**A project report on**

**BCA-CC-606**

# Labour Management System

Submitted to **Smt. K.B. Parekh College of Computer Science-Mahuva**

**(Affiliated to Maharaja Krishnakumarsinhji Bhavnagar University)**



In partial fulfillment for the award of degree of

# BACHELOR OF

# COMPUTER APPLICATIONS

**Submitted by**

**Jadav Anil Vijaybhai(Seat No: - )**

**Rajyaguru Harsh Vijaybhai ( Seat No: -)**

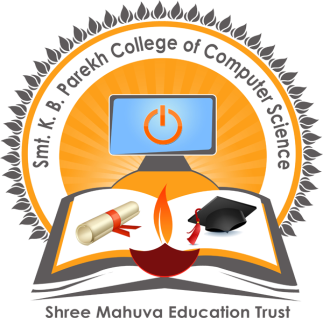
Guided by

VIPUL SOPARIYA

(Assistant Professor)

Smt. K.B. Parekh College of Computer Science-Mahuva

**March – 2022**



**Smt.K.B.Parekh College of Computer Science Mahuva-364290**

**(Affiliated to Maharaja Krishnakumarsinhji Bhavnagar University)**

**TO WHOMSOEVER IT MY CONCERN**

This is to certify that the Student **Jadav Anil , Rajyaguru harsh** of **Smt.K.B.Parekh College of Computer Science, Mahuva** has satisfactorily completed her Project **The website of Labour Management System** during the period December 2021 to March 2022 in the partial fulfillment of BCA-CC-606.

**Name & Signature of Project Guide Signature and Seal of Principal.**

**Mr. vipul sopariya**

**::Sign::**

**Address**: Smt.K.B.Parekh College of Computer Science, Prabhat Nagar Road, Near Parekh College Campus, Cooperative Housing Society, Mahuva, Gujarat 364290 Ph-02844/228332

**Email:kbpbcamahuva2000@gmail.com**



**A Farmers Digital Ledger………**

**Live Website :** [**http://WWW.LABMAN.XYZ**](http://WWW.LABMAN.XYZ)

**Acknowledgement**

I would like to take this opportunity to express my sincere thanks and gratitude to my college K.B.Parekh College Of Computer Science for all their guidance, inspiration, constructive suggestions which help me in the project.

We are taking this opportunity to thank Maharaja Krishna Kumarsinhji Bhavnagar University for giving us the opportunity to work in **PHP** This has helped us and taught us to work with discipline and perseverance. We are thankful to **Mr**. **Vinodbhai makwana** who gives their precious time to help us in our PHP project. They were there with me in every step of the way and their immense experience helped me when I faced seemingly insurmountable obstacles.

Last but not least I would like to thanks all the people who have directly indirectly helped and encouraged me in completing the project effectively and timely.

**Acknowledged by:**

**Jadav Anil v.**

**Rajyaguru Harsh v.**

**Abstract**

Labour management system is used to manage all the activity related to farm labours. Farmers work very hard to manage these labours ledger and accounts. Its very hard to track and manage work and payment details of labours.

Today all industries are digitalize but in agriculture we can’t see any digitalization. As all know agriculture is big and un-managed industry. Farmer daily write ledger which labours work in farm mention date and amount and many details. Famer has no records of payment and no tracks of transactions.

To solve this problem we developed this system To manage all these farm labour activity digitally. This software helps farmers to manage farm labours payment and track records easily.

Using this tool farmer create a list of labours work in their farm and set amount of labours

Farmer can create multiple list day vise or as per need easily.

Farmer just need to add labour only one time after adding labour system store labours. Its help farmer while creating a list just select which labour working today in their farm.

This tool shows the payment details and records. Famer easily track payment history , transactions , paid amount , unpaid amount , total day of present and more in any order like monthly , yearly , daily and all time. Also farmer get summary of labours.

Farmer easily modify list , labour or transactions and farmer able to perform operations like add , remove , update any list or labour.

This tool run in any platform like laptop , tablet or smartphone with single login just you need any browser.

The aim of this application was to develop a web based tool, which allows easily managing the farm labour management process and help farmer by providing tool for a easy to manage labours.

We create this tool using php,html,mysql,css and javascript. We use readymade template for a good user interface.

**INDEX**  page no

**• CHAPTER 1**: Introduction

1.1 Background 7

1.2 Objectives 7

1.3 Purpose, Scope, Applicability 8

**• CHAPTER 2**: Requirements and Analysis

2.1 Problem Definition 9

2.2 Requirement Specification 10

2.3 Software and Hardware Requirements 11

2.4 Planning and scheduling 11

2.5 Model 12

**• CHAPTER 3:** System Design

3.1 Overall System design using designing Tools 16

3.2 Data dictionary 21

3.3 Input and Output Design 27

**• CHAPTER 4:** Testing & Implementation

4.1 Testing Approach Uses 39

4.2 Test Cases 44

4.2 Implementation Approach 50

**• CHAPTER 5:** Conclusion

5.1 Conclusion 52

5.2 Limitation of System 53

**• CHAPTER 1**:

**1.1 Introduction & background**

Labour efficiency in agriculture refers to the amount of productive work accomplished per man on the farm per unit of time. In general , the higher the labour efficiency, the greater are the returns from farming. The farmer who dose not introduce new methods and techniques to increase productivity can soon go out of the business because of increasing labour costs. Inefficient labour also results in low production which in turn means low wages for the labour.

Famers are un-skilled but today 70%-80% farmers has a smartphone but they write labour present and maintain daily ledger in book. in the market this type of software are not available that solve this problem that’s why we develop this tool for famers.

Labour management system is a tool that manage farm labours. This tool is a ledger for famers that helps farmer to easily manage labours list , payment tracks , Amount(paid, unpaid , total) , statistics and summary of farm labours.

**1.2 Objectives**

The main objective of this tool is to provide a digital ledger for farmers to manage working labours in farm.

To reduce the paper work and provide a online tool for manage labours attendance , payment and track transactions.

Most of farmers are don’t know to use websites or software that’s why we provide a simple and easy tool with simple good user interface.

The aim is to help the famers to manage daily labours ledger and digitalize the famers to grow in field and use the information technologies.

* 1. **Purpose**

Purpose of this tool is To reduce the paper work and provide a online tool for manage labours attendance , payment and track transactions.

Digitalize the famers and provide a good labours ledger to get a perfect statistics of famer and track their payment.

**1.3**  **Scope**

Provide this tool to all farmers and help them to reach out the writing labours ledger problem. All farmers use this tool and give feedback and suggestions for future enhancement and add more features.

* labour legder
* payment tracks
* analytics & statistics
* list vise and date vise data
* sort and summary of lists and labours
* summary of payments
* show paid and unpaid amount
* message and chat with admin

**1.4 Applicability**

As far as applicability is concern our project is in such reliable format so that every people can get their required queries in a smooth manner. Since, our website is informative also due to which if someone wants details about oil, then they can also acknowledge themselves with the conventional guidance.

**CHAPTER 2**:

Requirements and Analysis

* Requirements analysis is also known as requirements engineering. The process of determining user expectations for a new or modified hardware or system. These features are called requirements and it must be quantifiable and relevant and detailed. In software engineering, these requirements are called functional specifications. Requirements analysis is an important aspect of developing and managing the project.
* Requirements analysis is a team effort. It demands a combination of hardware, software and human factors engineering.
* Requirements Analysis is the first major step in the Systems Engineering Process. This step examines each requirement to see if it meets the characteristics of a good requirement. Each requirement is then decomposed into a more refined set of requirements that are allocated to sub.

**2.1 Problem Definition**

It industry is growing in very short time in this era it is now every where because of its advantages and features. Now days software same time and provide accuracy compare to humans.

Every field adopt it but agriculture industry use less it. Farmers maintain their daily labour in record books they write labours attendance and payments in a ledger.

We found a issue and a opportunity to provide it advantages throw a software.

Farmers need to digitalize his records books in software so that he save his time easy to access data and records , easy to payment tracks and also he can see where my money spent how much.

**2.2 Requirement Specification**

This **System Requirement Specification** (SRS) aims to provide the readers and users information about the **system** and its functions and **specifications**. SRS describes the data, functional and behavioral **requirements** of the software. This will make it easier for the users to control and manage the use of **Labour Management System.**

1. **Functional Requirement**

* List module to Add labours (only one time)
* Create list providing basic information like list\_name,date,work,labour charge
* Dashboard to display amount (paid , un-paid)
* Payment module to manage transaction
* Summary and statistics module
* Message , Notification and Chat module

1. **Non-Functional Requirement**

* **Security:**  login module to block un-authorized access of the data using phone number and password
* **User friendly :** user interface is very simple and user friendly any non technical person can easily understand the use. Clean user experience with plain simple ui.
* **Maintainability:**  backup facility and multi device login support.
* **Performance:** easy to create update and delete list and manage transactions,

All task done smoothly and speedy

**2.3 Hardware Requirement**

* Windows 7/8/10
* Mac OS
* 2 GB RAM
* 1GB ROM
* System Type 64-bit operating system.
* any screen type

**2.4 Software Requirement**

* **Front End:** CSS3 ,HTML5
* **Back End:** MY-SQL , PHP
* **Documentation Tools:** MS OFFICE 2007
* **Other Tools :** JAVASCRIPT
* **Operating System :** ANY OPERATING SYSTEM
* **Web Browser:** Any
* **Admin panel template :** Purple dashboard (from www.bostrapdash.com)

**Client- Side Tools**

* + **Processor:** PC with a dual core processor or above is recommended.
  + **RAM:** 512 MB of available or onwards Recommended.
  + **Hard Disk:** 45 MB of available space required.
  + **Operating System:** Any OS
  + **Browser:** Any (chrome , firefox , opera etc..)

**Project planning & Scheduling**

* **Project Scheduling**

|  |  |  |
| --- | --- | --- |
| *Task* | *Duration* | *Date* |
| * **Analysis** | **1 week** | 1 December 2021 to 7 December 2021 |
| * **Design** | **2 weeks** | 8 December 2021 to 31 December 2021 |
| * **Coding** | **5 weeks** | 1 January 2022 to 7 February 2022 |
| * **Testing** | **1 week** | 8 February 2022 to 15 February 2022 |
|  | **9 weeks** | 1 December 2021 to 15 February 2022 |

**MODEL: Spiral Model**

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

**Spiral Model - Design**

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

**Identification**

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

**Design**

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

**Construct or Build**

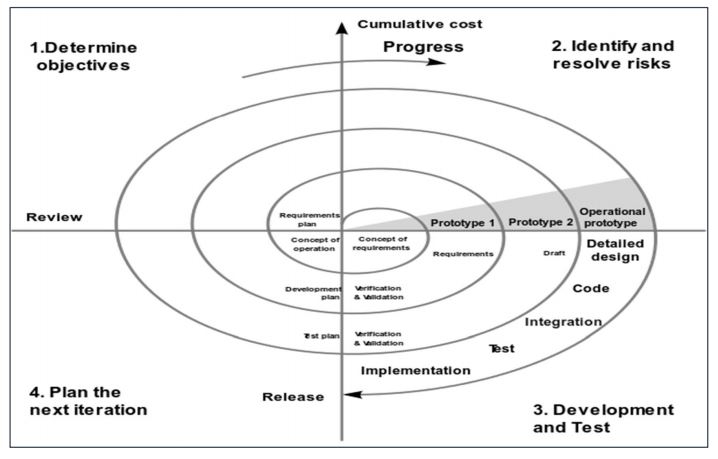
The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

**Evaluation and Risk Analysis**

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

The following illustration is a representation of the Spiral Model, listing the activities in each phase.



**(Figure 2)**

Based on the customer evaluation, the software development process enters the next iteration and subsequently follows the linear approach to implement the feedback suggested by the customer. The process of iterations along the spiral continues throughout the life of the software.

**Spiral Model Application**

The Spiral Model is widely used in the software industry as it is in sync with the natural development process of any product, i.e. learning with maturity which involves minimum risk for the customer as well as the development firms.

The following pointers explain the typical uses of a Spiral Model −

* When there is a budget constraint and risk evaluation is important.
* For medium to high-risk projects.
* Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
* Customer is not sure of their requirements which are usually the case.
* Requirements are complex and need evaluation to get clarity.
* New product line which should be released in phases to get enough customer feedback.
* Significant changes are expected in the product during the development cycle.

**CHAPTER 3:**

**System Design**

# 3.1 Over All System Design Using Designing Tools

The Purpose of Design Phase is to plan a solution for problem specified by the requirements. System Design aims to identify the modules that should be in the system, the specification of those modules and how they interact with other to produce the results. The goal of the design process is to produce a model that can be used later to build that system. The produced model is called design of the system.

System Design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements.

