PROJECT MANAGEMENT SYSTEM

Project Report SubmittedBy

MRUDUL A THAKADIYEL

Reg. No.:AJC17MCA-I040

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS
(INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**PROJECT MANAGEMENT SYSTEM**" is the bonafide work of **MRUDUL A THAKADIYEL (Reg.No: AJC17MCA-I040)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Mr. Jinson Devis
Internal Guide

Ms. Meera Rose Mathew
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **“PROJECT MANAGEMENT SYSTEM”** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date:	**MRUDUL A THAKADIYEL**
KANJIRAPPALLY	**Reg. No: AJC17MCA-I040**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Mr. Jinson Devis** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

MRUDUL A THAKADIYEL

ABSTRACT

The Project Management System addresses the management of software projects. It provides the framework for organizing and managing resources in such a way that these resources deliver all the work required to complete a software project within defined scope, time and cost constraints.

The proposed system enables the Manager to assign task and monitor the work progress of his/her team members. This system will also be able to provide the list of employees who are involved in particular project for particular module and time involved in development work. System will also be able to categorize the particular projects upon different modules and have look on them as and when required. User can add different tasks and split up the work. Users can also add comments and Checklist to them for better clarity and modularity for the given task. This system also provides a full-fledged chat system and file sharing for smooth communication and better collaboration.

CONTENT

List of Abbreviation

IDE	Integrated Development Environment
HTML	Hyper Text Markup Language.
CSS	Cascading Style Sheet
SQL	Structured Query Language
UML	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The Project Management System addresses the management of software projects. It provides the framework for organizing and managing resources in such a way that these resources deliver all the work required to complete a software project within defined scope, time and cost constraints. It looks for various phases which are involved under entire development work such as analysis, system design, coding, testing and maintenance work etc. Present system always facing time problem during their development work by which there exist extra investment. All the works were maintained on using manual file by which it creates difficulties for the organization to prepare reports on projects. It's difficult by the managers to keep track on projects which are working in different teams and take immediate action to complete the projects within stipulated time.

1.2 PROJECT SPECIFICATION

The proposed system enables the Manager to assign task and monitor the work progress of his/her team members. This system will also be able to provide the list of employees who are involved in particular project for particular module and time involved in development work. System will also be able to categorize the particular projects upon different modules and have look on them as and when required. This system also provides a full-fledged chat system and file sharing for smooth communication and better collaboration.

The system includes mainly 3 users. They are:

1. Manager

Managers can register and they can create new projects and manage them. They can create and manage teams and assign projects to different teams. They can assign tasks and track activities of team members. They can add checklist inside each task. They can manage all attachments added to the projects. They could also schedule meetings and other events.

2. Team Members

Team Members can register into the site with or without an invitation from their manager. They can view and track their progress. Activities related to a project can be visible in the activity tab. They can update the current status of that task and can also check completed checklist item.

They can also add comments, attach files with the tasks. They can also view their assigned team.

3. Admin

Admin must have a login into this system. He has the overall control of the system. He can add and manage all users of the system. He could also add new user roles and manage them. He also manages user permission.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analysed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analysing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM OF PROJECT MANAGEMENT SYSTEM

In the existing system always facing time problem during their development work by which there exist extra investment. All the works were maintained on using manual file by which it creates difficulties for the organization to prepare reports on projects. It’s difficult by the managers to keep track on projects which are working in different teams and take immediate action to complete the projects within stipulated time.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Time consuming.
- Consumes large volume of paper work.
- Less convenient in managing project details
- Project often delayed with no progress visibility.
- Human effort is needed

2.4 PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. It is necessary to have a system that is more user friendly. The proposed system enables the Managers to create new project and manages them. They can assign task and monitor the work progress of his/her team members. They can create and manage teams and assign projects to different teams. This system will also able to provide the list of employees who are involved in particular project for particular module and time involved in development work. System will also able to categorize the particular projects upon different modules and have look on them as and when required.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- User friendliness and interactive.
- Minimum time required.
- Security of data.
- Minimum time needed for the various processing.
- Greater efficiency.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development

3.1.1 Technical Feasibility

Technical feasibility is concerned with the availability of hardware and software required for the development of the system, to see compatibility and maturity of the technology proposed to be used and to see the availability of the required technical manpower to develop the system.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

After the study we came to conclusion that we proceed further with the tools and development environment chosen by us. This was important in our case as we were working on two various phases of the department that will need to be integrated in future to make an extended system. So, it's clear that the project PMS is Technically feasible.

3.1.2 Resource feasibility

This aspect looks at the resources that are required to complete the project and whether the amount of available resources is sufficient to complete the project effectively. Resources that are required for the PMS project includes: Programming device (Laptop), Hosting space (freely available), Programming tools (freely available), Programming individuals. So, it's clear that the project PMS has the required resource feasibility.

3.1.3 Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

Some of the aspects to be considered for determining the economic feasibility are:

- Are the operational and maintenance costs of the system manageable and affordable?
- Can the economic expenses of the prevailing system be reduced later after development of the system?
- Can the existing investment support the project development?
- Can the target group of your system afford your system?

