Information

Roll.No.	Name	Email ID	Mobile Number	Sign
50	Manoj Jadhav	heymanojjadhav@gmail.com	9881550575	
48	Aditya Jadhav	adijadhav7020@gmail.com	7020615650	
36	Abhay Dyavane	abhaydyavane @gmail.com	9421527059	

Department of Computer Engineering

Project Proposal

Contents

1. PROJECT PROPOSAL

PROJECT SUMMARY:

Farmers face significant labor problems, including shortages due to the seasonal nature of work, migration to urban areas, an aging workforce, and low wages. There's also a lack of skilled labor needed to operate modern farming technologies. Poor working conditions, lack of job security, and exploitation are additional challenges.

To address these issues, solutions include:

1. Mechanization and Automation: Using machinery and advanced technologies to reduce reliance on manual labor.

2. Improved Labor Policies: Ensuring fair wages, better working conditions, and stronger labor rights enforcement.

3. Training and Education: Providing skill development programs and engaging youth in agriculture.

4. Attracting Labor: Offering incentives, improving rural living conditions, and promoting rural development.

5. Support Systems: Government aid, cooperatives, and labor unions to strengthen the agricultural workforce.

These approaches aim to make farming more efficient, sustainable, and appealing to workers.

1.1 Background

Labor issues have been a significant challenge in the agricultural sector for many years, affecting both small and large-scale farmers. These problems are multi-faceted and can severely impact productivity, profitability, and the sustainability of farming operations.

Lack of Skilled LaborModern Farming Techniques: The adoption of advanced technologies and modern farming techniques requires skilled labor, which is often in short supply.Education and Training: There is a gap in the education and training of rural workers, making it difficult for them to operate modern machinery and implement new farming methods effectively.

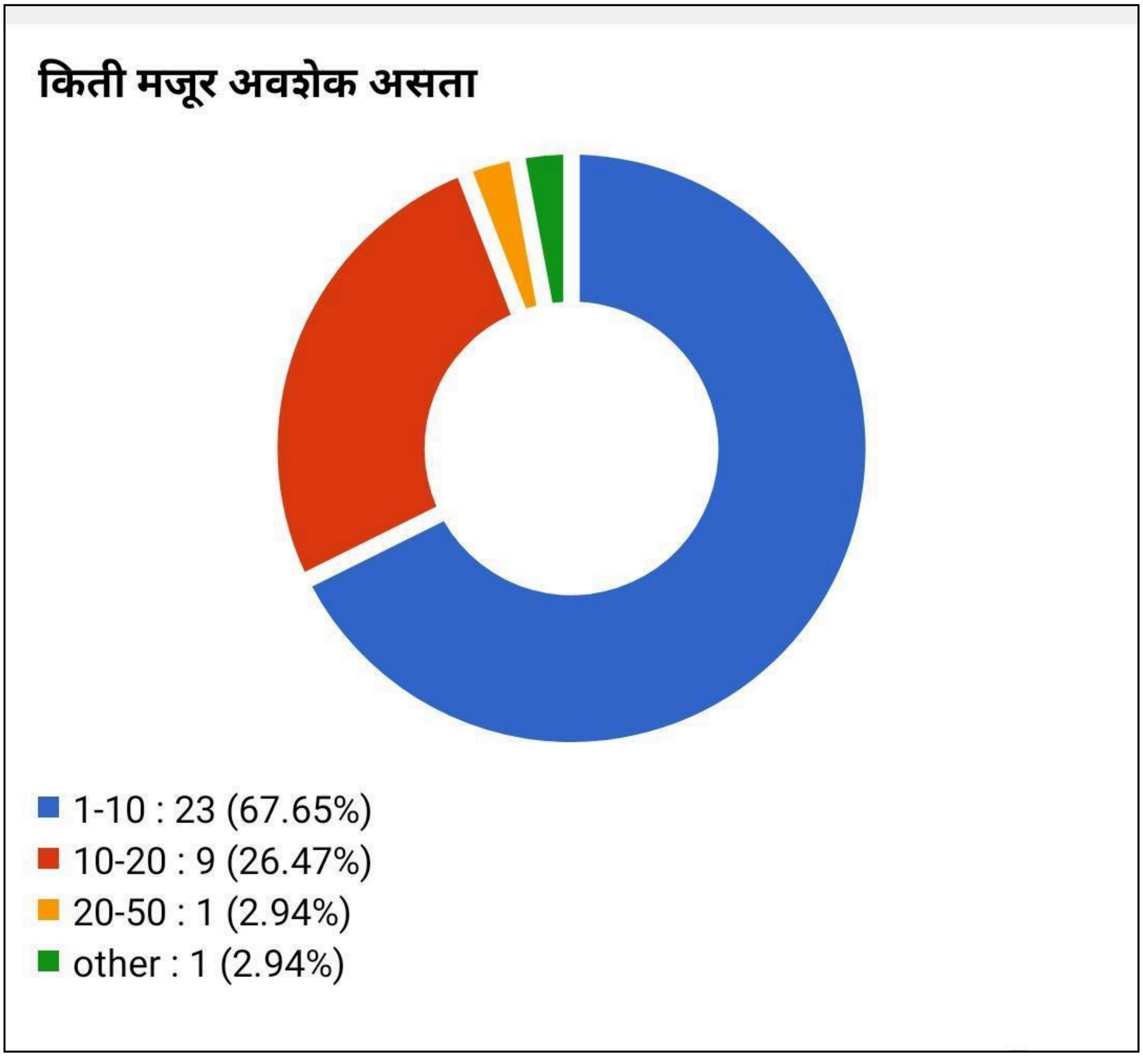


Fig.1.1.1 Collected farmer Feedback Detail

1.2 Statement of the problem

The original problem statements: “LABOUR PROBLEM FACE BY FARMER AND ITS SOLUTION”

GPS Map Camera
Kopargaon, Maharashtra, India
Swami rang Appartment, Subhadranagar, Kopargaon, Maharashtra 423601, India
Lat 19.89311°
Long 74.480813°
14/08/24 02:08 PM GMT +05:30
Google

Fig.1.2.1 Collected Feedback Detail

Importance of problem :

The issue of labor shortages in agriculture is critical for several reasons:

1.Decreased Agricultural Productivity: A shortage of farm laborers leads to reduced agricultural productivity, as there are not enough workers to plant, maintain, and harvest crops efficiently. This can result in lower yields and financial losses for farmers.

2. Economic Impact: Agriculture is a significant part of many economies, especially in rural areas. Labor shortages can lead to increased production costs, higher food prices, and reduced export potential, which can negatively impact the broader economy.

3. Rising Labor Costs: With fewer people willing to work in agriculture, the cost of hiring labor increases, making farming less profitable and accessible, particularly for small-scale farmers.

Addressing the labor shortage in agriculture is crucial to ensuring the sustainability of food production, the health of rural economies, and the overall well-being of the population.

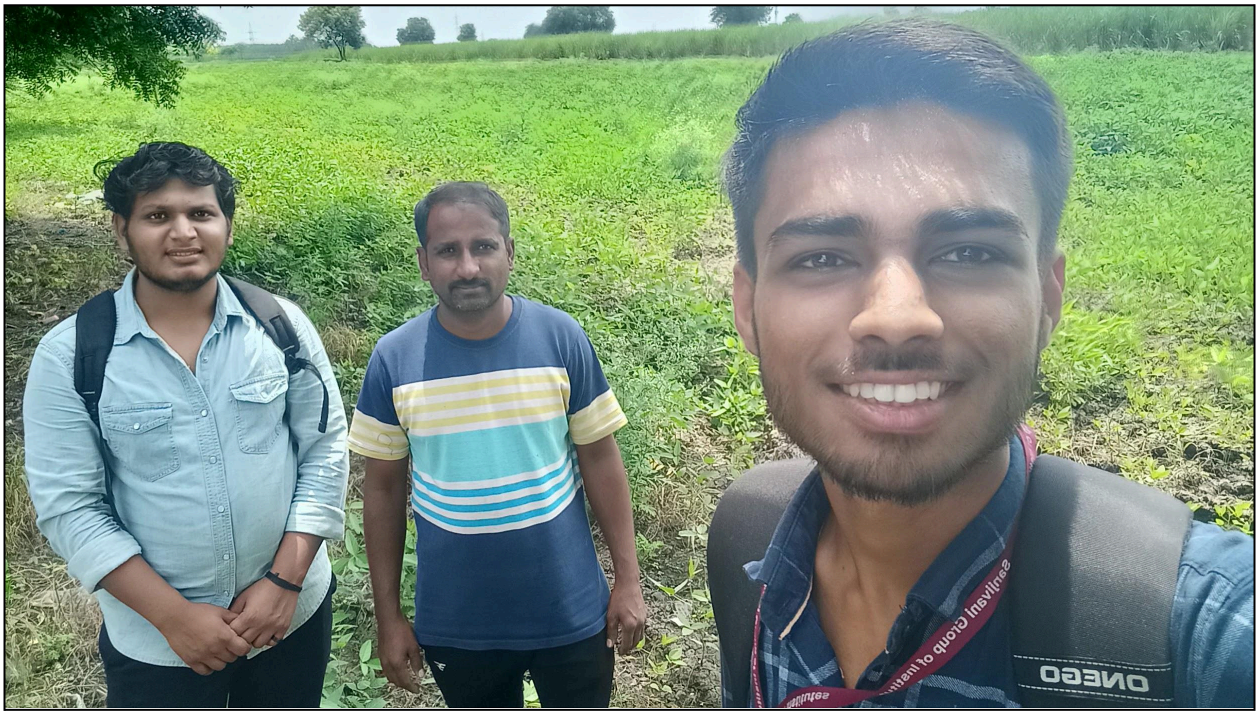


Fig.1.2.2 visiting farmer.