Normally, the design proceeds in 2 stages:

* Physical Design
* Database Design

### Physical Design

The Physical Design is a graphical representation of a system showing the system’s internal and external entities and the flow of data into and out of these entities. An internal entity is an entity within the system that transforms data.

To represent the Physical Design of the system, we use diagrams like Data Flow Diagrams, E-R Diagrams, Use Case Diagrams, etc…

**UML DIAGRAMS**

UML Stands for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML contain following diagram as follows:

* Use case Diagram
* Data Flow Diagram
* Activity Diagram
* Class Diagram

### Data Flow Diagram

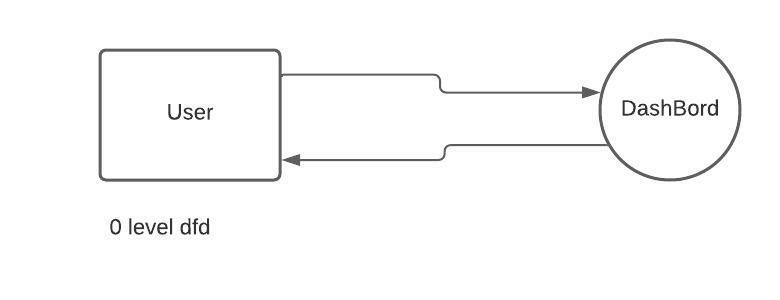
The Data Flow Diagrams (DFD) is a graphical representation of the flow of data through an information system. Data Flow Diagrams are used by systems analysis to design information processing systems but also a way to model whole organization. The main merit of DFD is that it can provide an overview of what data a system would processes. What transformations of data are done, what data are stored and which stored data is used, and where the result is flow.

### 

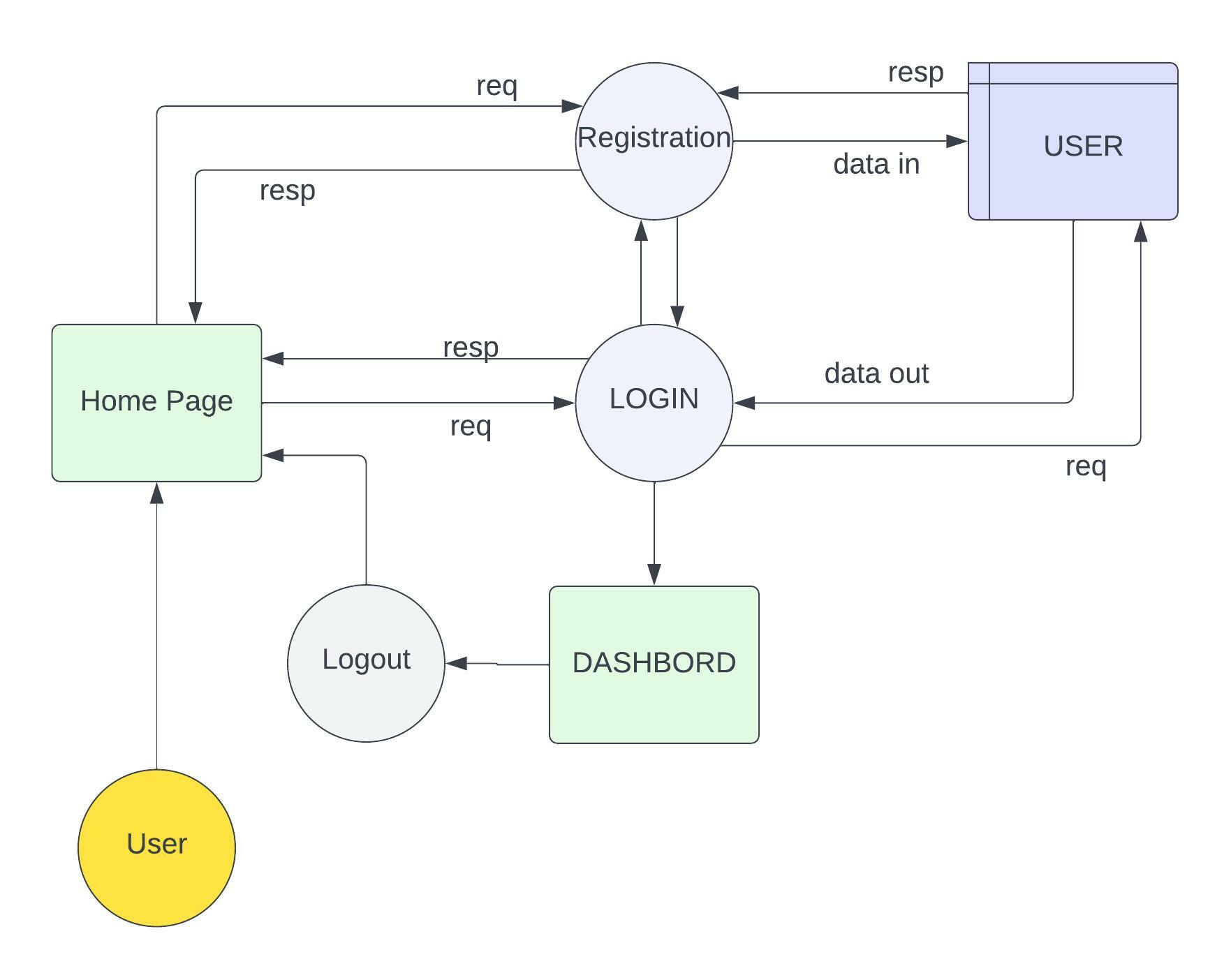
### Standard Symbols used in DFD:

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Function** |
|  | **Data Flow** | Used to connect processes to each other. The Arrowhead indicates the direction of Data Flow. |
|  | **Process** | Performs some transformation to input data to output data. |
|  | **Input / Output** | It is used to Input or Output data. |

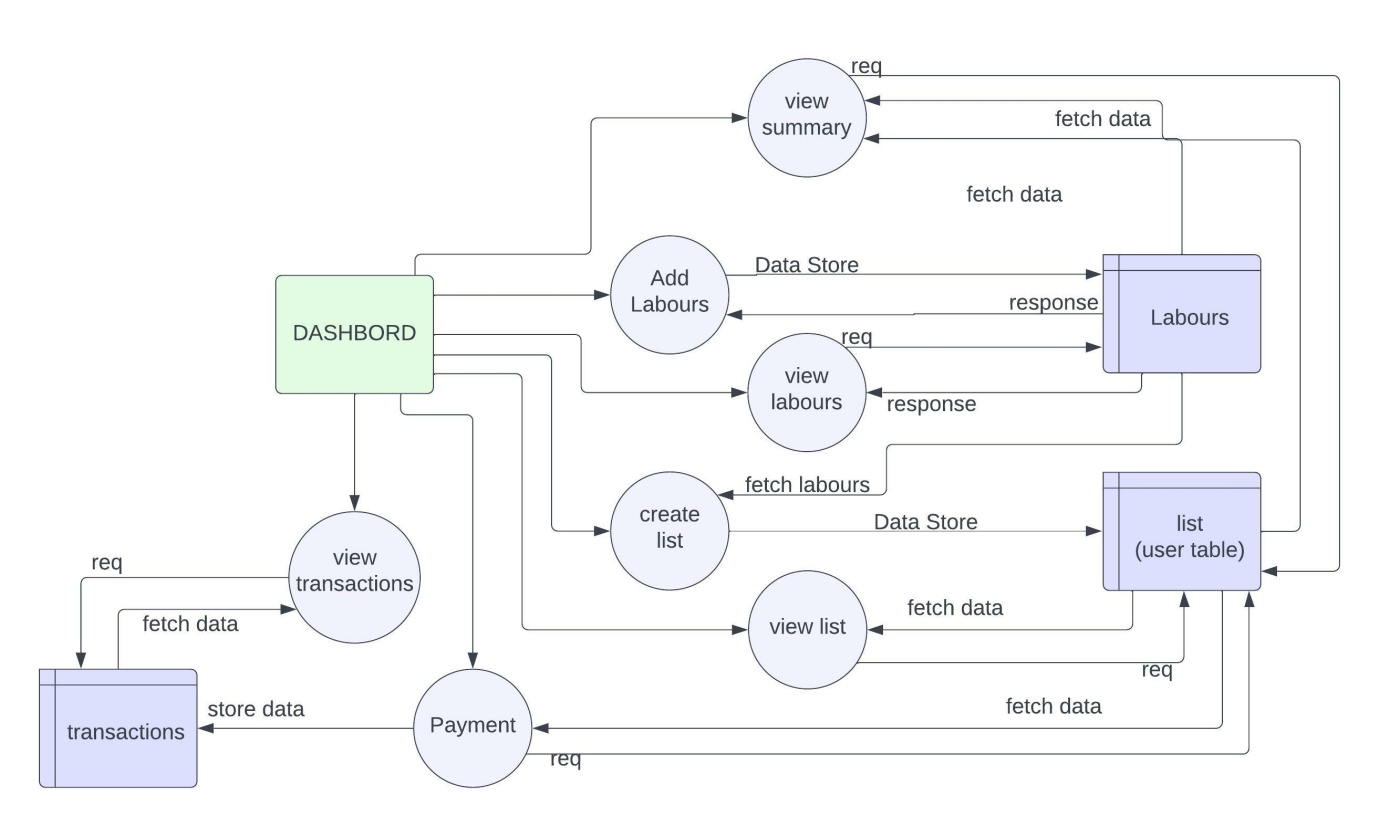
0 Level DFD



1st Level DFD



2nd Level DFD

****

**3.2 DATA DICTIONARY**

Database Name: **labman**

(1)Table name: **user**

Description: Store registered user data and also use for login

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Description |
| Id | Int(11) | Primary key (Auto Increment) |
| Phone | varchar(10) | Unique |
| password | varchar(50) | Not null |
| name | varchar(30) | Not null |
| type | varchar(6) | Not null |
| Img | varchar(100) | Not null |
| address | varchar(100) | Not null |
| reg\_date | Datetime | Not null |
| last\_login | Datetime | Not null |
| dbname | varchar(10) | Unique |

(2)Table name: **labours**

Description: Store labours details for of all accounts

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Description |
| lId | Int(11) | Primary key (Auto Increment) |
| uid | Int (11) | Foreign key on user table(id) |
| userdb | varchar(10) | Foreign key on user table(dbname) |
| name | varchar(50) | Not null |
| address | varchar(30) | Not null |
| location | varchar(20) | Not null |
| Number | varchar(12) | Not null |
| gender | varchar(6) | Not null |
| reg\_date | Datetime | Not null |

(3)Table name: **user table (table name is created using combination of account details)**

Description: this table is dynamically created when user register. Using combination of user id , name and numbers dynamic table is created and table name is stored in user table.

example: 21an7874 (21 is id , an is first 2 latter of name , 7874 is first 4 latter of mobile number)

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Description |
| Id | Int(11) | Primary key (Auto Increment) |
| listid | Int (11) | unique |
| lid | Int(11) | Foreign key on labour table(lid) |
| listname | varchar(100) | Not null |
| name | varchar(30) | Not null |
| location | varchar(20) | Not null |
| work | varchar(20) | Not null |
| count | Int(11) | Not null |
| amt | float | Not null |
| status | varchar(11) | Not null |
| pdate | Date | null |
| ldate | Date | Not null |