Being a web application PMS will have an associated hosting cost. Since the system doesn't consist of any multimedia data transfer, bandwidth required for the operation of this application is very low. Bug fixes and maintaining tasks will have an associated cost. Beside the associated cost, there will be many benefits for the customers. Especially the extra effort that is associated with paper making and marking will be significantly reduced while the effort to create descriptive statistical reports will be eliminated. From these it's clear that the project PMS is Economic feasible.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3
RAM - 4 GB
Hard disk - 1 TB

3.2.2 Software Specification

Front End - HTML, CSS, SCSS
Backend - MySQL, PHP
Client on PC - Windows 10 and above.
Technologies used - JS, HTML5, AJAX, jQuery, PHP, CSS, SCSS

3.3 SOFTWARE DESCRIPTION

3.3.1 HTML

HTML (Hypertext Markup Language) is the most basic building block of the Web. HTML is a fairly simple language made up of elements, which can be applied to pieces of text to give them different meaning in a document, structure a document into logical sections, and embed content such as images and videos into a page. In the proposed system, HTML 5 is used to carry out the design part of the webpage.HTML validations are also used.

3.3.2 CSS

While HTML is used to define the structure and semantics of your content, CSS is used to style it and lay it out. For example, you can use CSS to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. In the proposed system CSS 3 is used. Animation using CSS is also used to give the website a unique look & feel

3.3.3 SCSS/SASS

Sass stands for Syntactically Awesome Stylesheet. Sass is an extension to CSS. It is a CSS pre-processor. It is completely compatible with all versions of CSS. Sass reduces repetition of CSS and therefore saves time

Stylesheets are getting larger, more complex, and harder to maintain. This is where a CSS pre-processor can help. Sass lets you use features that do not exist in CSS, like variables, nested rules, mixins, imports, inheritance, built-in functions, and other stuff.

3.3.4 Java Script

JavaScript was initially created to “make web pages alive”. The programs in this language are called scripts. They can be written right in a web page’s HTML and run automatically as the page loads. In the proposed system loading screen, different popup animations are implemented using Java Script.

3.3.5 jQuery

jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation. The jQuery library contains the following features:

HTML/DOM manipulation, CSS manipulation, HTML event methods, Effects and animations, AJAX

3.3.6 PHP

PHP is a server scripting language used for making dynamic web pages. That means PHP allows you to use scripts on a web server to produce a response customized for each client's (user's) request.

The proposed system developed in PHP 7 and is hosted in a local server (Local Host). All database interactions are carried out using PHP.

3.3.7 MySQL

MySQL is a database management system that allows you to manage relational databases. It is open-source software backed by Oracle. All data related with the system is stored and manages using MySQL database.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML stands for Unified Modelling Language. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modelling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design.

4.2.1 Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modelling Language), a standard notation forth modelling of real-world objects and systems

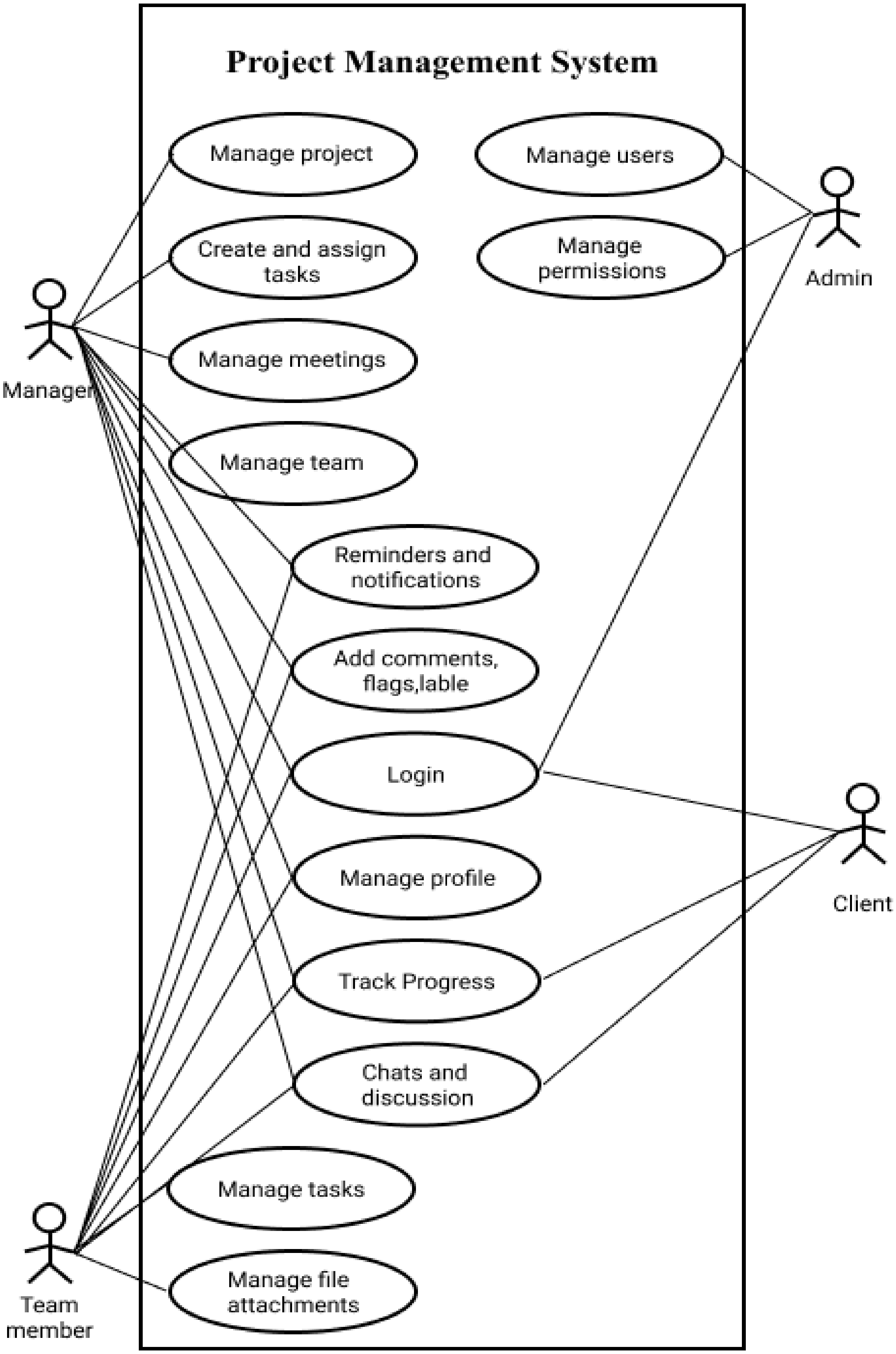
A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig 1: Use case diagram for Project Management System