1.3 Objectives

1.Hiring the labor for farming through easy and simple website.

2.save the time of farmer.

3.labour get easy job.

1.4 Technology/ Research Gap

farmer labour problem can be solve by

website for labour worker and vacancy in farmer work.

The website for a farmer and labour consists of several key components:

- Frontend Interface
- Backend Server
- Database

- **1..Frontend Interface:** The frontend interface is the part of the website that users interact with. It is built using HTML, CSS, and JavaScript This interface displays information about labour avalibility, schedules, and prices, and allows users to search for labour, view schedules, and check prices.
- **2.Backend Server**: The backend server processes requests from the frontend, manages data, and performs logic operations. It is typically built using technologies such as Node.js, Python (Django/Flask), or Ruby on Rails. The backend handles functionalities like costing and avalibility information, calculating prices, and managing user queries.
- **3.Database:** The database stores all the necessary data for the website, including availability and, schedules, prices, and user information. Common database systems used 8 include MySQL, PostgreSQL, or MongoDB. The database ensures that data is organised and quickly accessible

Website Features:

- **Frontend Development:** Uses modern web technologies to provide a responsive and user-friendly interface. JavaScript frameworks like

React or Angular may be employed to handle dynamic content and user interactions.

● **Backend Development:** Implemented with server-side technologies like python

● **Database Management:** Designed to store and efficiently retrieve data related to availability of labour, schedules, and prices. Database systems such as MySQL or PostgreSQL are used for this purpose. Expected Outcomes: The farmer website is intended to provide a streamlined and informative experience for users, making it easy to work, check schedules, and understand pricing. It should enhance user convenience and accessibility.

1.5 Resources and Budget

1. Front End Development Tools:

1.1 HTML, CSS, JavaScript: For building and styling the website's interface.

1.2 Frontend Frameworks/Libraries (optional): Tools like React, Vue.js, or Angular to streamline development.

1.3 Responsive Design Tools: Frameworks such as Bootstrap or Material UI to ensure the website works on mobile, tablet, and desktop devices.

2. Backend Development:

2.1 Server and Database Management: Tools such as Node.js, Django, or Ruby on Rails for the backend logic. Databases like MySQL for storing availability and price data.

2.2 API Development: If integrating with any external systems.

3. Web Hosting & Domain Registration:

3.1 Web Hosting Service: A reliable platform to host the website (e.g., AWS, DigitalOcean, or Bluehost).

3.2 Domain Name: A domain name for the website.

4. Content Management System (CMS) (Optional): A CMS like WordPress, or a custom-built CMS, allowing easy updates to availability of labour, schedules, and rent by non-technical staff.

5. Design & Branding:

5.1 Graphics: Icons, banners, logos, and other visual elements to make the site more user-friendly and visually appealing.

5.2 UI/UX Design Software: Tools such as Adobe XD, Figma, or Sketch for creating the website layout.

6.Testing Tools & Equipment:

6.1 Quality Assurance (QA) Tools: Automated testing tools like Selenium, and manual testing devices for ensuring the site functions properly across browsers anddevices.

6.2 Performance Testing Tools: Tools to assess website load times and performance under different user conditions.

7. Internet Connectivity:

7.1 Stable Internet Connection: Required during the development and testing phases to upload code and access remote hosting servers.

8. Security Measures:

8.1 SSL Certificate: To secure data transfer between the user and the website.

8.2 Firewall/Anti-malware Software: To protect the website from cyber threats

9. Installation and Setup:

9.1 Hosting Setup: Time and resources for setting up the hosting, domain, and databases.

9.2 Website Configuration: Ensuring the correct deployment of the site and testing it in a live environment.

10. Maintenance & Updates:

10.1 Regular Updates: Ongoing development to update the content, fix bugs, and add new features.

10.2 Technical Support: A team or person to handle any issues or downtimes after the website is launched.

1.6 Project Plan with Milestones

Sr.No.	Milestone	Activity	Duration in months
1	Planning Phase	Farmer labour problem	5 to 10 days
2	Design Phase	design web page	22 day
3	Development & Deployment Phase	Showing website to farmers anf labours	2 weeks
4	Maintenance & Support Phase	updates and improvements made as needed.	2 weeks

1.7 Project Plan with Milestones

SR. NO	Milestone	Activity	Duration
01	Requirements Gathering	Collect detailed requirements from community members through survey forms and interviews.	Week 2-3
02	Data Analysis	Analyse the gathered data and find appropriate solutions.	Week 4
03	Website Design	Selection of a particular development model .	Week 5
04	Content Creation	Gather content (text, images, videos) to be inserted into the website.	Week 6
05	Development	Build the website based on approved designs and requirements.	Week 7-9
06	Testing	Conduct thorough testing (functional, usability, performance, and security testing) to ensure the website is bug-free.	Week 10-11
07	Deployment	Deploy the website to the live server and ensure it's accessible to users.	Week 12
08	Post-Launch Review	Monitor the website for issues and gather feedback for improvements.	Week 13-14
09	Project Closure	Complete project documentation, handover to the maintenance team, and close the project.	Week 15-16

SKILL COMPETENCY:

Sr. No	PRN	NAME	Required Skill Sets						
			Documentation	**Presentation**	**Coding**	**HTML**	**CSS**	**SQL**	**JavaScript**
01	UCS23M1050	Manoj Rambhau Jadhav	4	3	4	4	4	2	2
02	UCS23M1048	Aditya Narayn Jadhav	4	**3**	3	4	3	**2**	3
03	UCS23M1036	Abhay Suresh Dhayane							

TOOLS REQUIRED:

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

2. CSS
    - **Selectors and specificity:** CSS selectors apply styles to HTML components based on IDs, classes, and elements, with specificity determining precedence.
    - **Cascading:** 'Cascading' in CSS means later, more specific styles override earlier ones, ensuring a consistent and adaptable design across a website.
    - **Box Model:** The CSS box model defines how HTML elements are arranged using content, padding, border, and margin, which is crucial for layout and positioning.
    - **Typography:**CSS allows developers to manage element colours, including text, background, and borders, with additional features for gradients, background images, and transparency using RGBA and HSLA.
    - **Colours and background:**CSS properties control layout and positioning, including dimensions and content flow, helping create responsive websites that adapt to all devices.
    - **Layout and positioning:**CSS provides tools for interactivity and animation, such as transitions and animations, enhancing user experience and engagement with dynamic elements.