(4)Table name: **Transactions**

Description: store payment transactions details

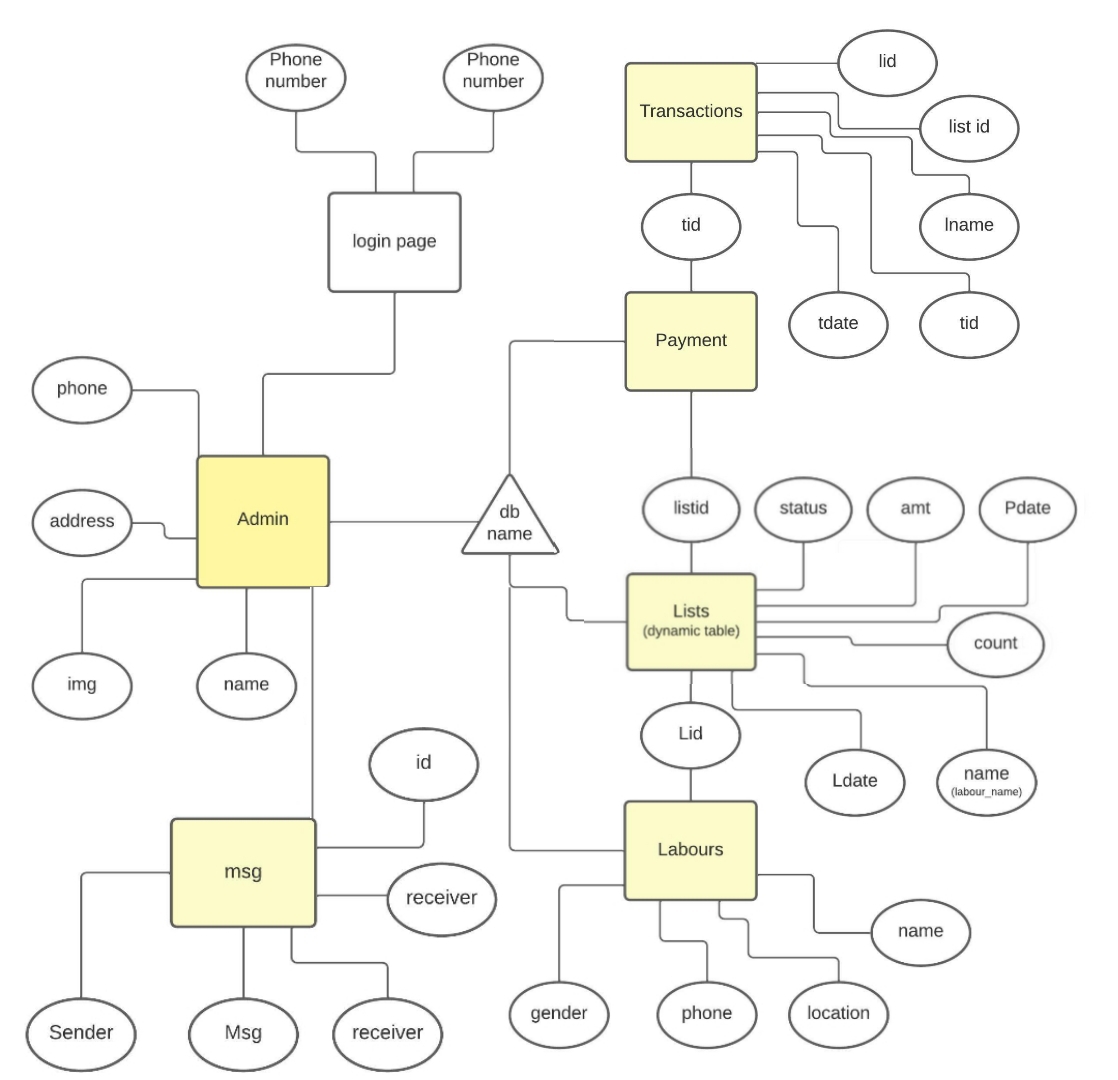
|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Description |
| tid | Int(11) | Primary key (Auto Increment) |
| userdb | varchar(10) | Foreign key on user table(dbname) |
| lid | Int(11) | Foreign key on labour table(lid) |
| listid | Int(11) | Foreign key on dynamic table(listid) |
| lname | varchar(20) | Not null |
| amt | float | Not null |
| rid | Int(11) | Foreign key on dynamic table(id) |
| tdate | Date | Not null |

(5)Table name: **msg (Message table)**

Description: user send and receive messages using this table

|  |  |  |
| --- | --- | --- |
| Column name | Data Type | Description |
| id | Int(11) | Primary key (Auto Increment) |
| reciver | varchar(10) | Foreign key on user table(dbname) |
| Msg | varchar(150) | Not null |
| title | text | Not null |
| sender | varchar(10) | Foreign key on user table(dbname) |
| status | varchar(6) | Not null |
| date | Date | Not null |

**E R diagram:**

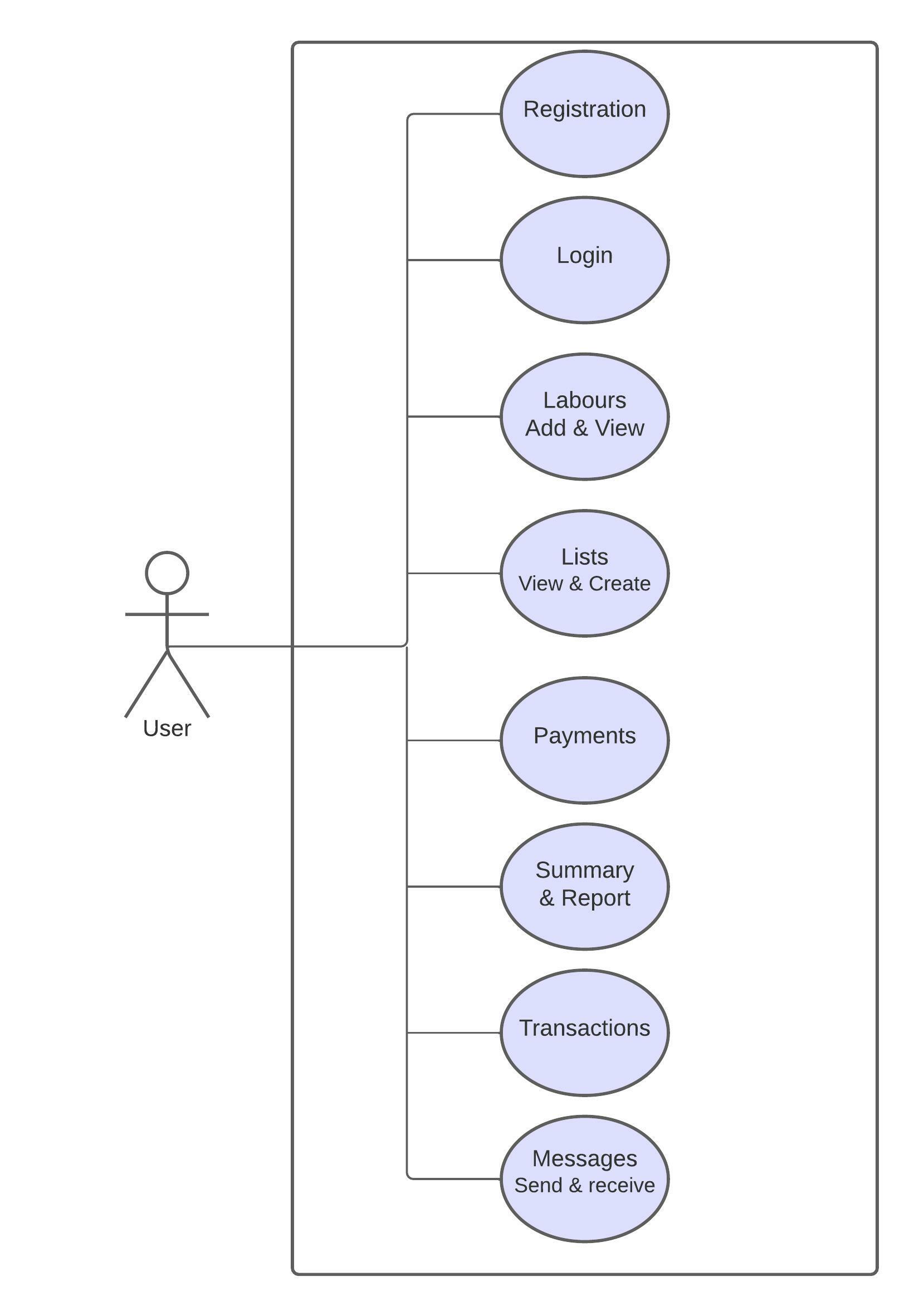
****

**Use Case Diagram:**

* **the purposes of use case diagrams can be said to be as follows −**
  + Used to gather the requirements of a system.
  + Used to get an outside view of a system.
  + Identify the external and internal factors influencing the system.
  + Show the interaction among the requirements are actors.

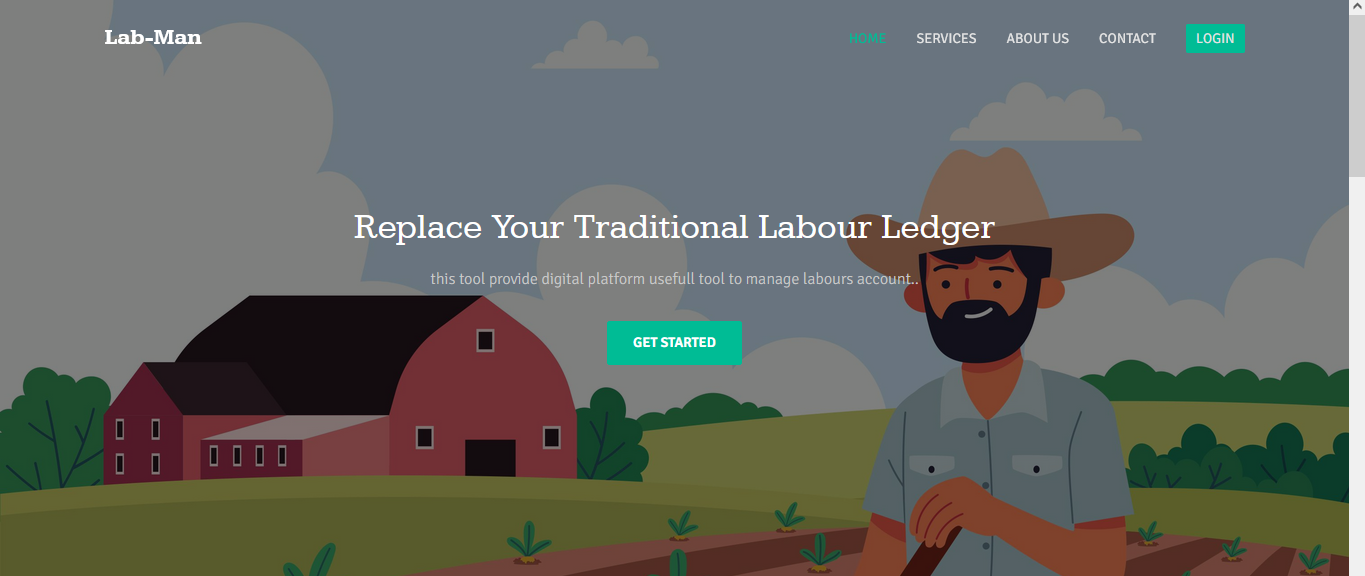
|  |  |
| --- | --- |
| Boundary for Use cases |  |
| For Actor |  |
| For use cases ( Known as oval) |  |
| For connection between Actors and ovals |  |