4.2.2 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

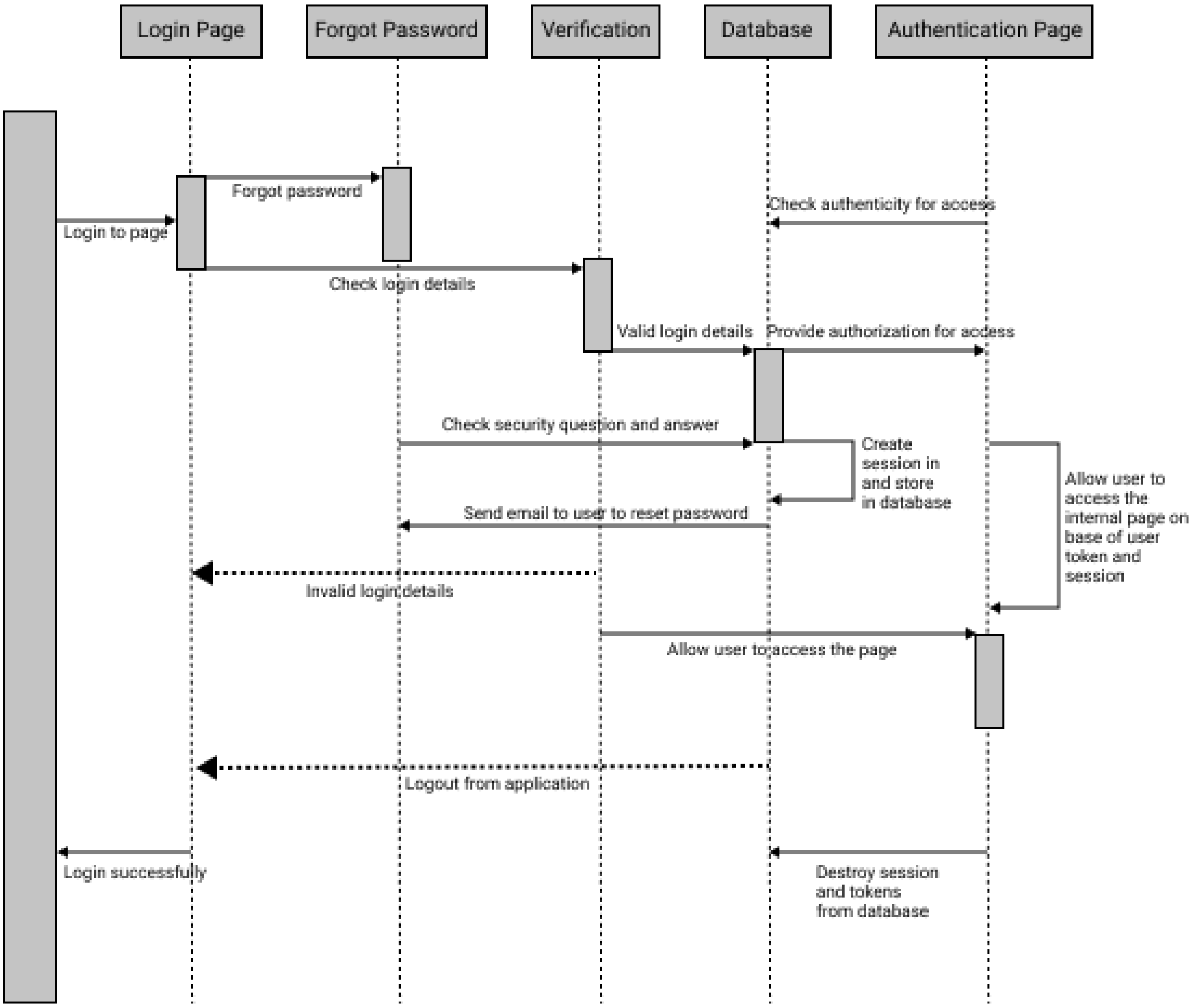
iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

iv. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

Fig 2: Sequence diagram for Project Management System


Login Page
Forgot Password
Verification
Database
Authentication Page
Forgot password
Login to page
Check authenticity for access
Check login details
Valid login details
Provide authorization for access
Check security question and answer
Create session in and store in database
Allow user to access the internal page on base of user token and session
Send email to user to reset password
Invalid login details
Allow user to access the page
Logout from application
Login successfully
Destroy session and tokens from database