3.SQL

MySQL:

- **Data Storage**: Efficiently stores large amounts of data.
- **Data Retrieval**: Allows quick and easy access to data.
- **Data Manipulation**: Supports operations like inserting, updating, and deleting data.
- **Data Security**: Offers robust security features to protect data.
- **Scalability**: Can handle small to large applications with ease

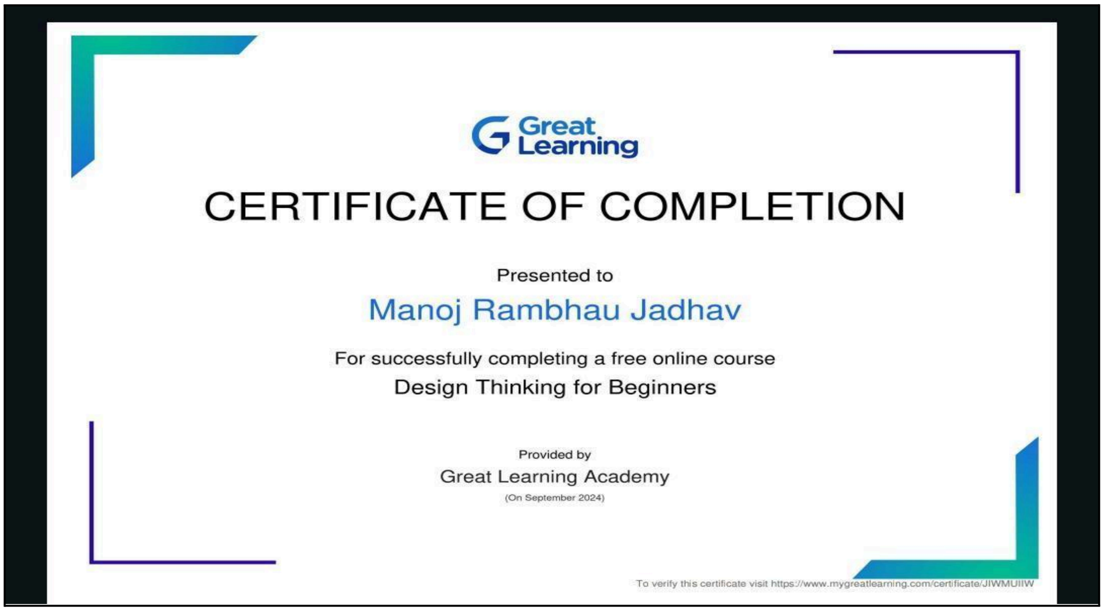
4.Javascript

- **Dynamic Typing:** JavaScript doesn't require variable types to be explicitly declared. Types are determined at runtime.
- **Event-Driven:** JavaScript can respond to user actions (like clicks and keystrokes) and other events asynchronously.
- **Object-Oriented:** JavaScript uses prototypes for inheritance rather than classes, though ES6 introduced class syntax for a more traditional OOP approach.
- **Functional Programming:** Functions are first-class objects, meaning they can be passed as arguments, returned from other functions, and assigned to variables.
- **Asynchronous Operations:** JavaScript supports asynchronous operations with callbacks, promises, and async/await syntax**.**
- **DOM Manipulation:** JavaScript can interact with and modify the HTML and CSS of a web page through the Document Object Model (DOM).
- **Built-in Objects:** JavaScript includes a variety of built-in objects

like

- `Math`, `Date`, `Array`, and `JSON` for common programming tasks.
- **Closures:** JavaScript functions can capture and remember the lexical environment in which they were defined, enabling powerful patterns like data encapsulation.

CERTIFICATIONS ON DESIGN THINKING:



Great Learning

CERTIFICATE OF COMPLETION

Presented to

Manoj Rambhau Jadhav

For successfully completing a free online course

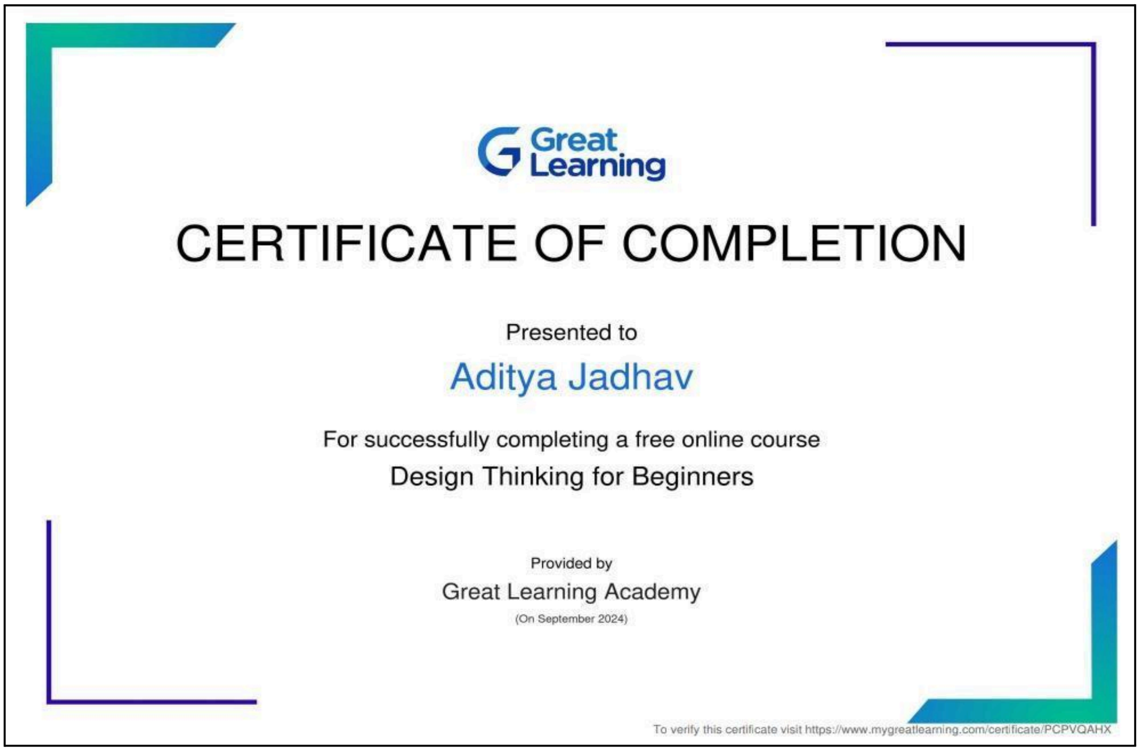
Design Thinking for Beginners

Provided by

Great Learning Academy

(On September 2024)

To verify this certificate visit https://www.mygreatlearning.com/certificate/JIWMUIIW



Great Learning

CERTIFICATE OF COMPLETION

Presented to

Aditya Jadhav

For successfully completing a free online course

Design Thinking for Beginners

Provided by

Great Learning Academy

(On September 2024)

To verify this certificate visit https://www.mygreatlearning.com/certificate/PCPVQAHX

PROJECT MODEL USED:

AGILE MODEL: So that changes can be made in the middle of software development and the software project can be completed quickly.

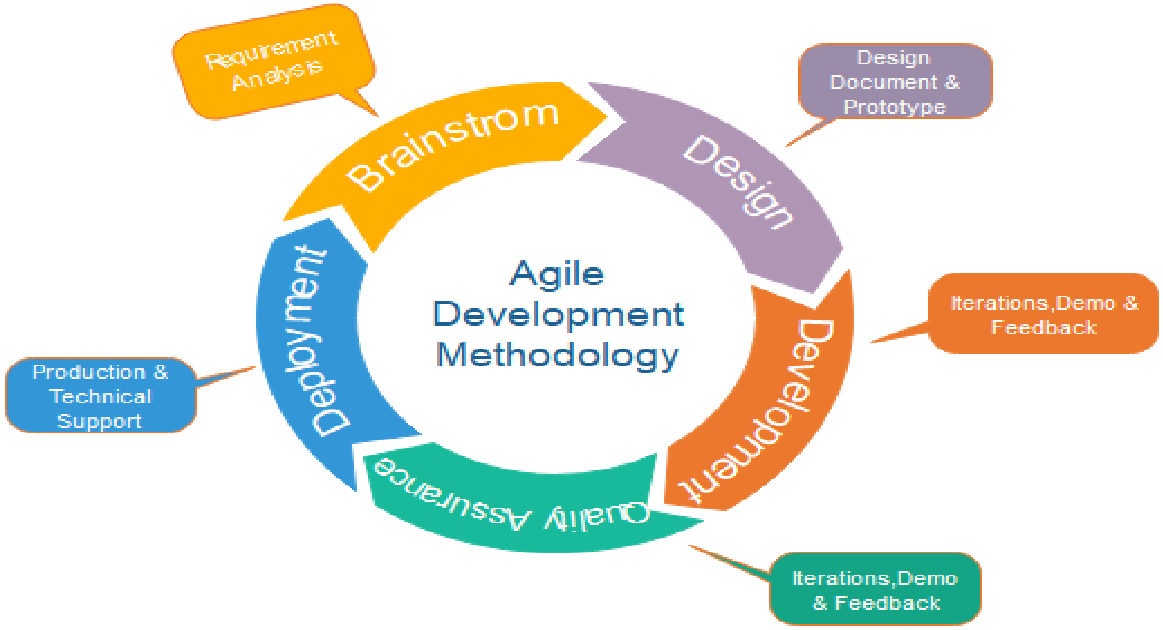

Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Iterations,Demo & Feedback
Agile Development Methodology
Production & Technical Support
Deployment
Development
Quality Assurance
Iterations,Demo & Feedback

Fig. Agile Model

- **Steps in the Agile Model**
  - ➢ Requirement gathering
  - ➢ Design the Requirements
  - ➢ Construction / Iteration
  - ➢ Testing / Quality Assurance
  - ➢ Deployment
  - ➢ Feedback

- **Advantages of the Agile Model**
  1) It reduces the total development time of the whole project.
  2) Emphasises face to face communication among team members

- **Disadvantages of the Agile Model**

  1.It is not suitable for handling complex dependencies.