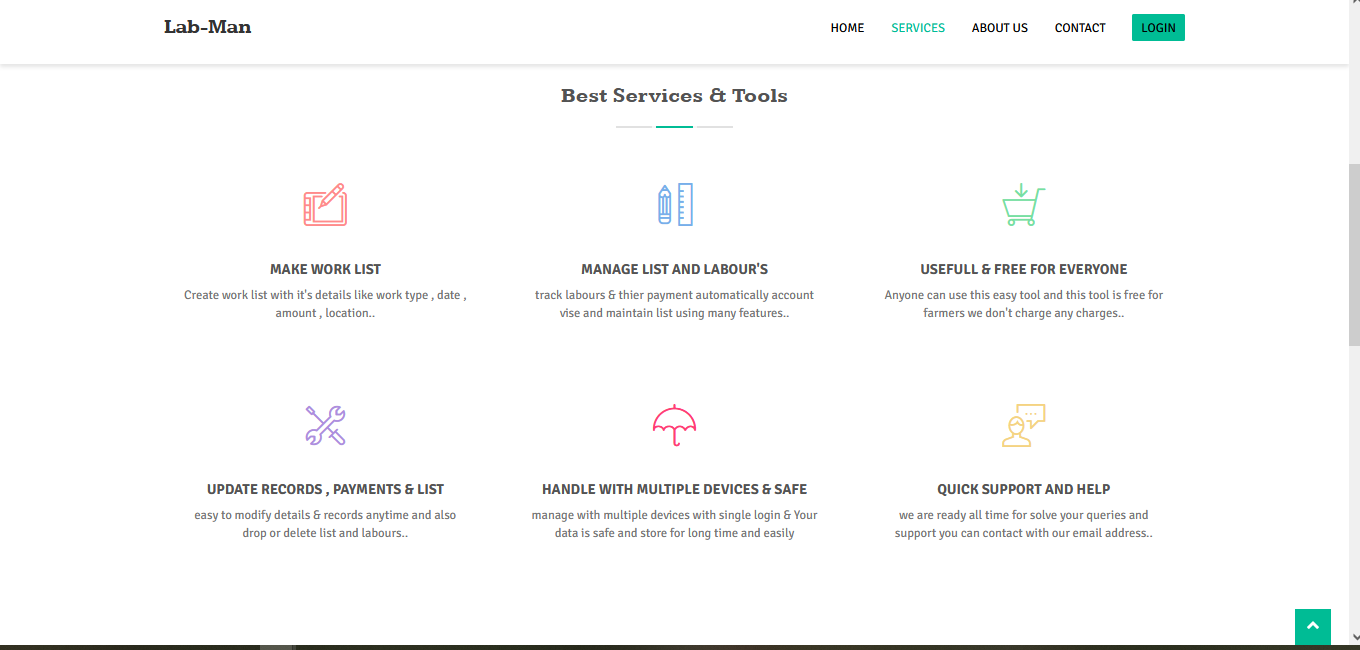
**Use case Diagram**

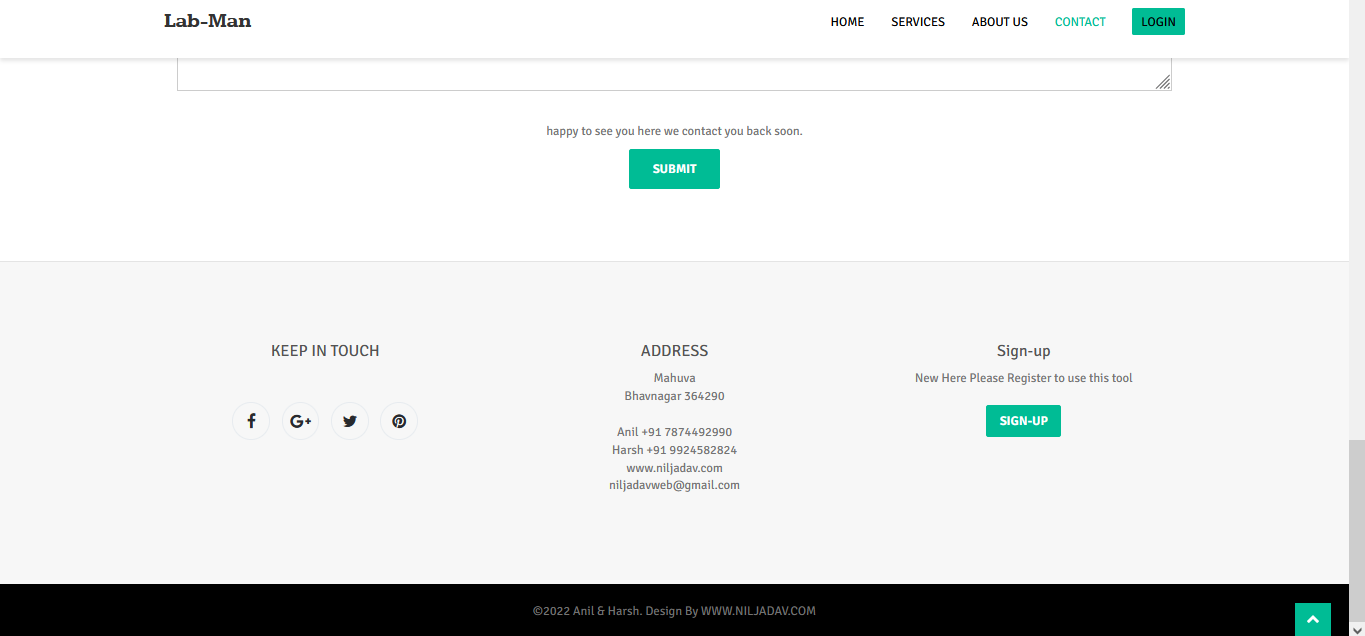
****

**3.3 Input / Output Design**

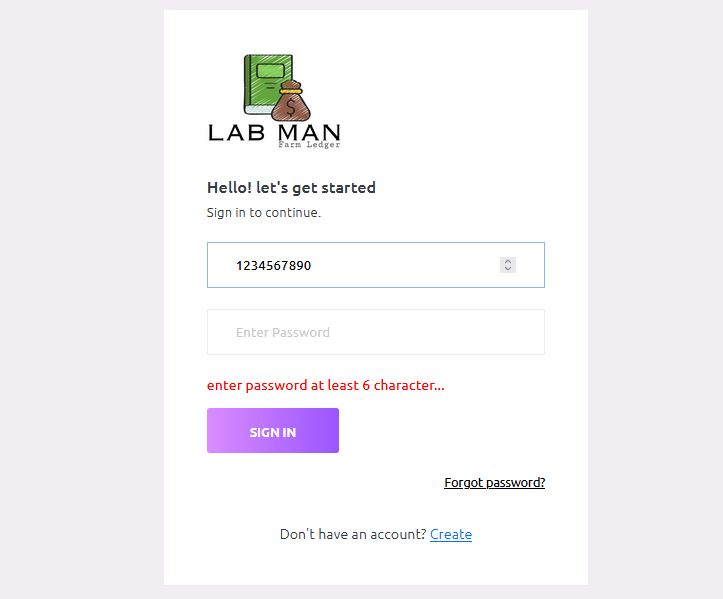
* Home Page

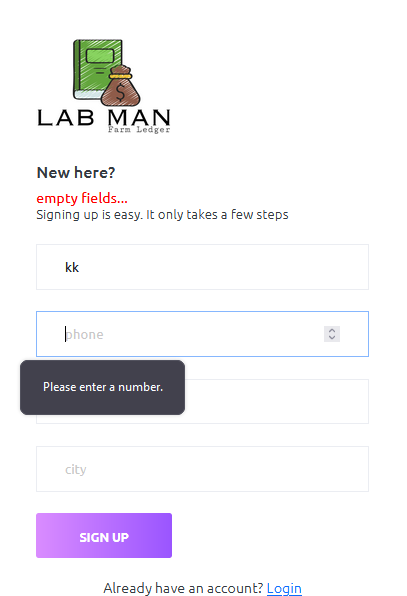




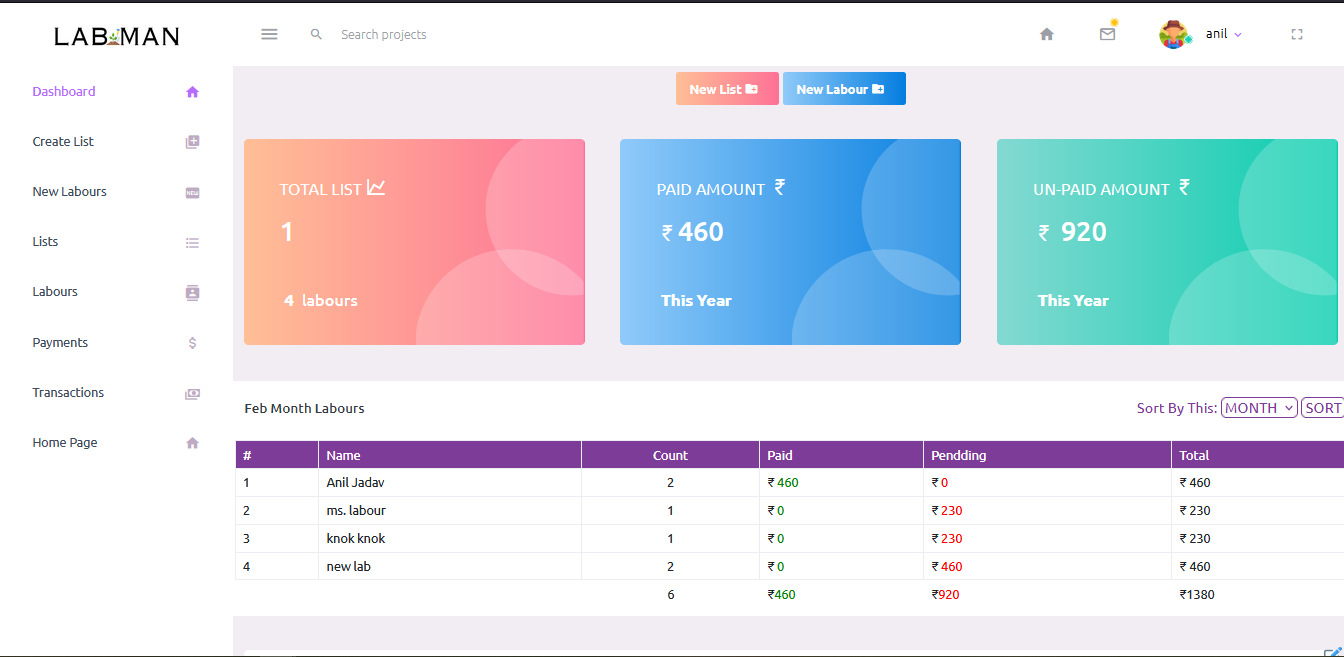


* Login / Registration page

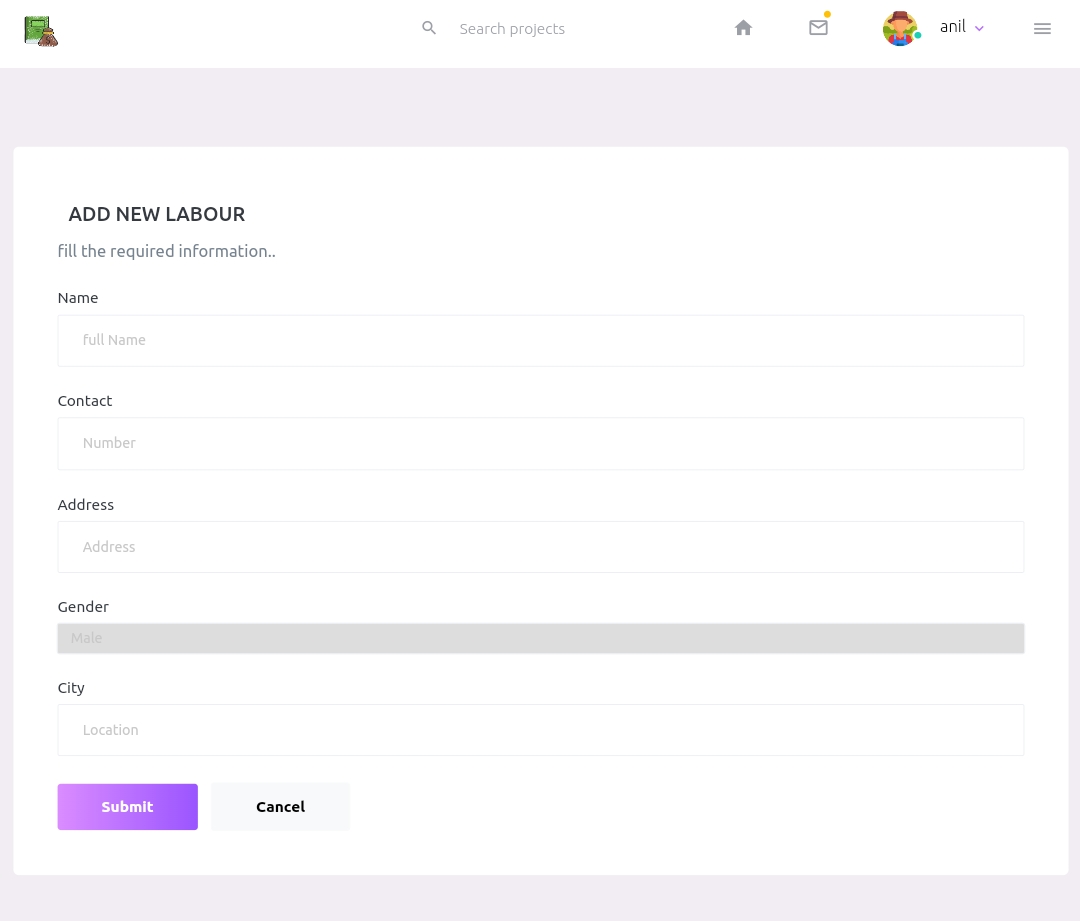




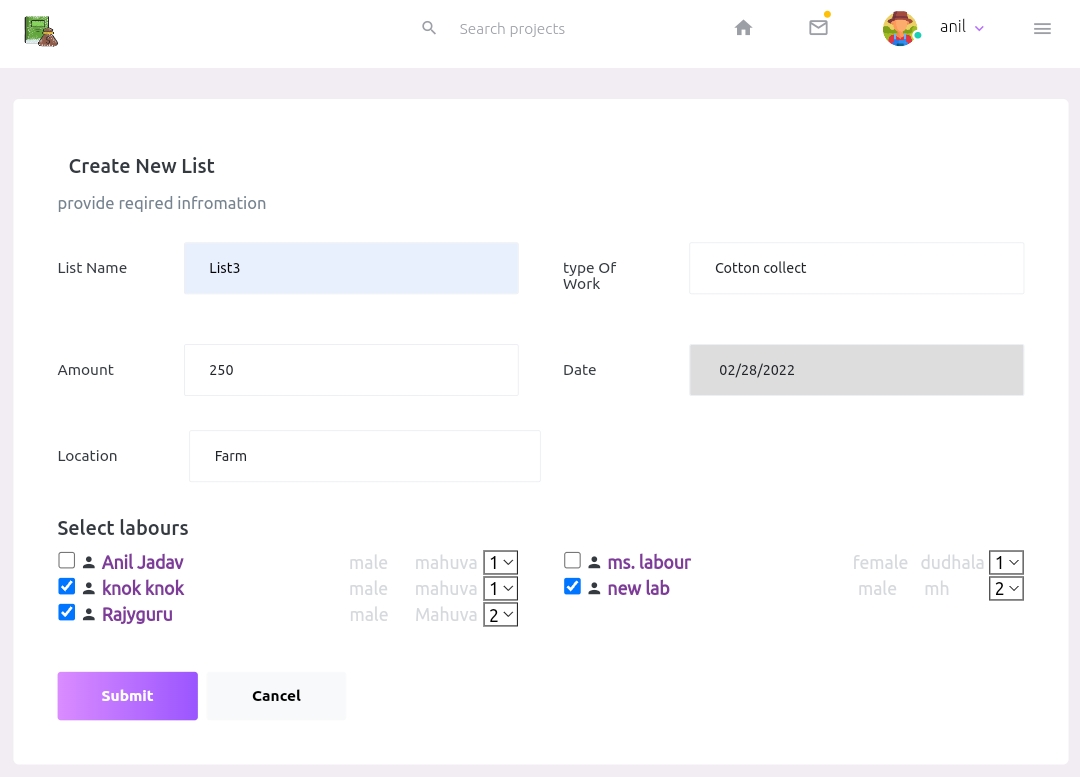
* **Dashbord**



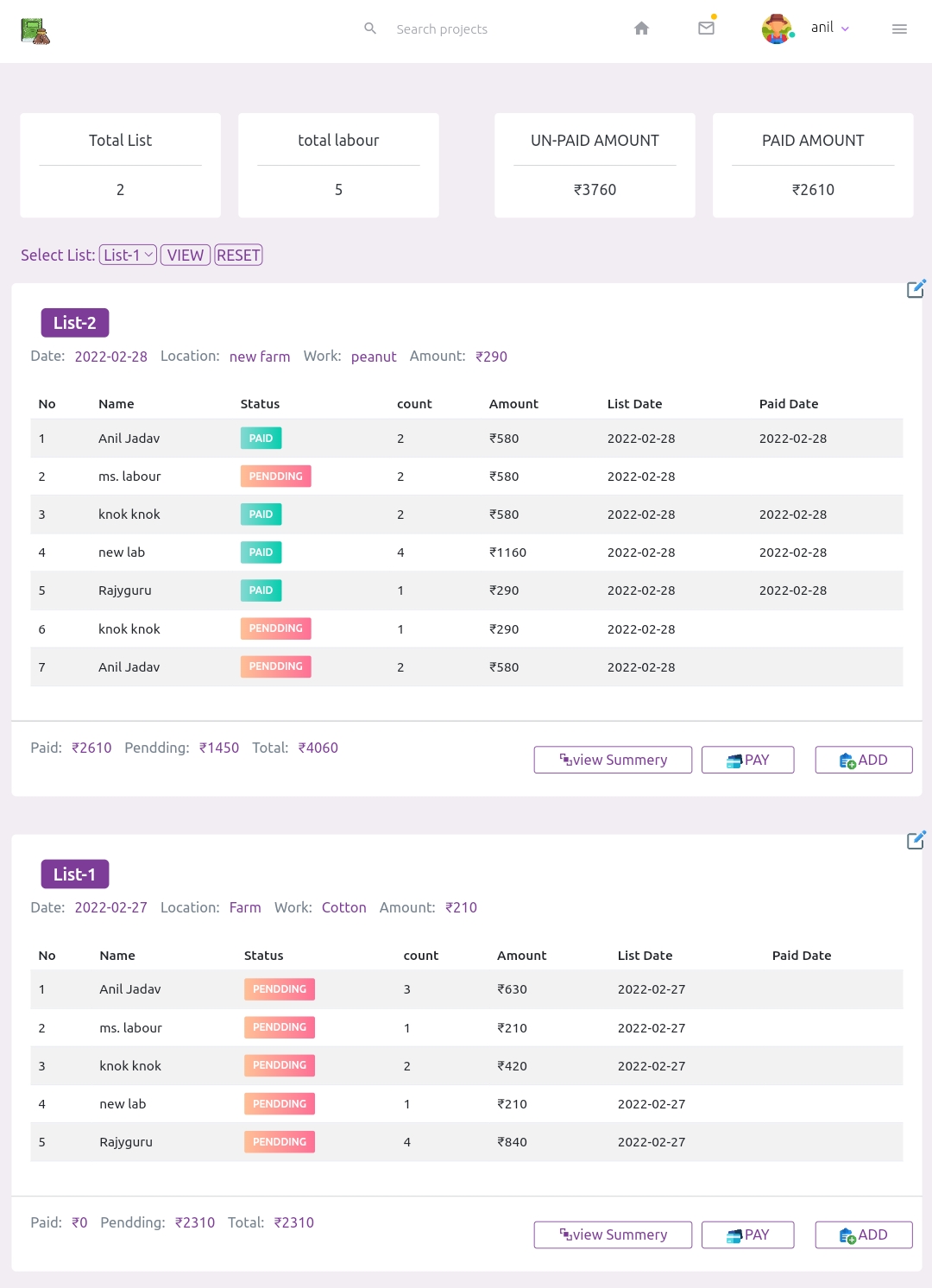
* Add New Labours



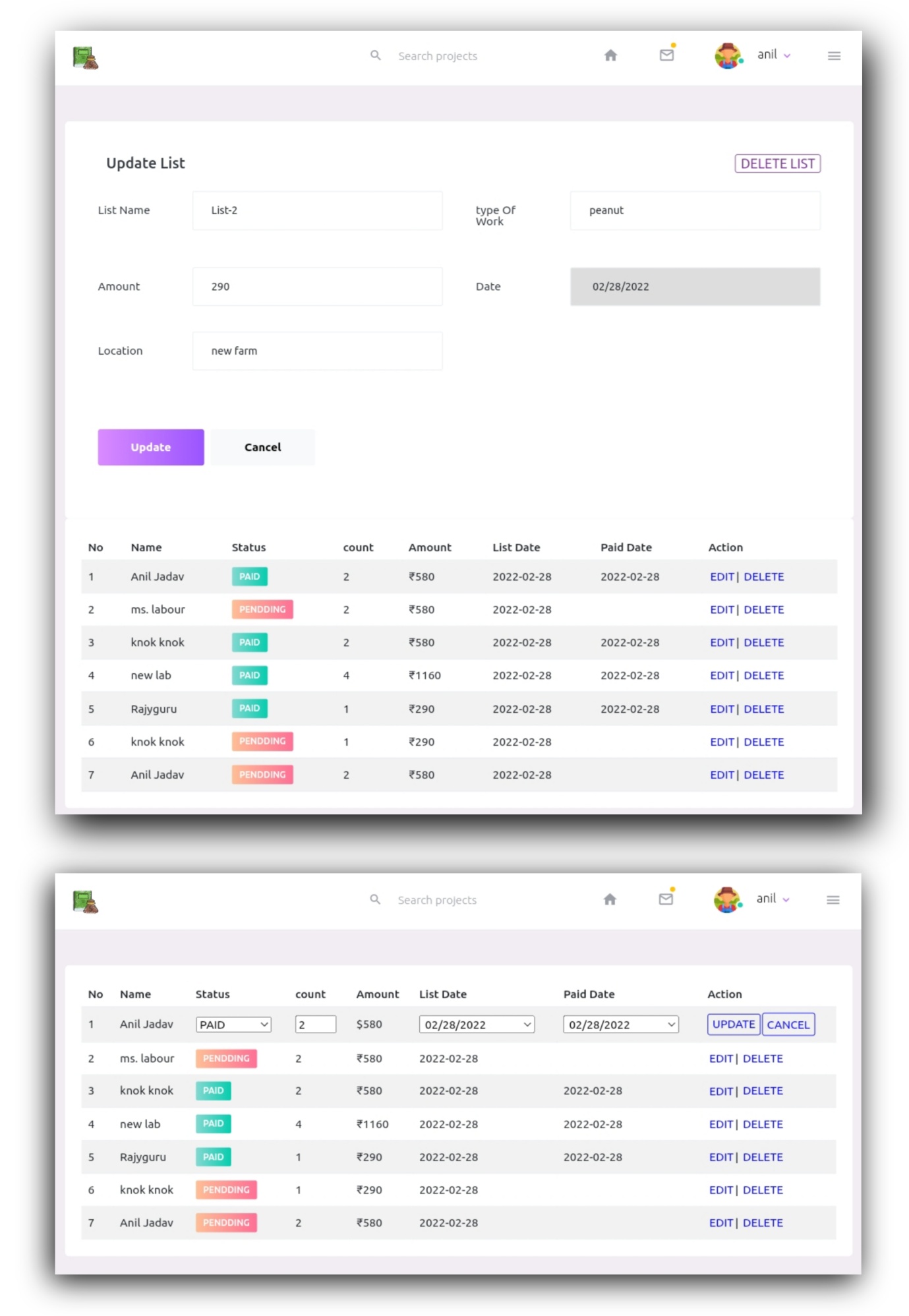
* Create New List



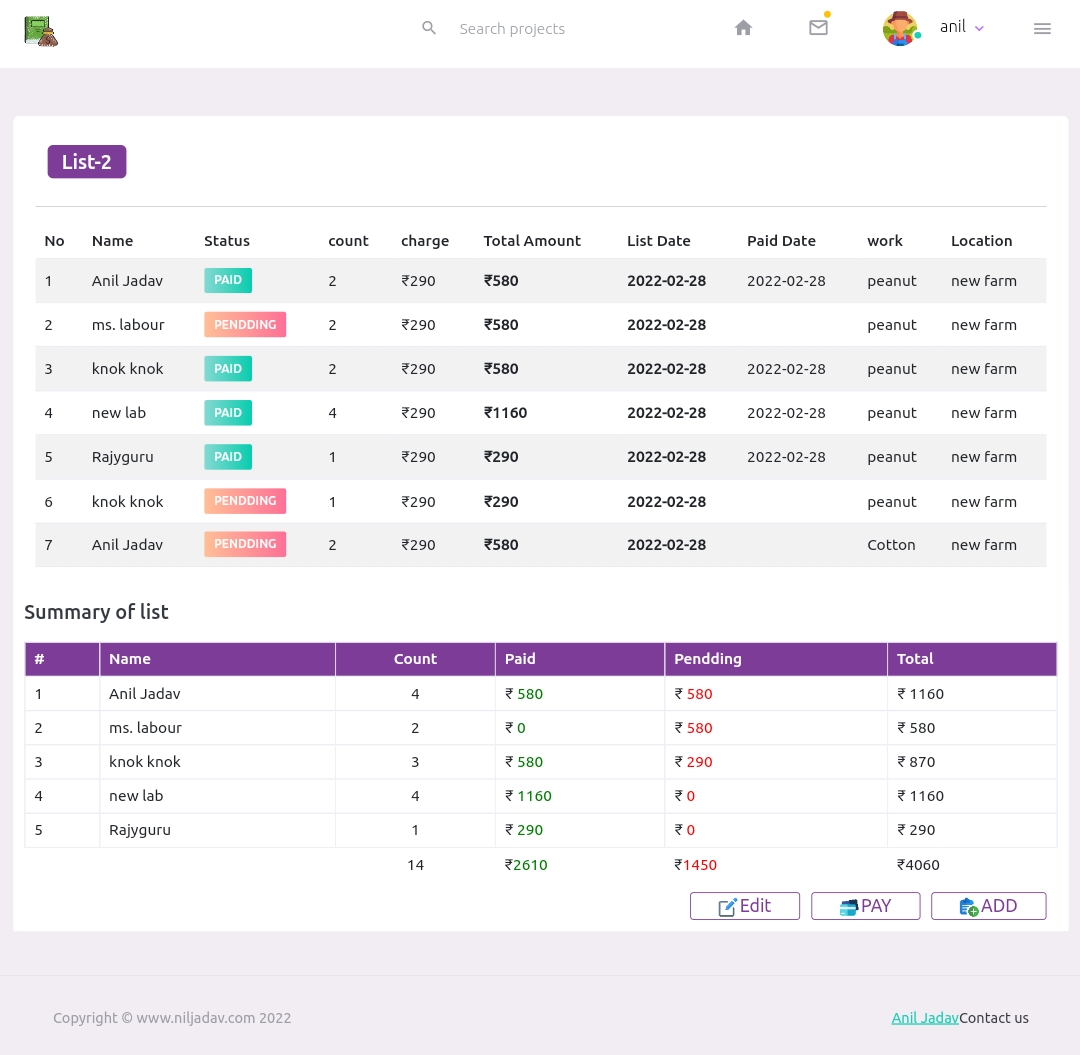
* View List



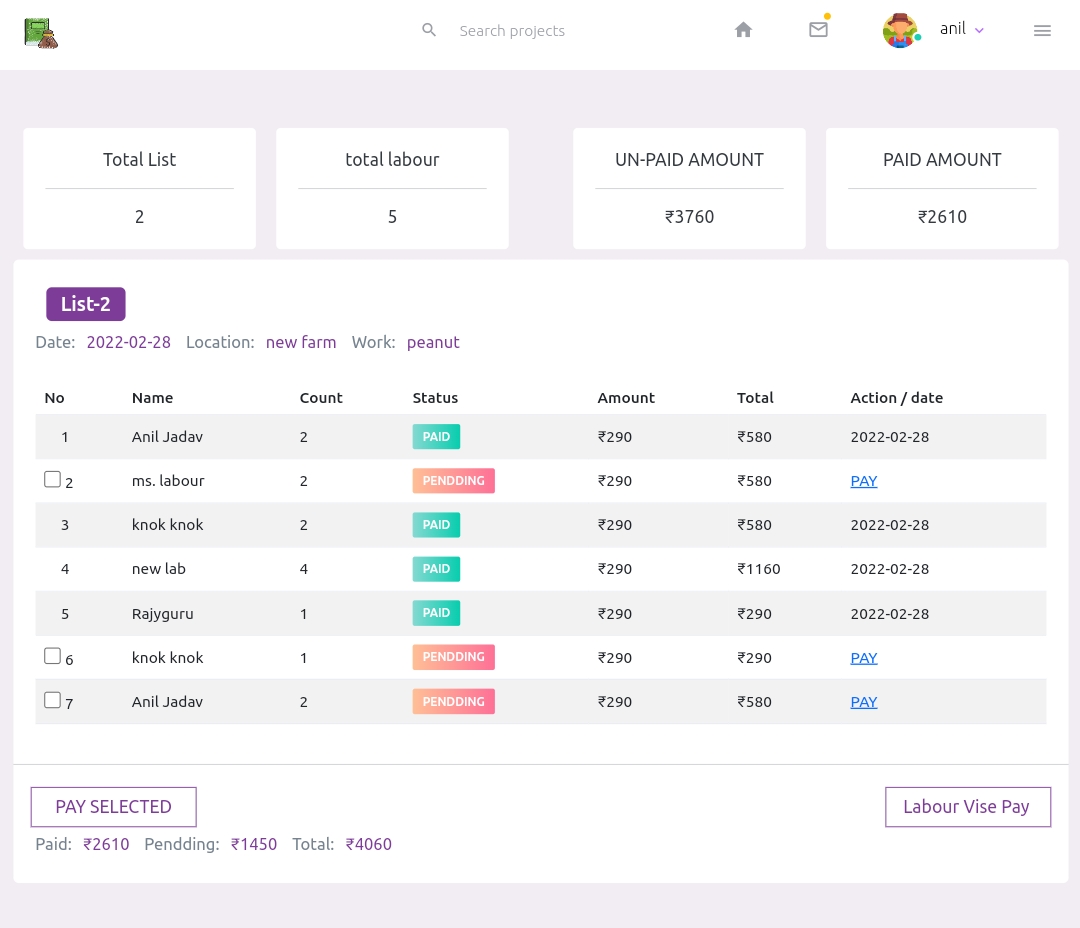
* Edit / update list



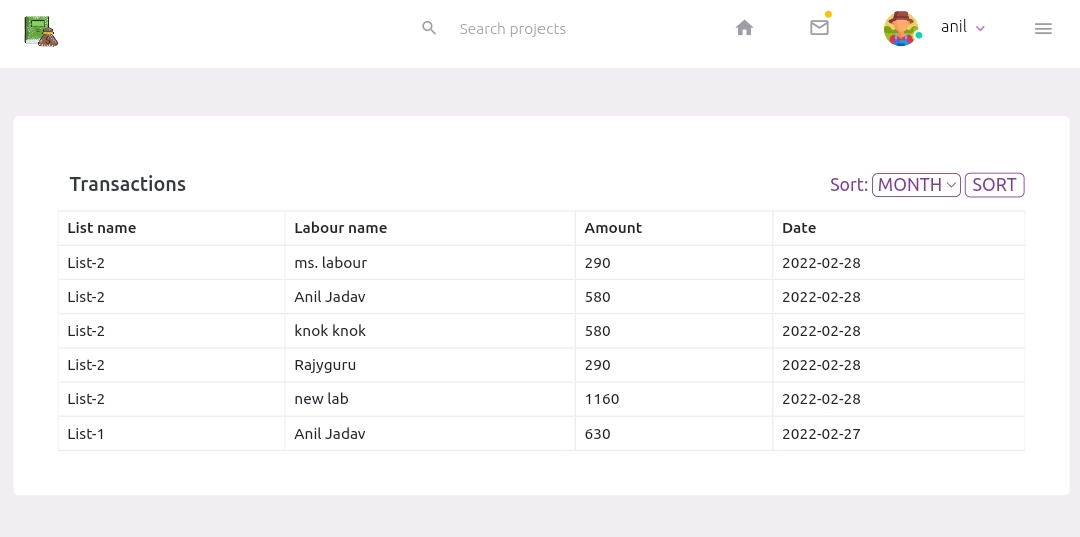
* View Summary of List

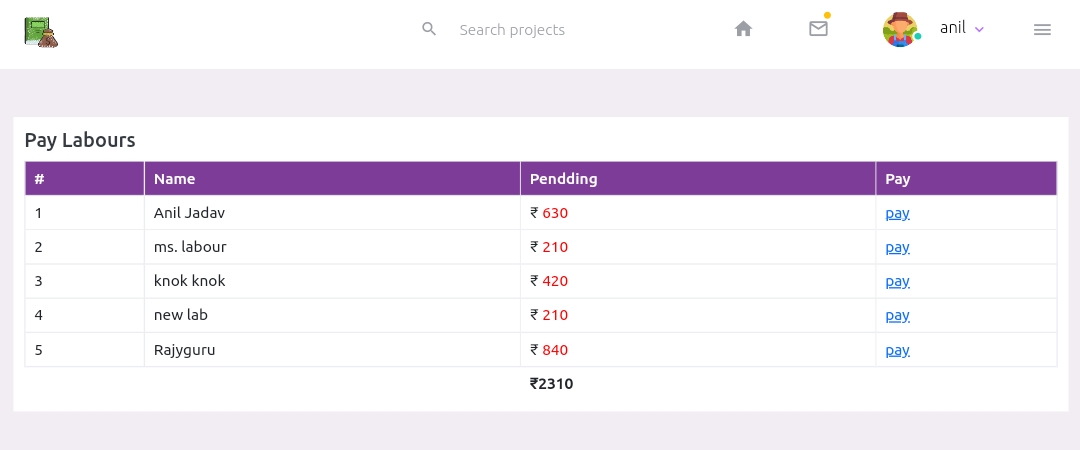


* Payment

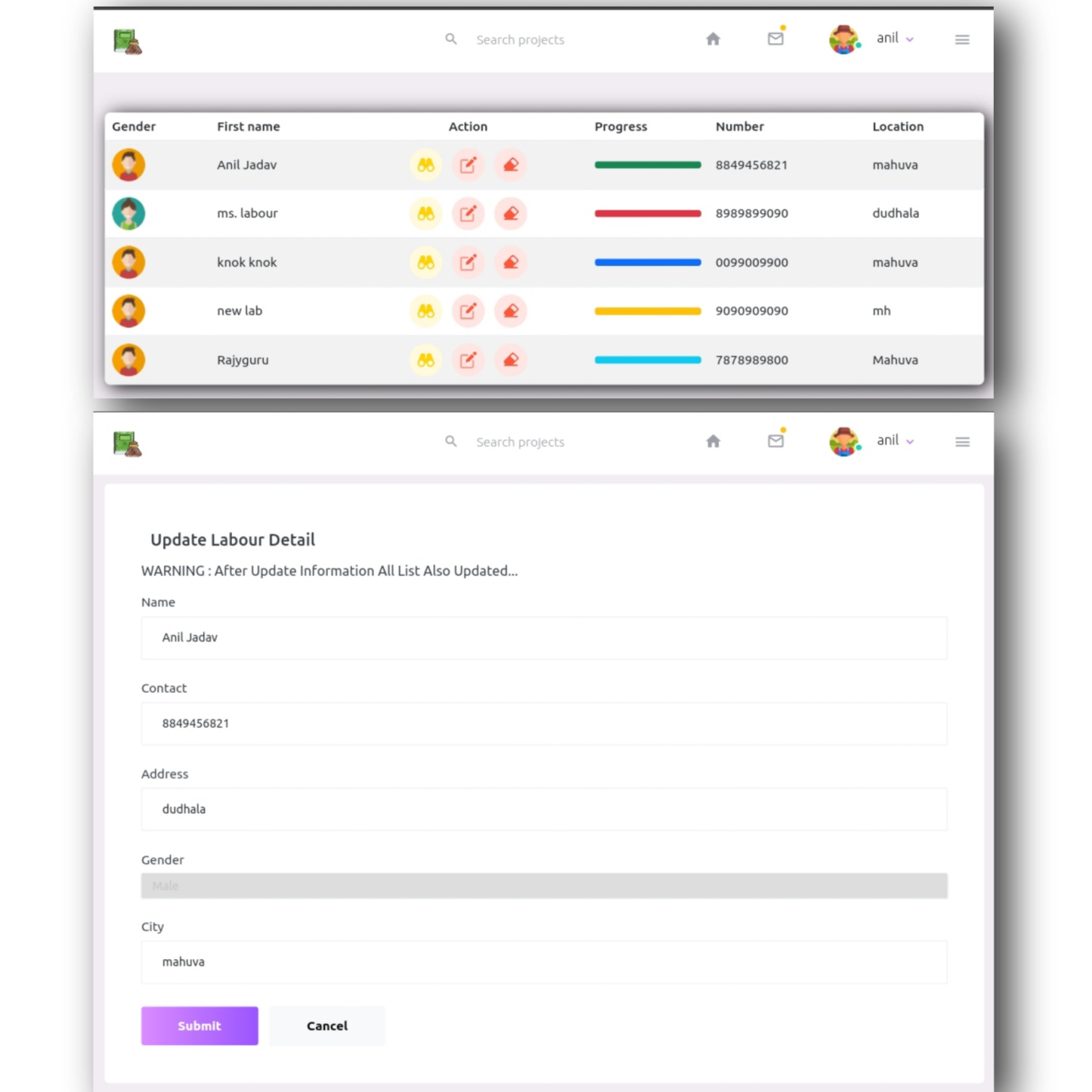


* Transactions page

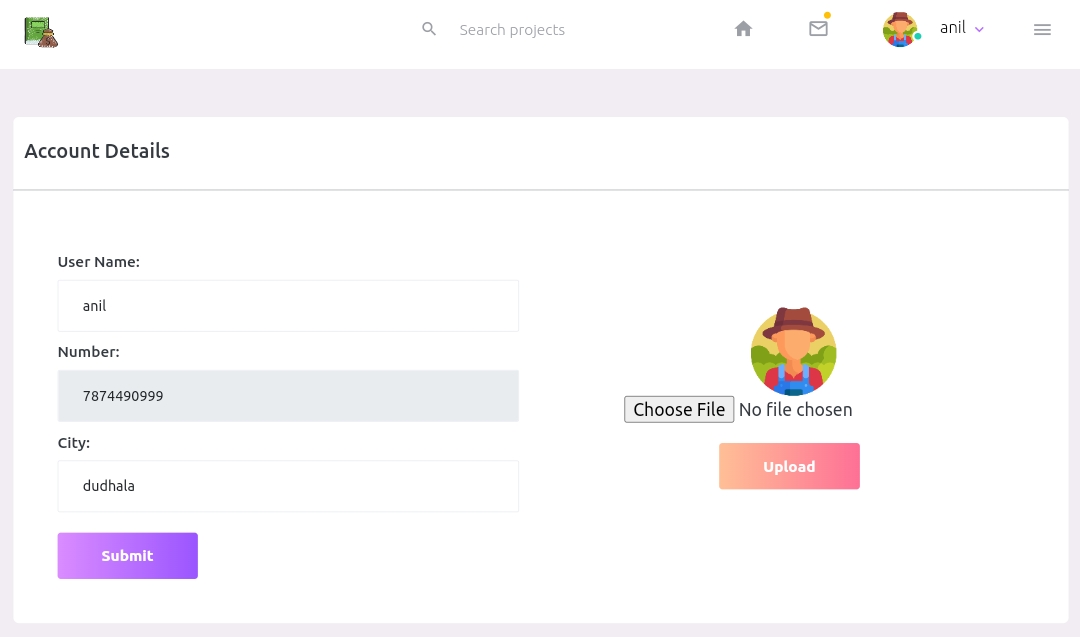


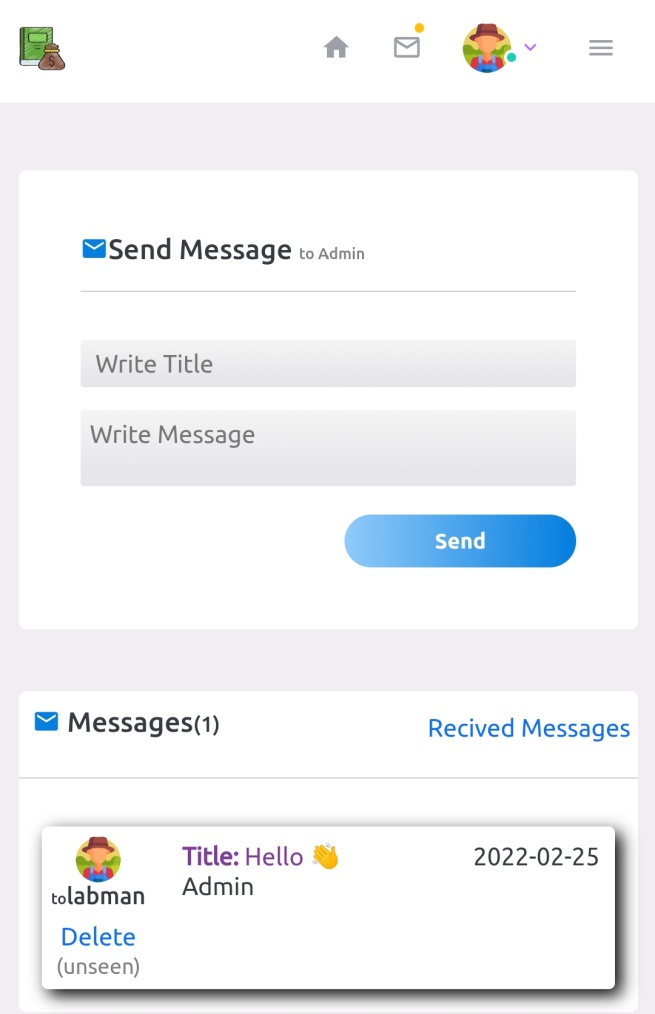


* View Labours & Update labours



Account Page :



Messages Page :

**• CHAPTER 4:** Testing & Implementation

**4.1:Testing**

**Introduction**

Software testing is a critical element of software quality assurance and represents the ultimate view of specification, design and coding.

When a system is developed, it is hoped that it works properly. In practice, however, some errors always occur. The main purpose of testing an information system is to find the errors and correct them. A successful test is one finds an error.

**Objectives of Testing:**

* To ensure that during operation the system will perform as per specifications.
* To make sure that the system meets user’s requirements during operation.
* To verify that the controls incorporated in the system function as intended.