4.2.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

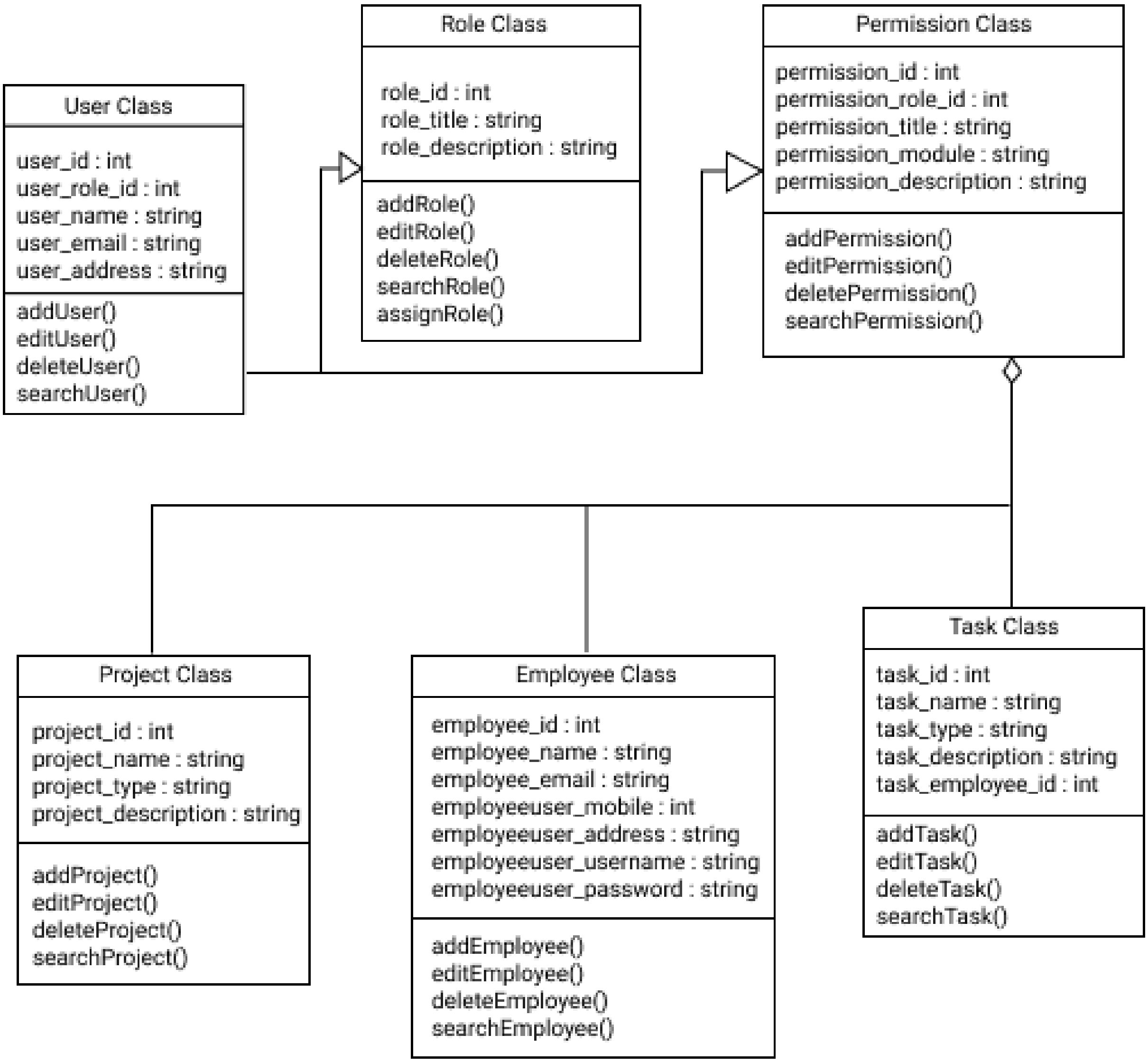
Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The following points should be remembered while drawing a class diagram −

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified
- For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. At the end of the drawing, it should be understandable to the developer/coder.