- **Flowchart Breakdown:**

1. Home Page:

- Users arrive and select their role: Farmer or Labour.

2. User Registration / Login:

- Farmer: Sign up or log in to post jobs and view laborer profiles.
- Labor: Sign up or log in to update availability and browse jobs.

3. Farmer Workflow:

- Create Profile: Farmers fill out their basic information, location, and type of labour they need.
- Post Job: Farmers post job details (location, type of labor, date needed).
- Search Labour: Farmers can search for available laborers using filters (location, skills, availability).
- Contact Laborer: Farmers view laborer profiles and contact them directly (via messaging or phone).
- Rate Laborer: After work is completed, farmers can rate laborers based on performance.

4. Labour Workflow:

- Create Profile: Labour fill out their details, skills, and contact info.
- Update Availability: Labour update their availability status regularly.
- Browse Jobs: Laborers can browse job listings based on location, type of work, and dates.
- Apply for Jobs: Laborers can apply for jobs they are interested in.
- Rate Farmer: After completing a job, laborers rate farmers.

5. Admin Workflow (optional):

- Monitor Platform: Admins can monitor activity, approve new users, manage disputes, and handle complaints.
- Content Management: Admins can update content (e.g., FAQs, terms, etc.) and make updates to the site.

- **Assumptions**

1. Internet Access:

   - Farmers and laborers in rural areas have access to the internet, even if at a basic level (e.g., mobile data).

   - Internet speed might be low in some areas, so the website should be optimized for slower connections (lightweight, fast loading).

2. Device Usage:

   - A majority of users (both farmers and laborers) will access the site via mobile devices rather than desktops.

   - The design should prioritize mobile-first development, ensuring a responsive and user-friendly experience on smartphones and tablets.

3. Language and Literacy Levels:

   - Some farmers and laborers may have limited literacy or prefer local languages.

   - The website should have simple, intuitive design elements, with minimal text where possible. Local language options and visual cues (icons, buttons) can enhance usability.

4. User Profiles:

   - Both farmers and laborers will need to create profiles on the platform to post jobs or indicate availability.

   - Users may not be tech-savvy, so the registration process must be simple, with guidance on how to use the site.

5. Labor Availability Data:

   - Laborers will regularly update their availability status, and this data needs to be stored and made searchable by farmers.

   - Farmers will require easy filtering options based on location, type of labor, and availability dates.

6. Communication:

- After finding matching laborers, farmers will want a way to contact them directly through the platform (e.g., chat or phone number sharing).

- Laborers will expect notifications (SMS or email) when a farmer is interested in hiring them.

7. Geographic and Regional Considerations:

- The system should allow users to filter laborers by geographic proximity (e.g., villages, towns) to ensure farmers find nearby labor.

- The platform should also consider labor demand variations by season (e.g., planting, harvesting) and enable seasonal job posting.

8. Labor Categories:

- Different types of labor (e.g., plowing, harvesting, irrigation) will need to be clearly categorized, with laborers able to specify their skills and experience.

9. Trust and Security:

- Users will need confidence that their data (especially contact details) is secure. Authentication processes must ensure that both farmers and laborers are verified.

- Farmers and laborers should be able to rate each other to build trust on the platform.

- **Declarations**

1. Purpose and Vision:

- The website's primary goal is to connect farmers with laborers quickly and efficiently, reducing the time it takes to find and hire suitable help for agricultural tasks.

2. Target Users:

- The website is designed specifically for two user groups: farmers seeking labor for their fields and laborers seeking agricultural work.
- Farmers may range from small-scale to larger commercial farms. Laborers might include both full-time workers and those looking for seasonal work.

3. Key Features:

- Labor Search Functionality: A search engine that allows farmers to filter laborers based on availability, location, type of work, and experience.
- Job Posting and Notifications: Farmers can post job listings, and laborers can update their availability. The platform should notify laborers of relevant job postings.
- Profile Creation: Both parties create profiles, allowing farmers to review laborers' past experiences and skills, and laborers to see farmers’ reviews.
- Communication System: A method for direct communication (messaging or contact sharing) between farmers and laborers.
- Rating System: Farmers and laborers will be able to rate each other after completing jobs to build trust and reliability.

4. Technology and Architecture:

- Mobile-First Development: The website will be developed with a mobile-first approach, ensuring it is fully functional on smartphones.
- Fast Loading Time: The site will be optimized for performance, focusing on fast load times, even on slow internet connections.
- Database for Job Listings: A backend system will manage profiles, job postings, and laborer availability.
- Security Measures: Security features (such as SSL, secure login, data encryption) will be in place to protect user data.

5. Localization:

- The platform will support multiple languages relevant to the region where the farmers and laborers operate, ensuring accessibility for all users.
- GPS integration or location-based filtering will help farmers find local labor quickly.

6. Monetization (optional):

- Potential monetization strategies could include charging farmers a small fee for job postings or offering premium services, though the basic functionality should remain free to encourage adoption.

7. Scalability:

- The website will be built to scale, allowing for future growth in user numbers and potential expansion into other regions or industries beyond agriculture.

RISK ASSESSMENT:

1. **Technical Risks:**

   - Difficulty in integrating labor availability data.
   - Bugs or delays in feature development.
   - Poor scalability of the website as user base grows.

2. **Operational Risks**

   - Delays in gathering or maintaining up-to-date labor information.
   - Insufficient testing before deployment.
   - Inconsistent performance across cloud platforms.

3. **Security Risks:**

   - Data breaches of farmer and laborer information.
   - Insecure APIs or third-party services.
   - Lack of proper encryption for sensitive data.

4. **Market/Usage Risks:**

   - Low adoption by farmers or laborers.
   - Difficulty in onboarding users without digital literacy.
   - Regional internet access issues affecting usage.

5. **Financial Risks**:

- Unforeseen hosting or platform costs.
- Potential loss of funding or revenue streams.
- High maintenance costs post-launch.

IMPLEMENTATION DETAILS:

7.1 Implementation Platform Details

Device name	MrJadhav
Processor	12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz
Installed RAM	16.0 GB (15.7 GB usable)
Device ID	7615595A-FFA3-4775-A1E1-CB51F11D9436
Product ID	00342-42635-57167-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