* To see that when correct inputs are fed to the system the outputs are correct.
* To make sure that during operations, incorrect input, processing and outputs will be detected.
* Testing is a process of executing a program with the intent of finding an error.

**Test Procedure:**

* **Unit Testing**

All the modules included in the system were subjected to unit testing.

In case of input data, testing was done to see as to what the results would be if wrong data were given and in adverse conditions.

* **Integration Testing**

All the modules of the system were integrated together and tested Again for proper functionality. It is tested whether all parameters that is Filled in the form are passed to database for processing. Also check Whether these values properly update the respective databases when passed to them.

* **System Testing**

System testing is performing against requirement specification. This stage is intended to find defects that can be exposed only by testing the entire system.

* **Acceptance Testing**

The project guide tested module what are given to see that it satisfies the necessary guidelines and user friendliness.

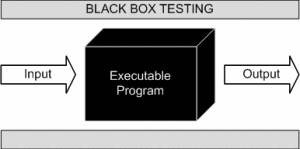
* **Installation Testing**

Once the system was ready, project guide tried installing the system and thus the installation testing was performed.

**Black box testing:**

Testing, either functional or non-functional, without reference to the internal structure of the component or system.

Black Box, also known as Behavioural Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.



This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see.

This method attempts to find errors in the following categories:

• Incorrect or missing functions.