Fig 3: Class diagram for Project Management System

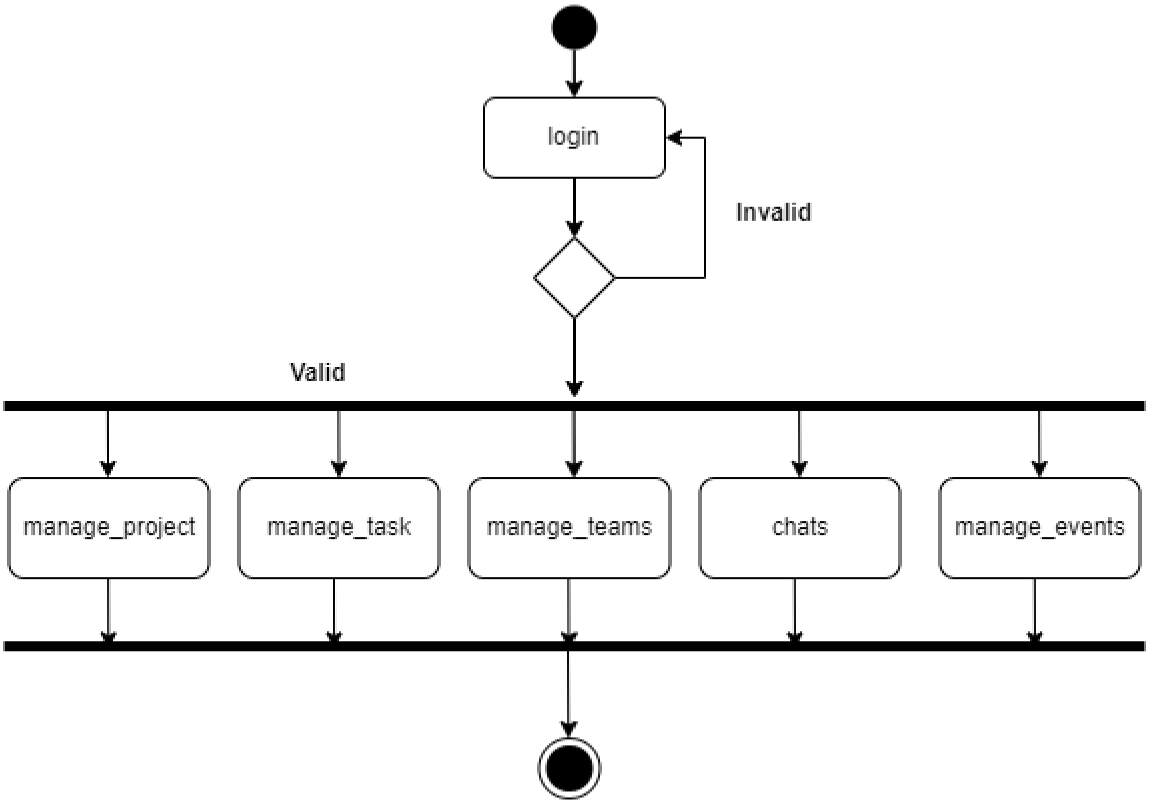


4.2.4 Activity Diagram

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system.

Fig 4: Activity diagram for Project Management System



4.2.5 Deployment Diagram

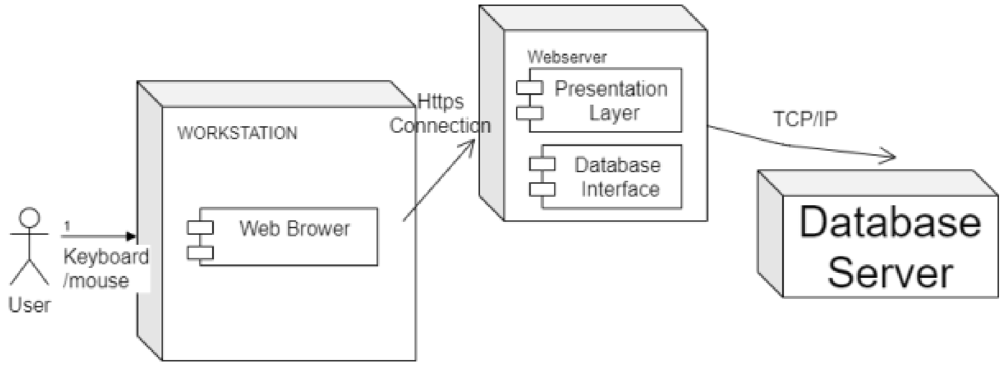
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

The purpose of deployment diagrams can be described as −

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.

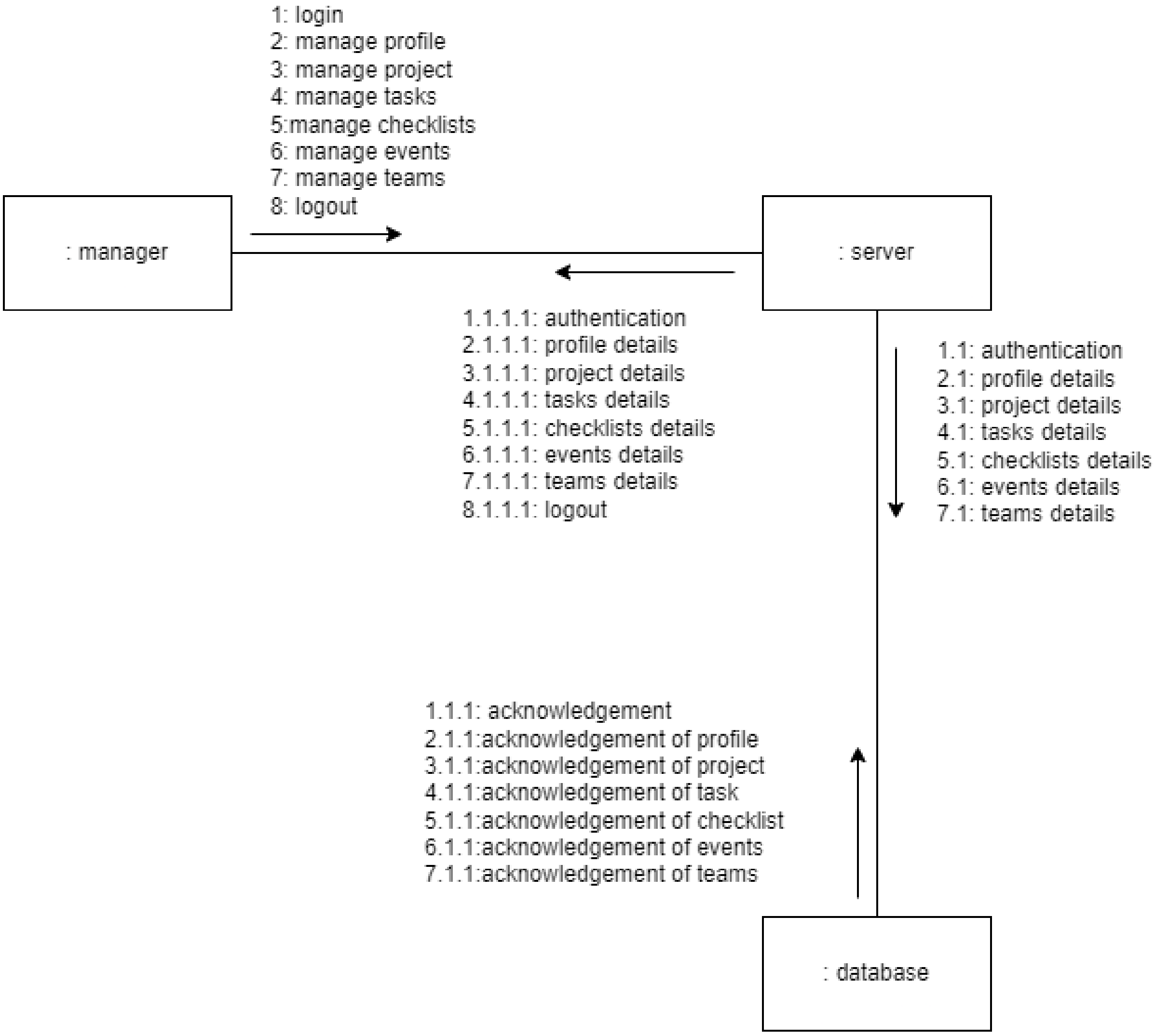
Fig 5: Deployment diagram for Project Management System