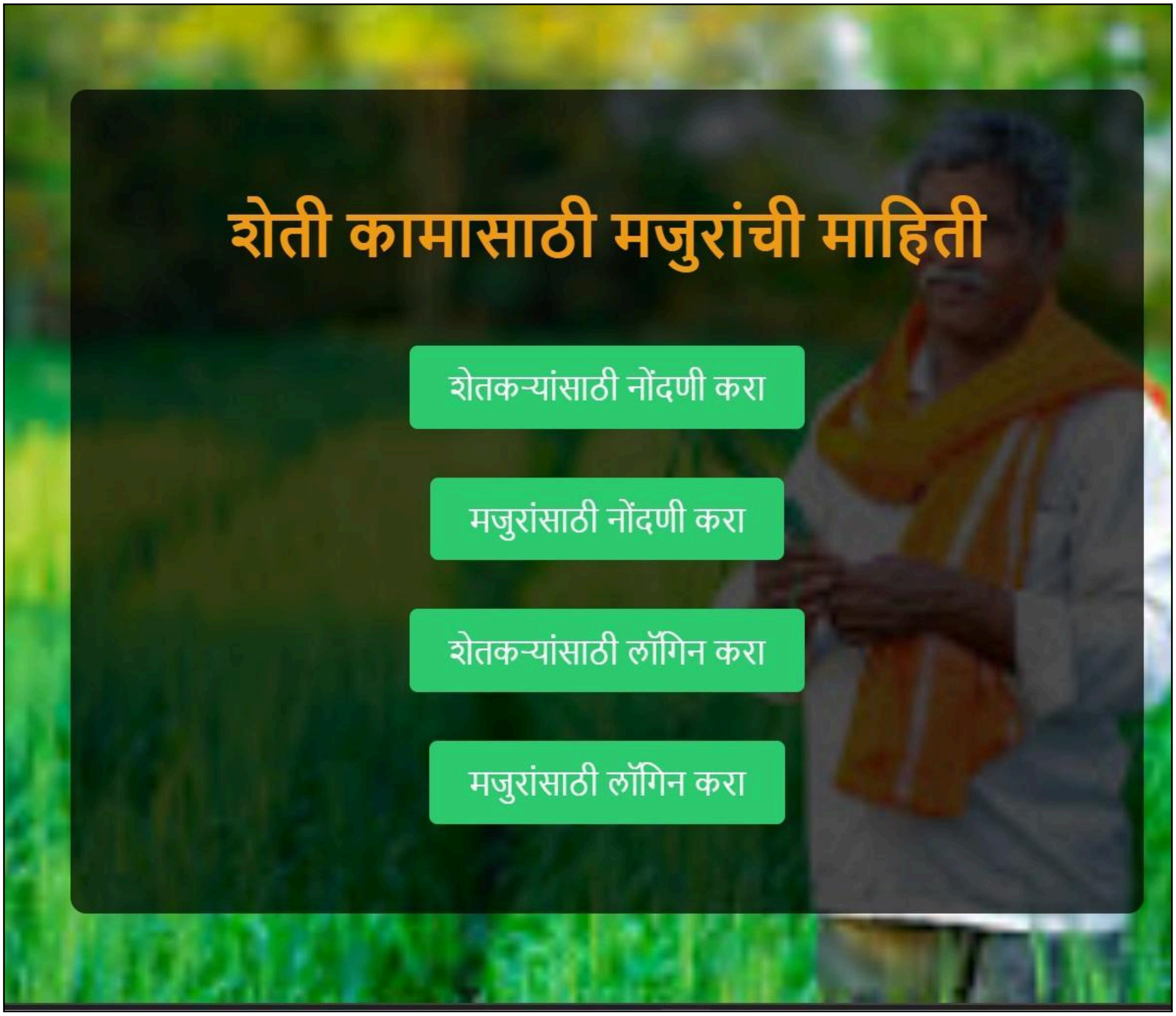
7.2 Modulus for the system:

System Modules

1. **User Registration and Authentication Module**
   - **Purpose:** Allows farmers and laborers to create accounts and securely log in to access personalized features.
   - **Features:** Registration form, login page, password reset, and account management.
2. **Labor Availability Module**
   - **Purpose:** Enables farmers to check the availability of labor in their area.
   - **Features:** A search function based on location, real-time data display of available labor, and filters for specific labor skills.
3. **Labor Profile Management Module**
   - **Purpose:** Allows laborers to manage their profiles, update availability, and specify skills.
   - **Features:** Profile creation, skill listing, availability toggle, and location update.

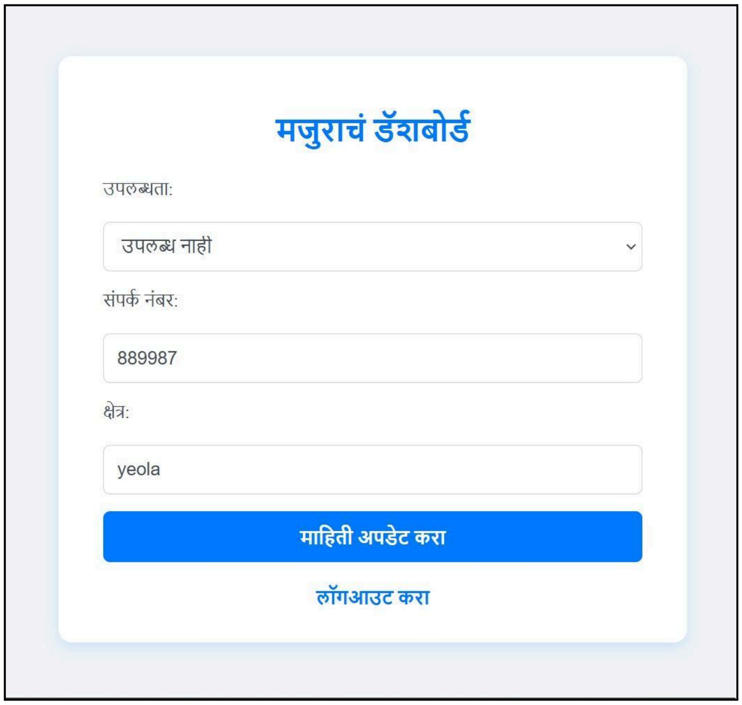
4. **Notification and Communication Module**
    - **Purpose:** Facilitates communication between farmers and laborers.
    - **Features:** In-app notifications, messaging system, and alert generation when there is a nearby labor match.
5. **Admin Management Module**
    - **Purpose:** Provides an admin interface to monitor and manage users and system activities.
    - **Features:** Dashboard view, user management, system analytics, and report generation.

PAGES OF WEBSITE:



शेती कामासाठी मजुरांची माहिती

2



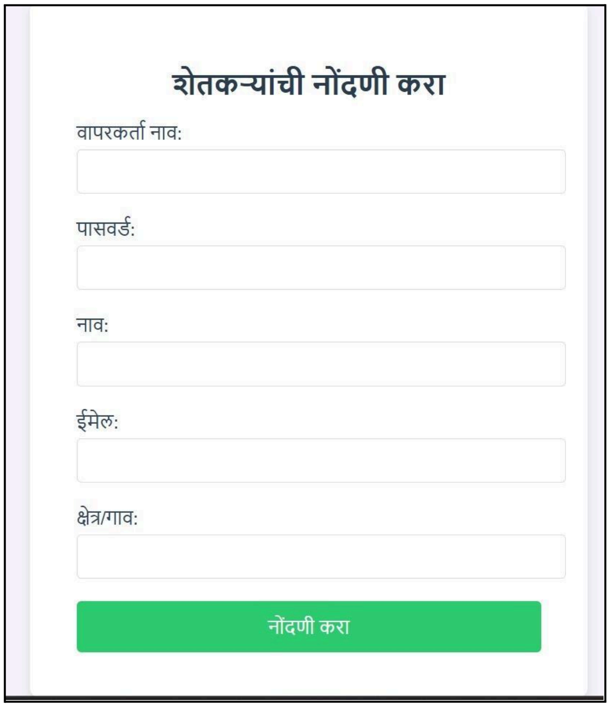
मजुराचं डॅशबोर्ड

उपलब्धता: उपलब्ध नाही

संपर्क नंबर: 889987

क्षेत्र: yeola

3



शेतकऱ्यांची नोंदणी करा

वापरकर्ता नाव:

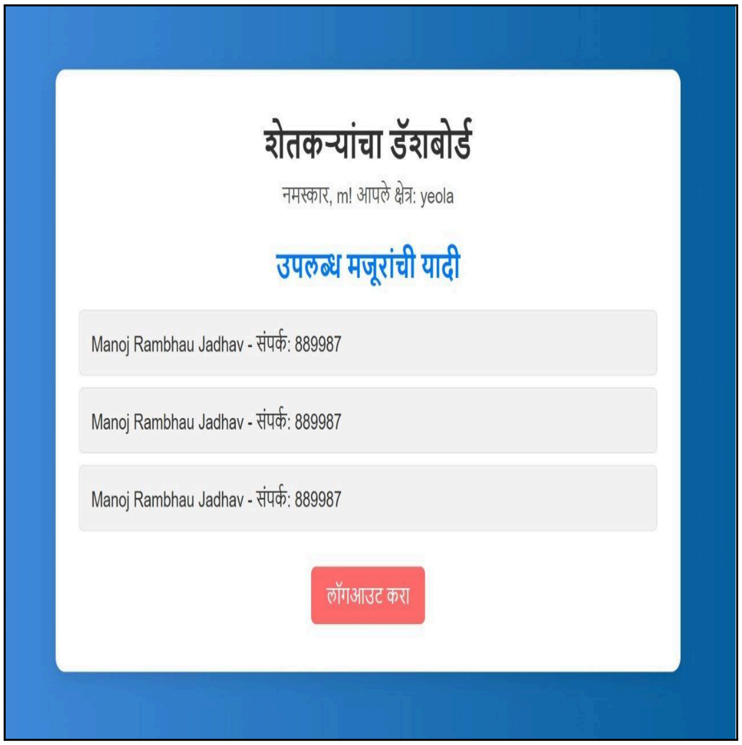
पासवर्ड:

नाव:

ईमेल:

क्षेत्र/गाव:

4



शेतकऱ्यांचा डॅशबोर्ड

नमस्कार, m! आपले क्षेत्र: yeola

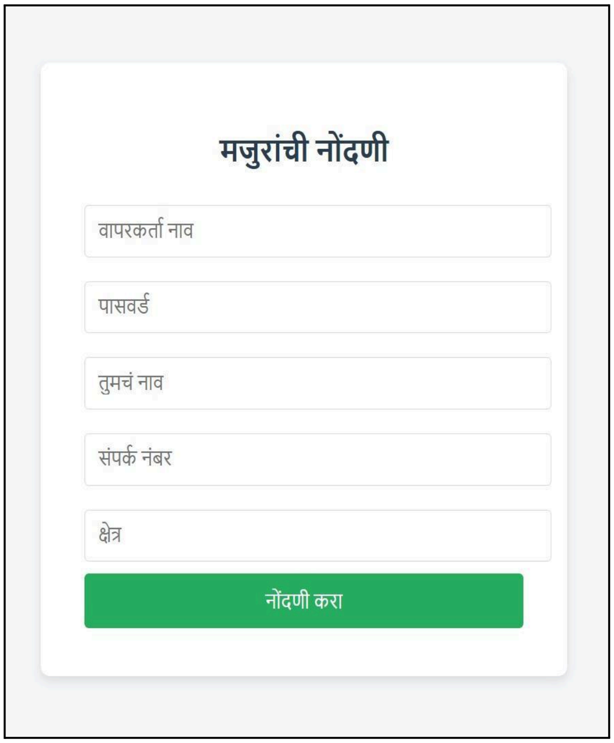
उपलब्ध मजूरांची यादी

Manoj Rambhau Jadhav - संपर्क: 889987

Manoj Rambhau Jadhav - संपर्क: 889987

Manoj Rambhau Jadhav - संपर्क: 889987

5



मजुरांची नोंदणी

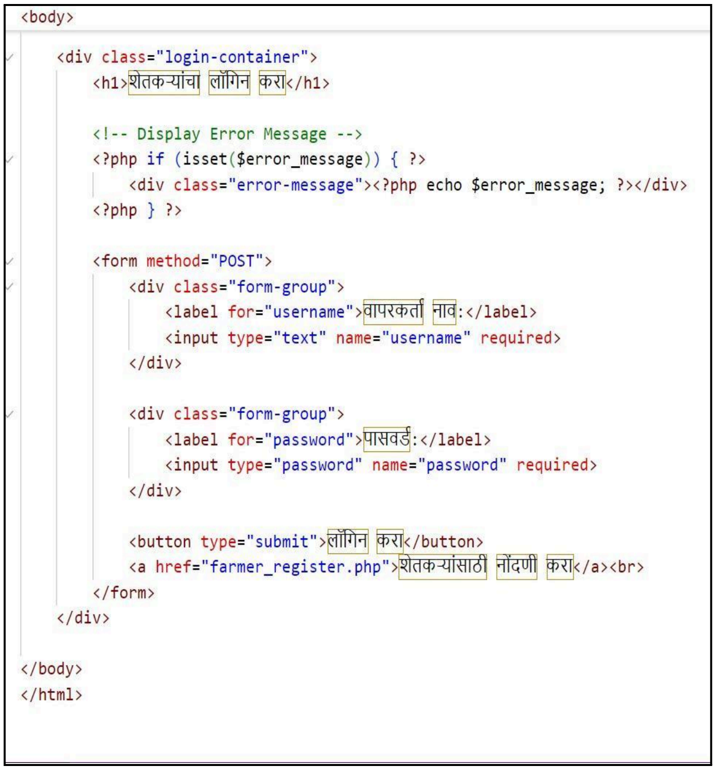
वापरकर्ता नाव

पासवर्ड

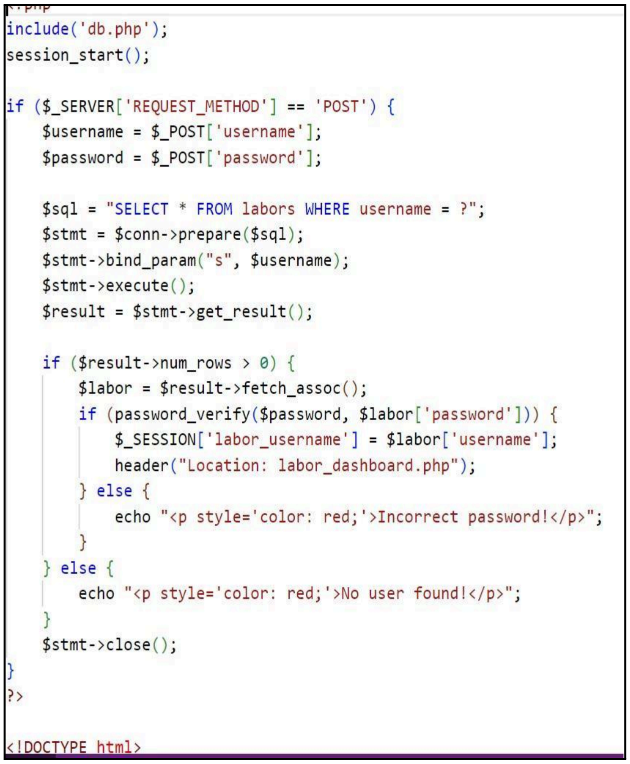
तुमचं नाव

संपर्क नंबर

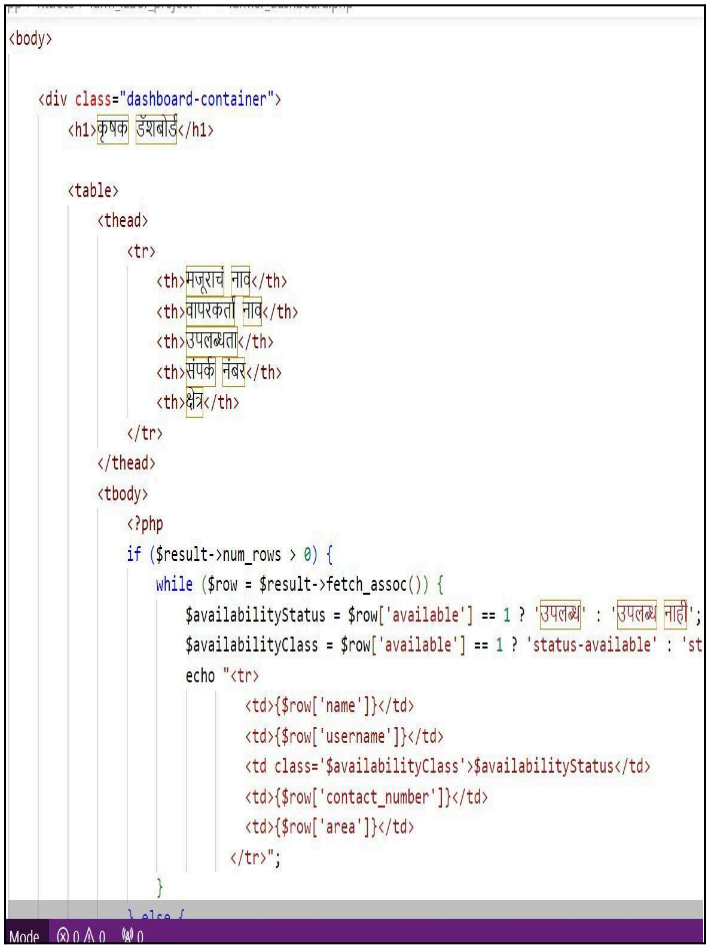
क्षेत्र



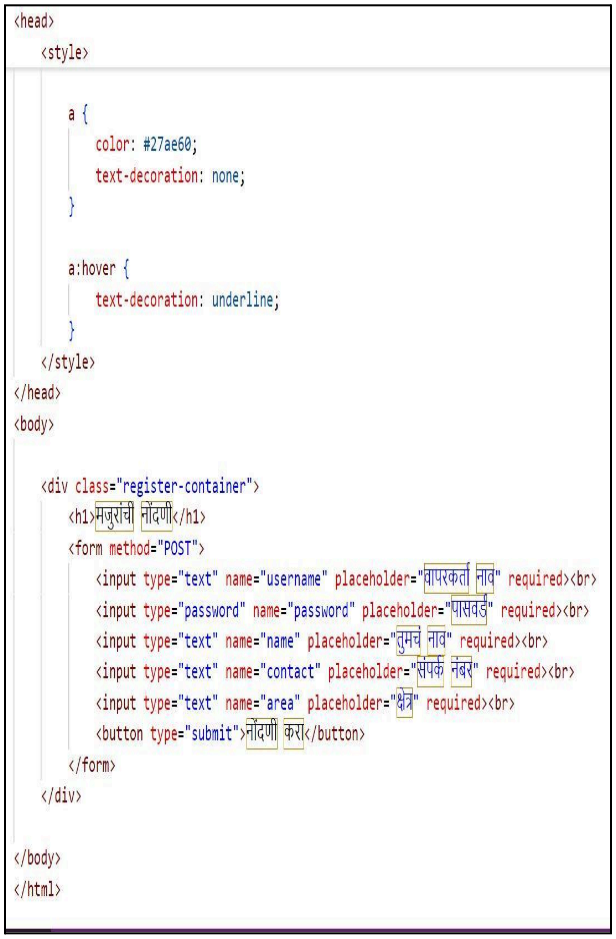
```
<body>

    <div class="login-container">
        <h1>शेतकऱ्यांचा लॉगिन करा</h1>

        <!-- Display Error Message -->
        <?php if (isset($error_message)) { ?>
            <div class="error-message"><?php echo $error_message; ?></div>
        <?php } ?>

        <form method="POST">
            <div class="form-group">
                <label for="username">वापरकर्ता नाव:</label>
                <input type="text" name="username" required>
            </div>

            <div class="form-group">
                <label for="password">पासवर्ड:</label>
                <input type="password" name="password" required>
            </div>

            <button type="submit">लॉगिन करा</button>
            <a href="farmer_register.php">शेतकऱ्यांसाठी नोंदणी करा</a><br>
        </form>
    </div>

</body>
</html>
```



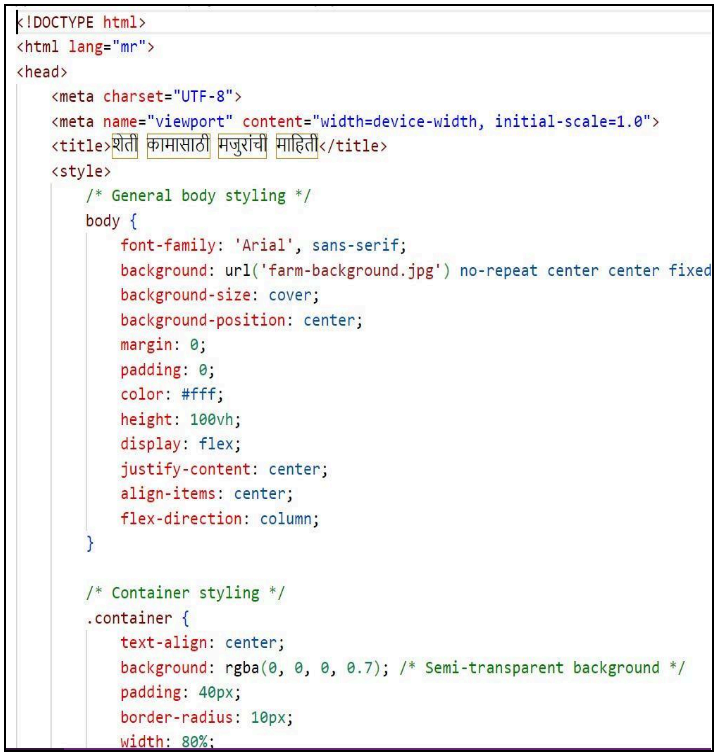
```
<?php
include('db.php');
session_start();

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    $sql = "SELECT * FROM labors WHERE username = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $labor = $result->fetch_assoc();