• Interface errors.

• Errors in data structures or external database access.

• Behaviour or performance errors.

• Initialization and termination errors.

**White box testing:**

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

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

**Advantages**

* Testing can be commenced at an earlier stage. One need not wait for the GUI to be available.
* Testing is more thorough, with the possibility of covering most paths.

**Disadvantages**

* Since tests can be very complex, highly skilled resources are required, with a thorough knowledge of programming and implementation.
* Test script maintenance can be a burden if the implementation changes too frequently.
* Since this method of testing is closely tied to the application being tested, tools to cater to every kind of implementation/platform may not be readily available.

**Grey box testing:**

Grey Box testing is testing technique performed with limited information about the internal functionality of the system. Grey Box testers have access to the detailed design documents along with information about requirements.

Grey Box tests are generated based on the state-based models, UML Diagrams or architecture diagrams of the target system.



**Grey-box testing Techniques:**

* Regression testing
* Pattern Testing
* Orthogonal array testing
* Matrix testing

**Benefits:**

Grey-box testing provides combined benefits of both white-box and black-box testing.

It is based on functional specification, UML Diagrams, Database Diagrams or architectural view.

Grey-box tester handles can design complex test scenario more intelligently.

The added advantage of grey-box testing is that it maintains the boundary between independent testers and developers.