4.2.6 Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Fig 6: Collaboration diagram for Project Management System

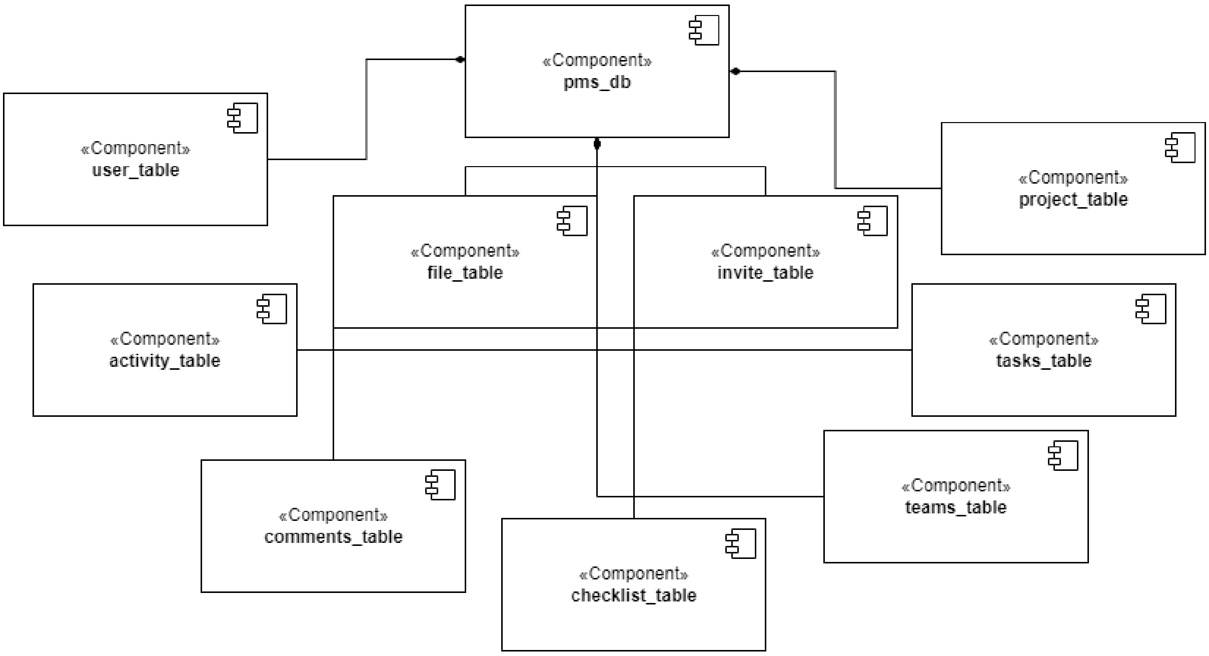


4.2.7 Component Diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behaviour is explained by the provided and required interfaces.

Fig 7: Component diagram for Project Management System



4.2.8 Object Diagram

The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams.

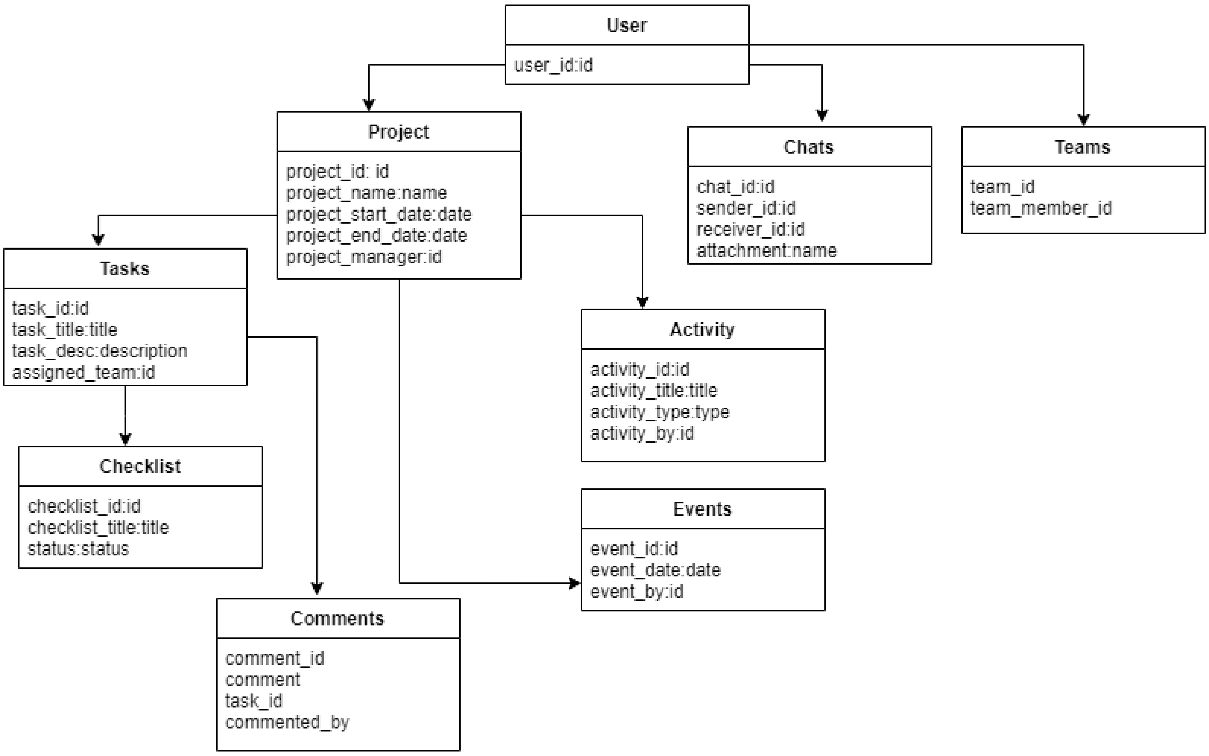
The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature.