        if (password_verify($password, $labor['password'])) {
            $_SESSION['labor_username'] = $labor['username'];
            header("Location: labor_dashboard.php");
        } else {
            echo "<p style='color: red;'>Incorrect password!</p>";
        }
    } else {
        echo "<p style='color: red;'>No user found!</p>";
    }
    $stmt->close();
}
?>

<!DOCTYPE html>
```



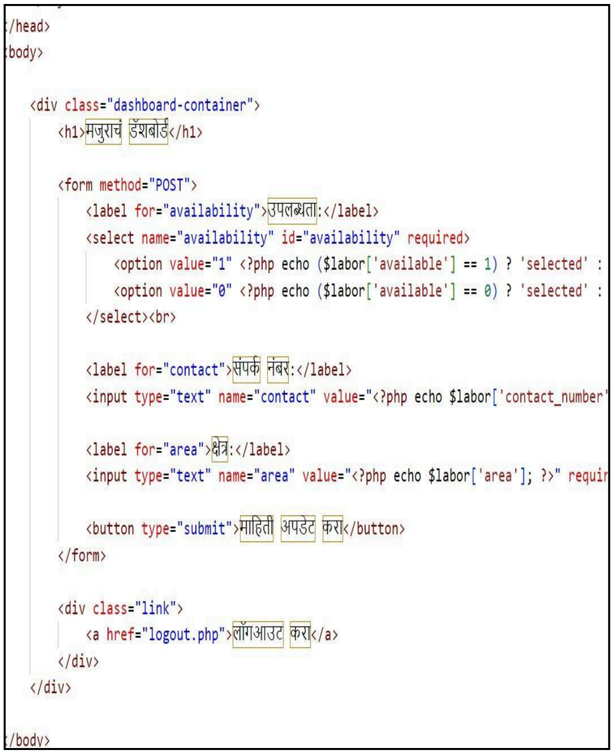
```
<body>

    <div class="dashboard-container">
        <h1>कृषक डॅशबोर्ड</h1>

        <table>
            <thead>
                <tr>
                    <th>मजूराचं नाव</th>
                    <th>वापरकर्ता नाव</th>
                    <th>उपलब्धता</th>
                    <th>संपर्क नंबर</th>
                    <th>क्षेत्र</th>
                </tr>
            </thead>
            <tbody>
                <?php
                if ($result->num_rows > 0) {
                    while ($row = $result->fetch_assoc()) {
                        $availabilityStatus = $row['available'] == 1 ? 'उपलब्ध' : 'उपलब्ध नाही';
                        $availabilityClass = $row['available'] == 1 ? 'status-available' : 'st
                        echo "<tr>
                                <td>{$row['name']}</td>
                                <td>{$row['username']}</td>
                                <td class='$availabilityClass'>$availabilityStatus</td>
                                <td>{$row['contact_number']}</td>
                                <td>{$row['area']}</td>
                              </tr>";
                    }
                } else {
```