**Drawbacks:**

In grey-box testing, complete white box testing cannot be done due to inaccessible source code/binaries.

It is difficult to associate defects when we perform Grey-box testing for a distributed system.

**4.2 Test Cases**

* This web application has secured using session
* Without login no one can access the site data or visit dashboard
* Without login user can only visit 3 pages

1. Home page
2. Login page
3. Registration page

# Testing Approach

## System Testing

* System testing is the process of evaluation system to detect differences

Between given input and expected output. Also to assess the feature of a system .

* Testing assesses the quality of the product. System testing is a process that should be done during the development process.
* other words, System testing is a verification and validation process.

## Verification

* Verification is the process to make sure the product satisfies the conditions

Imposed at the start of the development phase .

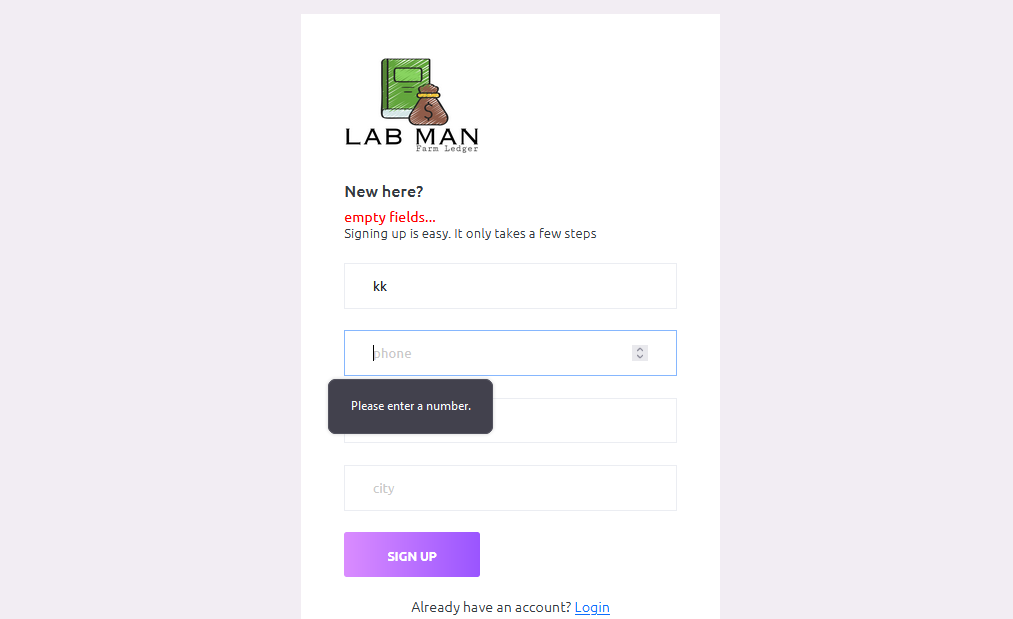
* In other words, to make sure the product behaves the way we want it to.

## Validation

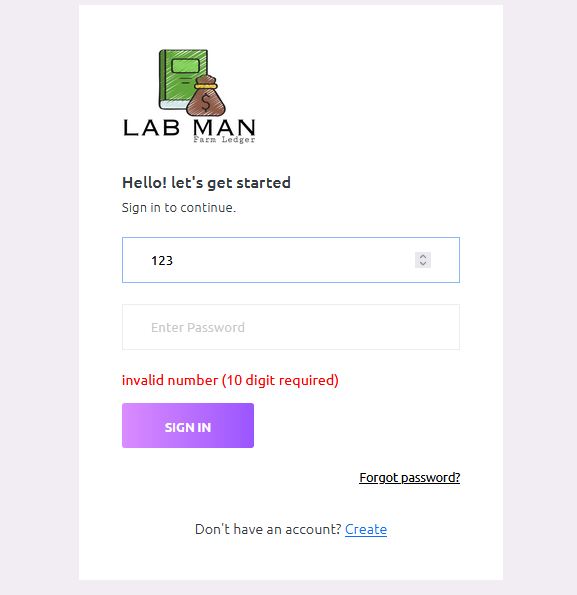
* Validation is the process to make sure the product satisfies the specified

Requirements at the end of the development phase.

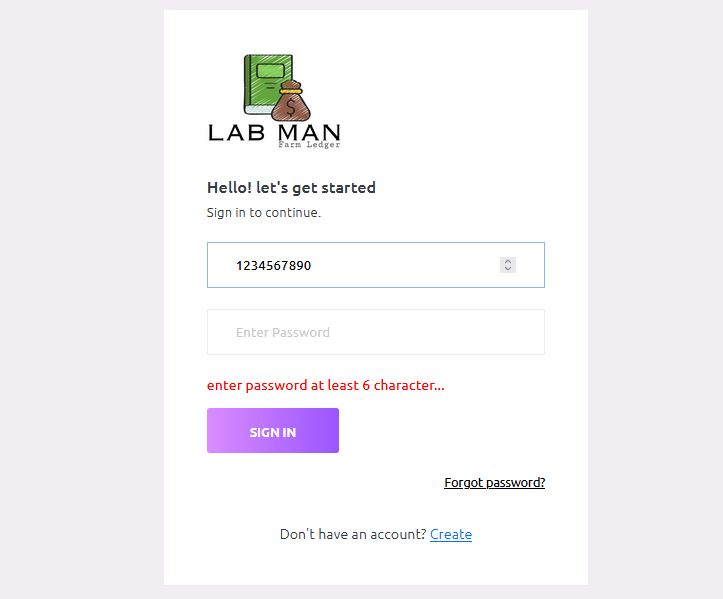
* In other words, to make sure the product is built as per customer requirements.
* **User Signup / registration page:**
* **Positive Test Case:**
* When a user enters correct detail than it will redirect next page.
* **Negative Test Case:**
* When user leave any fields empty (client side validation)
* Required field prompt shows this field is required