It means the object diagram is closer to the actual system behaviour. The purpose is to capture the static view of a system at a particular moment.

The purpose of the object diagram can be summarized as −

- Forward and reverse engineering.
- Object relationships of a system
- Static view of an interaction.
- Understand object behaviour and their relationship from practical perspective

Fig 8: Object diagram for Project Management System



4.2.9 Statechart Diagrams

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

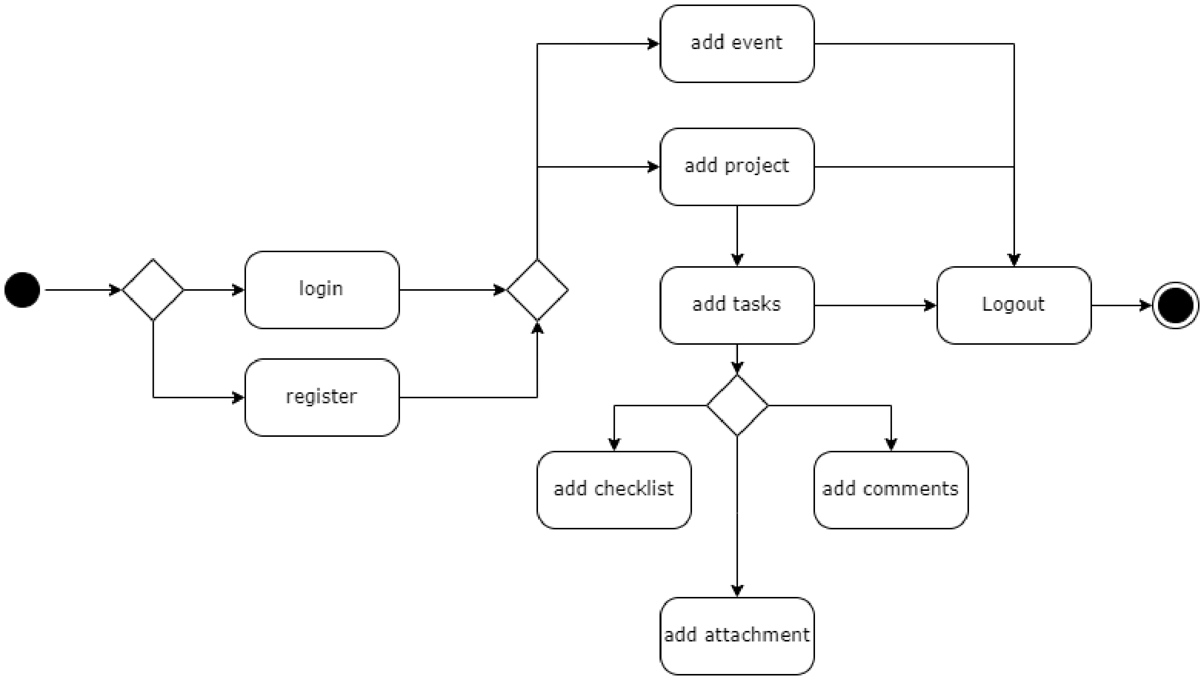
State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

Following are the main purposes of using State chart diagrams −

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.

- Define a state machine to model the states of an object.