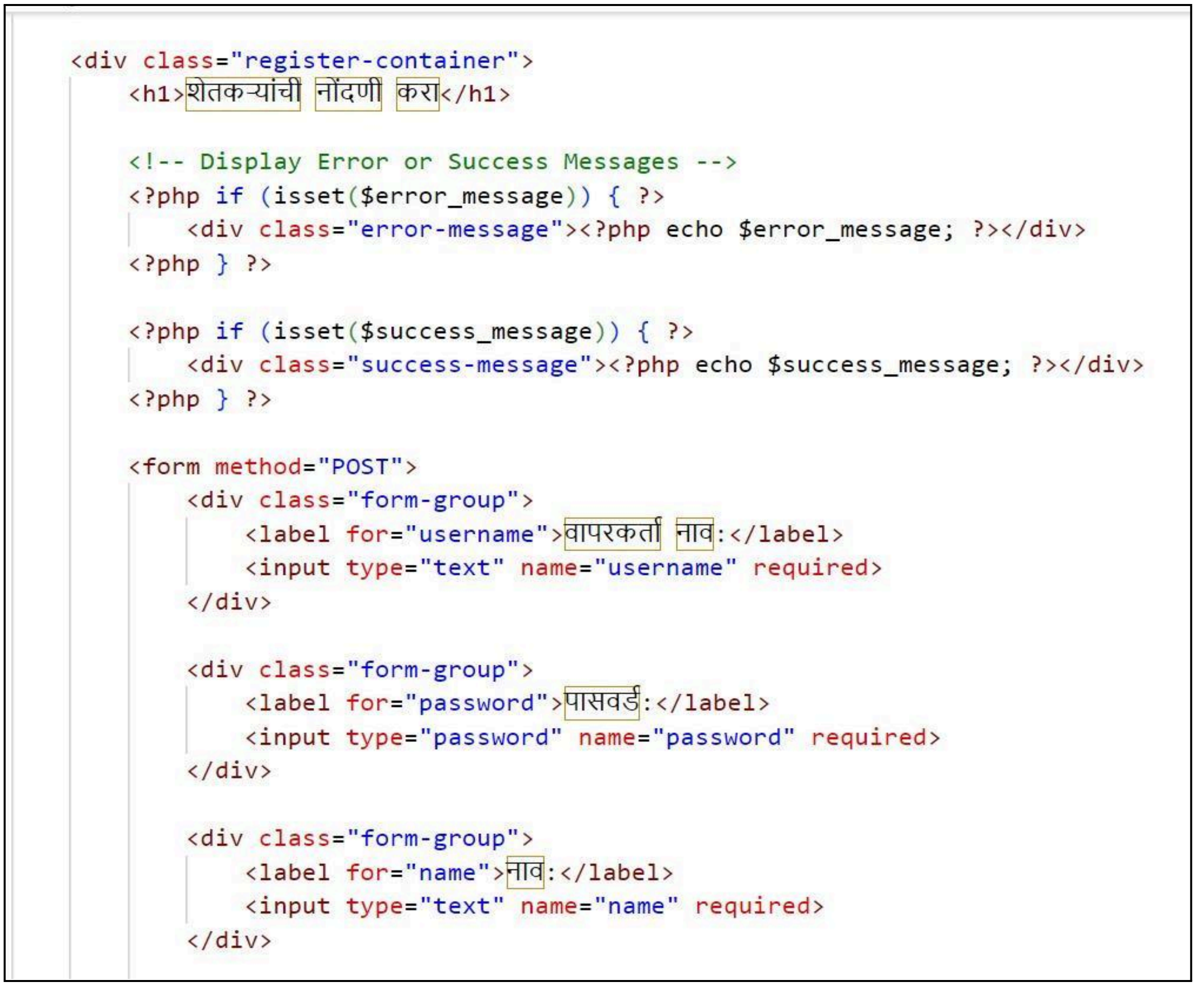
```
<head>
    <style>

        a {
            color: #27ae60;
            text-decoration: none;
        }

        a:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>

    <div class="register-container">
        <h1>मजुरांची नोंदणी</h1>
        <form method="POST">
            <input type="text" name="username" placeholder="वापरकर्ता नाव" required><br>
            <input type="password" name="password" placeholder="पासवर्ड" required><br>
            <input type="text" name="name" placeholder="तुमचं नाव" required><br>
            <input type="text" name="contact" placeholder="संपर्क नंबर" required><br>
            <input type="text" name="area" placeholder="क्षेत्र" required><br>
            <button type="submit">नोंदणी करा</button>
        </form>
    </div>

</body>
</html>
```



```
<!DOCTYPE html>
<html lang="mr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>शेती कामासाठी मजुरांची माहिती</title>
    <style>
        /* General body styling */
        body {
            font-family: 'Arial', sans-serif;
            background: url('farm-background.jpg') no-repeat center center fixed
            background-size: cover;
            background-position: center;
            margin: 0;
            padding: 0;
            color: #fff;
            height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;
            flex-direction: column;
        }

        /* Container styling */
        .container {
            text-align: center;
            background: rgba(0, 0, 0, 0.7); /* Semi-transparent background */
            padding: 40px;
            border-radius: 10px;
            width: 80%;
```



```
</head>
<body>

    <div class="dashboard-container">
        <h1>मजुराचं डॅशबोर्ड</h1>

        <form method="POST">
            <label for="availability">उपलब्धता:</label>
            <select name="availability" id="availability" required>
                <option value="1" <?php echo ($labor['available'] == 1) ? 'selected' :
                <option value="0" <?php echo ($labor['available'] == 0) ? 'selected' :
            </select><br>

            <label for="contact">संपर्क नंबर:</label>
            <input type="text" name="contact" value="<?php echo $labor['contact_number'

            <label for="area">क्षेत्र:</label>
            <input type="text" name="area" value="<?php echo $labor['area']; ?>" requir

            <button type="submit">माहिती अपडेट करा</button>
        </form>

        <div class="link">
            <a href="logout.php">लॉगआउट करा</a>
        </div>
    </div>

</body>
```



```
<div class="register-container">
    <h1>शेतकऱ्यांची नोंदणी करा</h1>

    <!-- Display Error or Success Messages -->
    <?php if (isset($error_message)) { ?>
        <div class="error-message"><?php echo $error_message; ?></div>
    <?php } ?>

    <?php if (isset($success_message)) { ?>
        <div class="success-message"><?php echo $success_message; ?></div>
    <?php } ?>

    <form method="POST">
        <div class="form-group">
            <label for="username">वापरकर्ता नाव:</label>
            <input type="text" name="username" required>
        </div>

        <div class="form-group">
            <label for="password">पासवर्ड:</label>
            <input type="password" name="password" required>
        </div>

        <div class="form-group">
            <label for="name">नाव:</label>
            <input type="text" name="name" required>
        </div>
```

Homepage

- Introduction to the platform, explaining its purpose and benefits for farmers and laborers.
- Links or buttons for farmers and laborers to log in or sign up.

Farmer Login / Registration

- A simple login and registration form for farmers to access their dashboard.
- Registration includes fields for username, password, and area.

Laborer Login / Registration

- Login and registration form for laborers to access their dashboard.
- Registration requires details like username, password, contact information, and area.

Farmer Dashboard

- Personalized dashboard showing available laborers in the farmer's area.
- Each laborer’s name and contact number are displayed for easy communication.
- Logout option for security.

Laborer Dashboard

- Interface for laborers to update their availability status, contact number, and area.
- Shows current information and allows editing.
- Logout option included.

Logout Page

- A simple page confirming the user has been logged out, redirecting them to the login page.

FUTURE SCOPE:

Real-time Availability Tracking

- Enable farmers to see the real-time availability of laborers. Laborers can update their availability with precise time slots, making it easier for farmers to plan.

Location-based Matching

- Use GPS or map integration to show laborers available within a specific radius from the farmer's location, improving relevance and reducing travel needs.

Rating and Feedback System

- Allow farmers to rate and provide feedback on laborers, helping others make informed decisions and promoting reliable work relationships.

Job Posting and Application System

- Farmers can post specific jobs with details like duration, work type, and payment. Laborers can view and apply, making it easier to match work requirements and skills.

Multilingual Support

- Expand beyond Marathi to include other regional languages, making the platform accessible to a wider audience of farmers and laborers across different regions.

Automated Notifications and Reminders

- Implement SMS or app notifications to remind laborers of their scheduled jobs or to notify them of new job postings in their area.

Payment Integration

- Include a secure payment gateway for farmers to pay laborers directly through the platform, with options for online transactions or tracking cash payments.

Weather and Crop Advisory

- Provide weather forecasts, crop management tips, and pest alerts tailored to the farmer's area, making the platform more valuable for farm planning.

Data Analytics and Insights

- Offer farmers and laborers insights such as job trends, demand forecasting, and payment analysis. This can help users make better business decisions.

Mobile Application Development

- Develop a mobile app to increase accessibility for users who may prefer accessing the platform via their phones, which can work offline or in low-connectivity areas.

Laborer Skill Profiles and Certifications

- Allow laborers to showcase specific skills or certifications, helping farmers choose laborers based on the expertise required for specialized work.

Conclusion :

In conclusion, our website serves as a practical, accessible solution for bridging the gap between farmers and laborers in rural areas. By offering a user-friendly platform in Marathi with simple features, it enables farmers to easily connect with available laborers in their region, while allowing laborers to efficiently update their availability and contact information. This direct communication streamlines the process of finding labor for seasonal agricultural work, which can often be challenging in rural settings.

Looking forward, our platform has the potential to incorporate additional features like real-time availability, ratings, job postings, and mobile accessibility, making it even more valuable to users. Ultimately, our website not only addresses an immediate need for labor access but also lays the foundation for future growth, empowering farmers and laborers alike to enhance productivity and build sustainable, supportive communities in agriculture.

REFERENCES:

Digital Solutions in Agriculture

- FAO (2021). *Digital Agriculture*: Overview of digital tools in agriculture for rural support.
- World Bank (2019). *Enabling the Business of Agriculture*: Digital solutions for rural labor access.

Rural Labor Market and Employment

- ILO (2017). *World Employment and Social Outlook*: Insights into rural employment and tech's role.
- Singh, G. & Hiremath, G. M. (2020). *Agricultural Labor in India*: Analysis of rural workforce in agriculture.

Web Development and Database Resources

- Duckett, J. (2014). *PHP & MySQL: Server-side Web Development*: Practical guide on building data-driven websites.
- W3Schools. *PHP Tutorial*: Online resource for PHP basics. w3schools.com

User-Centric and Multilingual Web Design

- Nielsen, J. (1994). *Usability Engineering*: User-centered design principles for accessible interfaces.
- Localisation Research (2022). *Multilingual Interfaces for Rural Communities*: Designing for local languages.

Digital Empowerment in India

- Ministry of Electronics & IT, Govt. of India (2020). *Digital India*: Initiatives for connecting rural communities.