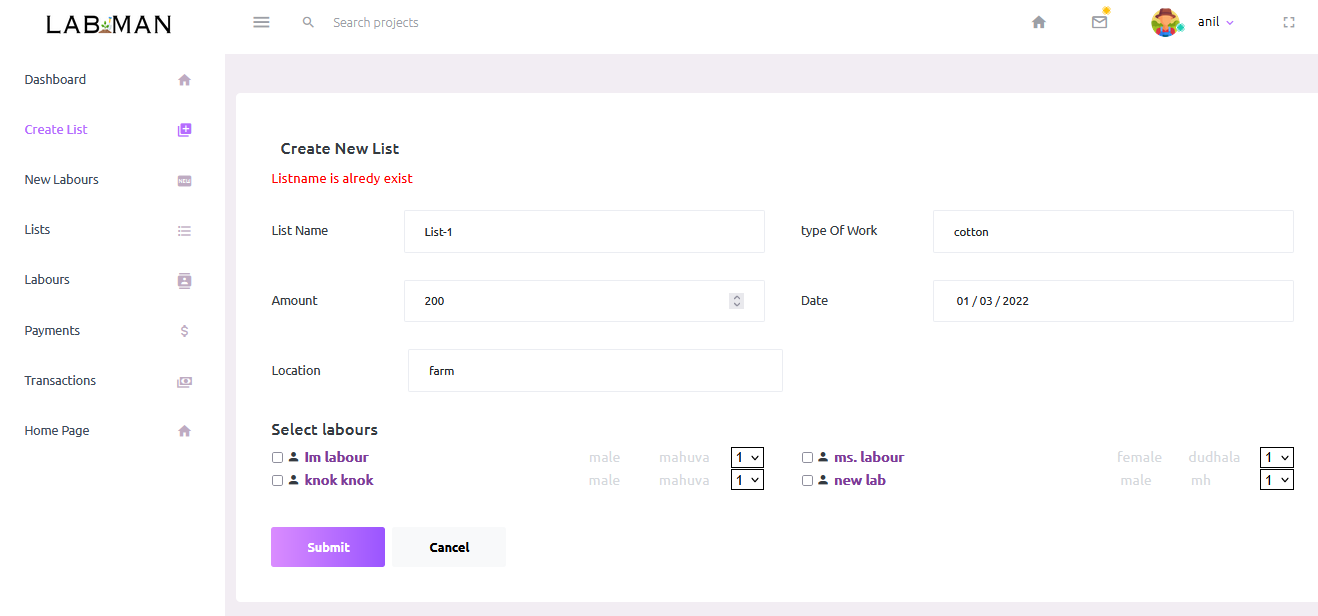
* Number must be 10 digit
* Using client side validation number must be in length of 10 digit
* In case any reason (modification in html code) then sever side validation is also check data and throw an error
* **User Login / Sign In page:**
* **Positive Test Case:**
* When a user enters correct detail than it will redirect to Dashbord or Home Page.
* **Negative Test Case:**
* When user leave any fields empty (number or password)
* Required field prompt shows this field is required
* Number must be 10 digit



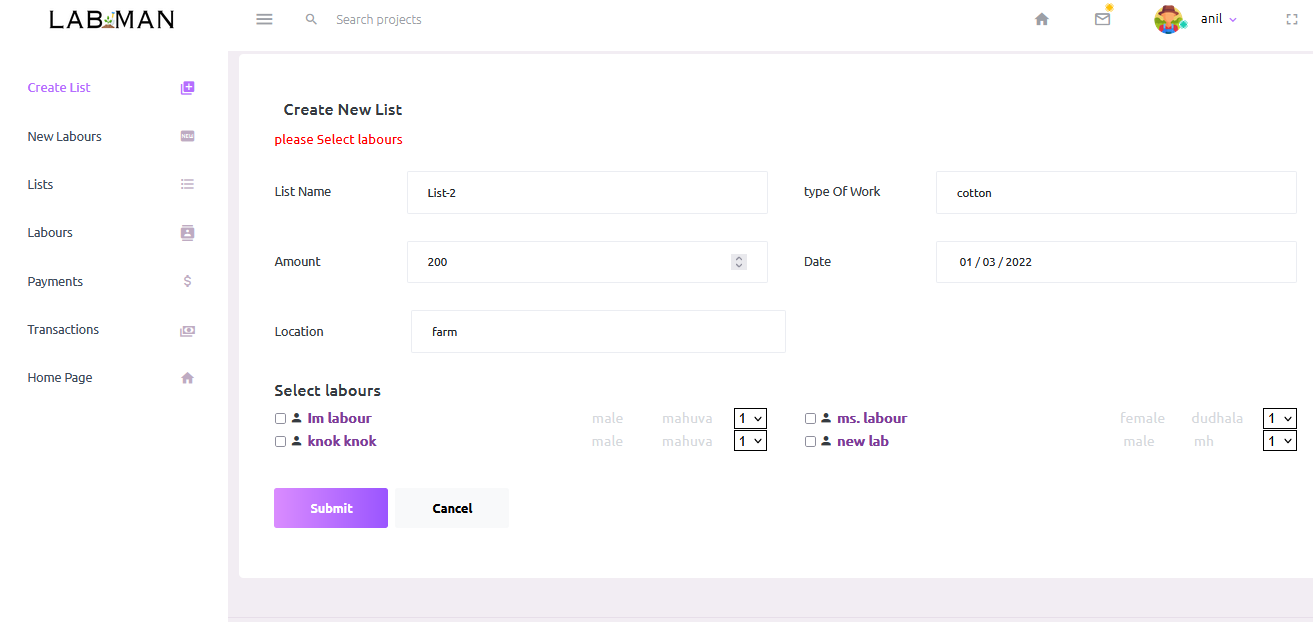
* Using client side validation number must be in length of 10 digit
* If user enter password less then 5 charchter
* Sever side and client side check the length if its not satisfied then show an error



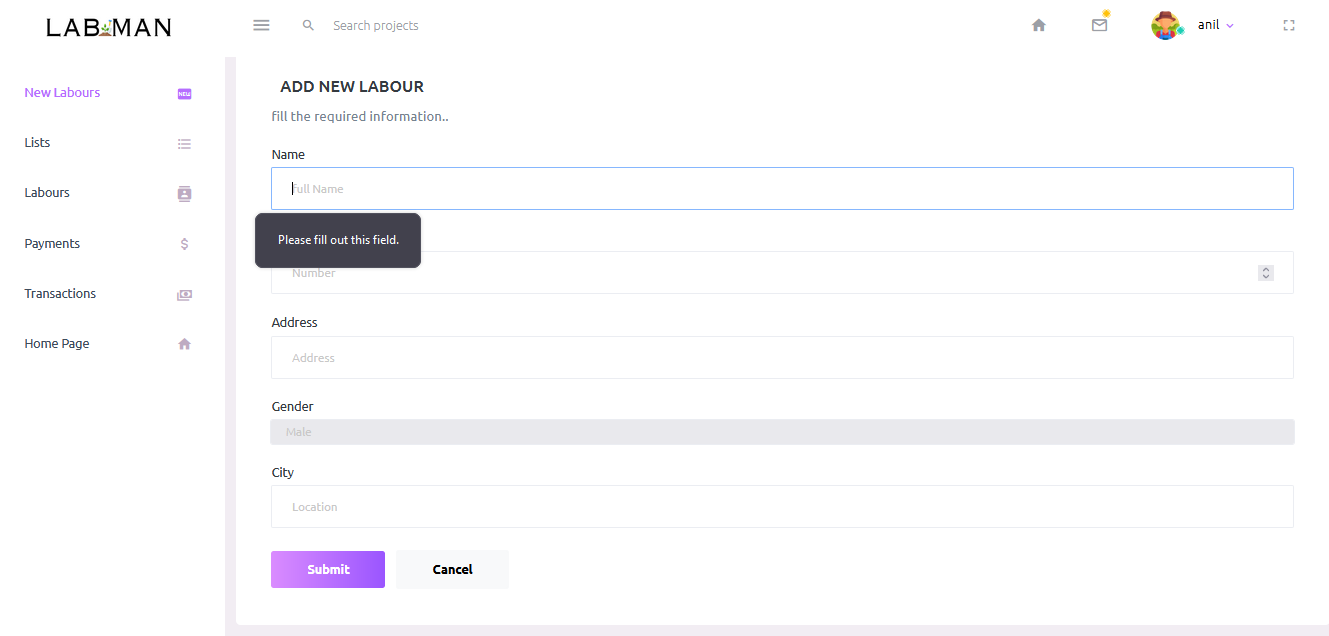
* After all checks password and number is match in database then user can access the dashboard
* In case any reason (modification in html code) then sever side validation is also check data and throw an error
* **List Create page:**
* **Positive Test Case:**
* When a user enters correct detail than List is created.
* **Negative Test Case:**
* When user leave any fields empty
* Required field prompt shows this field is required (using html required filed tag)
* If user enter same list name (existing list name)



* If in the new list no labours selected
* If no labour is selected then site throw an error please select at least one labour to create list.



* **Add Labours page:**
* **Positive Test Case:**
* When a user enters correct detail then labour is added .
* **Negative Test Case:**
* When user leave any fields empty
* Required field prompt shows this field is required (using html required filed tag)



* If user enter same labour name (duplicate labour)
* Then it will throw an duplicate labour error “please enter another name”.
* Also all data is check on server before insertion.
* Number is valid , name not empty , date is valid , valid labour etc..

## Implementation Approaches

* There are a variety of options that a project manager could consider when implementing a solution. There are Advantages and Dis-Advantages to each type, and the choice usually depends on the client organizational setup and the complexity of the solution to be implemented.
* These implementation choices available to a project manager are:
* Parallel Implementation
* Phased Implementation
* Crash Implementation

1. **Parallel Implementation:**

* A parallel Implementation or approach implies that a new solution is implemented parallel to the current operating system in use. Those who are using the system will not see major downtime once it is implemented. The trick here is to implement the system.
* Once the new solution is tested and up and running, it is “switched” on and the older version is “switched” off.
* The advantages with a parallel implementation include:
* Less disruption to the business
* No loss of business if the new system suddenly fails.

1. **Phased Implementation:**

* Sometimes trying to implement a solution all at once is not feasible because many clients have essential operations that run during normal working hours and cannot afford the luxury of having their entire operation close down for a lengthy period of time.
* Often, clients have front office staffs that attend to these operations (such as Call centers, Help Desks, etc.), and they work in 24-hour shifts. This is why many clients approve of a phased implementation approach, and the project team must ensure that the phased implementation is possible.
* This approach involves implementing the solution to a certain number of users and then rolling them onto the new solution, while the rest of the users are rolled out in a similar fashion until the entire solution is rolled out within the client environment.
* The phase approach works well because:
* There is minimal disruption to the client’s operation.
* Problems are resolved quickly.
* The phased approach could also be used if there is more than one department. The project manager could decide that implementing the solution in one department at a time could be more reliable than trying to roll out all departments at the same time.

1. **Crash Implementation:**

* Careful planning needs to take place when considering a crash (also known as full-blown) implementation. It takes an incredible amount of planning to ensure no problems arise. In fact, with this type of implementation, the necessary contingencies need to be prepared and reviewed well in advance of the actual implementation, in order to minimize any potential failure.
* The necessary IT support staffs also need to be available on the chosen implementation period. A full-blown implementation should be scheduled to take place over a slow period, such as a Holiday or Weekend.

**CHAPTER 5:** Conclusion

**5.1 Conclusion**

In this farm labour management system we provide a digital platform for farmer that makes farmers life little bit easier.

Farmer can’t remember every labours work days their payments history or paid and un-paid amount of each week or month. This tool is helpful for farmers in their daily hard working life.

This system will help to manage labours attendance and payment. Farmer can easily track labours with they count of work days in moth vise or year or any date vise.

We want to make farm labours ledger digital where farmer can make they ledger anywhere without any pen or book.

**5.2 Limitations**

* Farmers are not educated and non-technical persons because of this its hard to educate or provide information how to use this web application.
* Because of website farmers need to remember the web address.
* Because this web app in English, Language is biggest limitation of this tool , not every farmer know English they understand they local languages
* Famers don’t know about IT that’s why he don’t want to use Applications or websites.

**5.2 Future Scope Of System**

* Multiple language support for every state
* Make an android app for easy task
* More graphical interface
* Report and graph analytics
* Report export to pdf
* Whatsapp share
* Labours login
* Online Transaction
* Otp verification
* Email notification

## 5.2 Bibliography

* [https://www.google.com/](https://www.google.com/#spf=1622394388356)
* [www.w3schools.com](https://www.google.com/search?q=www.w3schools.com&source=hp&ei=G8azYLOaPIWf9QPqzIjQAw&iflsig=AINFCbYAAAAAYLPULGGERhxtAvPN9bNN_C2AA8jPzz3X&oq=www.w3&gs_lcp=Cgdnd3Mtd2l6EAMYATICCAAyBQgAELEDMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIADICCAA6CAgAEOoCEI8BOggIABCxAxCDAToICC4QxwEQowI6CggAELEDEIMBEAo6BAgAEAo6BwgAELEDEApQlg1YuCxglDpoAnAAeACAAcUCiAGHDpIBBzAuMS4zLjOYAQCgAQGqAQdnd3Mtd2l6sAEK&sclient=gws-wiz)
* [www. stackoverflow.com](https://stackoverflow.com/)
* [www.youtube.com](https://www.google.com/search?q=www.youtube.com&ei=tMazYLDqFtPJrQHB9LnQCw&oq=www.you&gs_lcp=Cgdnd3Mtd2l6EAMYADILCAAQsQMQgwEQkQIyCAgAELEDEIMBMggIABCxAxCDATIFCAAQsQMyCAgAELEDEIMBMggIABCxAxCDATIICAAQsQMQgwEyBQgAELEDMgUIABCxAzIICAAQsQMQgwE6BQgAELADOgIIADoFCAAQkQI6BAgAEANQurwBWIDQAWDx1wFoAXAAeACAAYUDiAHGFpIBBzAuMi41LjSYAQCgAQGqAQdnd3Mtd2l6yAEKwAEB&sclient=gws-wiz)