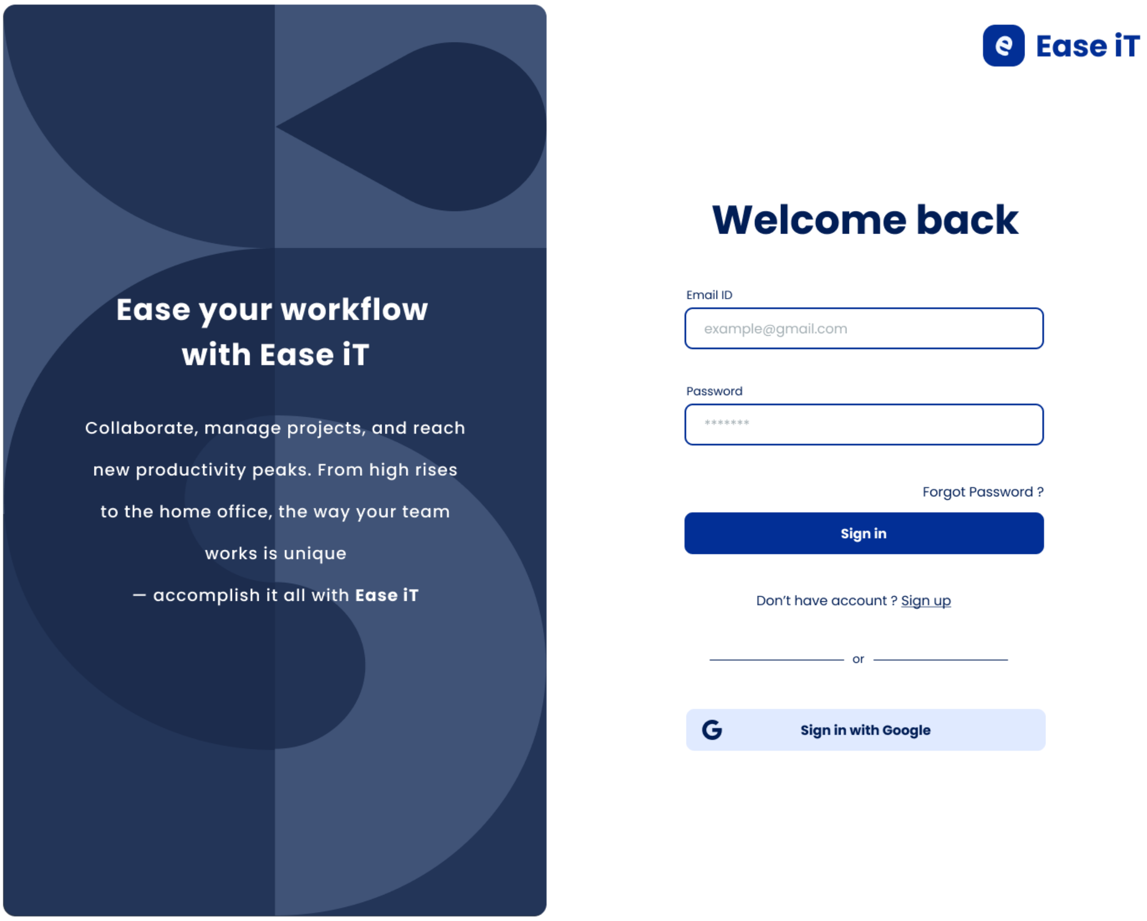
Fig 9: Statechart diagram for Project Management System



4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1 Design Prototype

Login Page



Ease your workflow with Ease iT

Collaborate, manage projects, and reach new productivity peaks. From high rises to the home office, the way your team works is unique

— accomplish it all with **Ease iT**

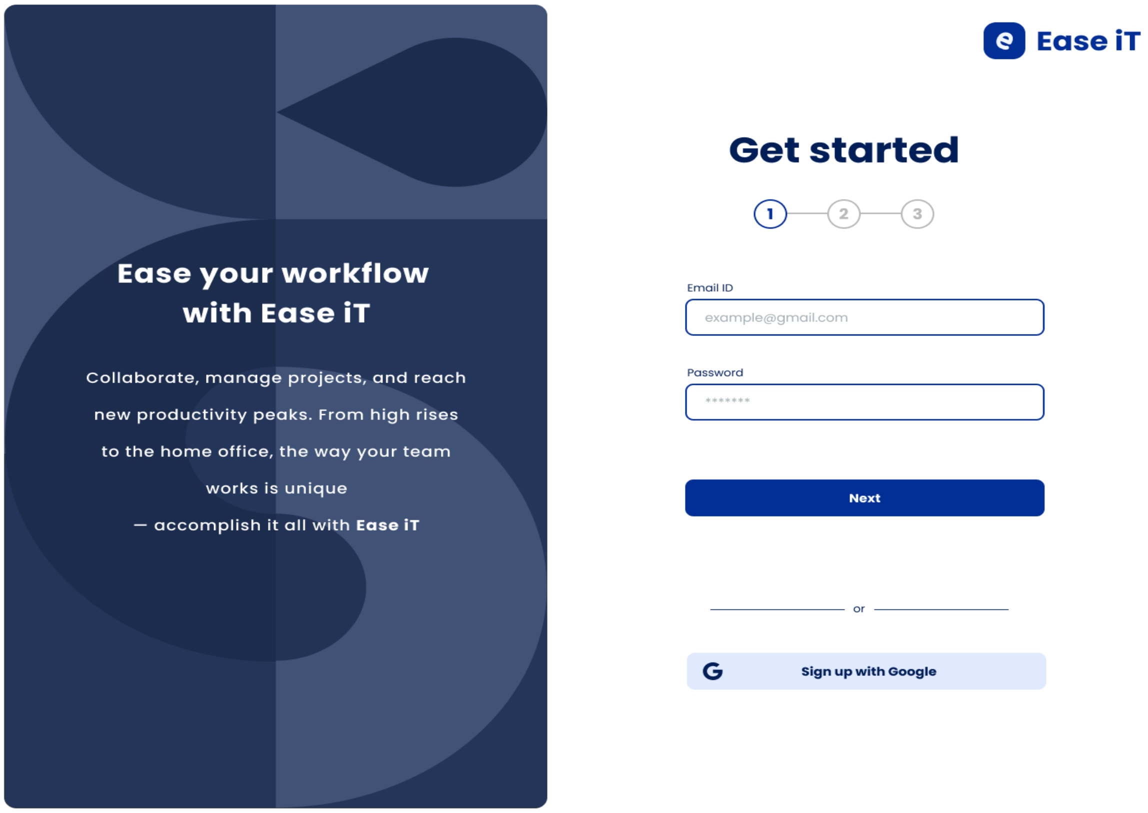
Welcome back

Email ID: example@gmail.com

Password: *******

Don't have account ? Sign up

Registration Pages



Ease iT

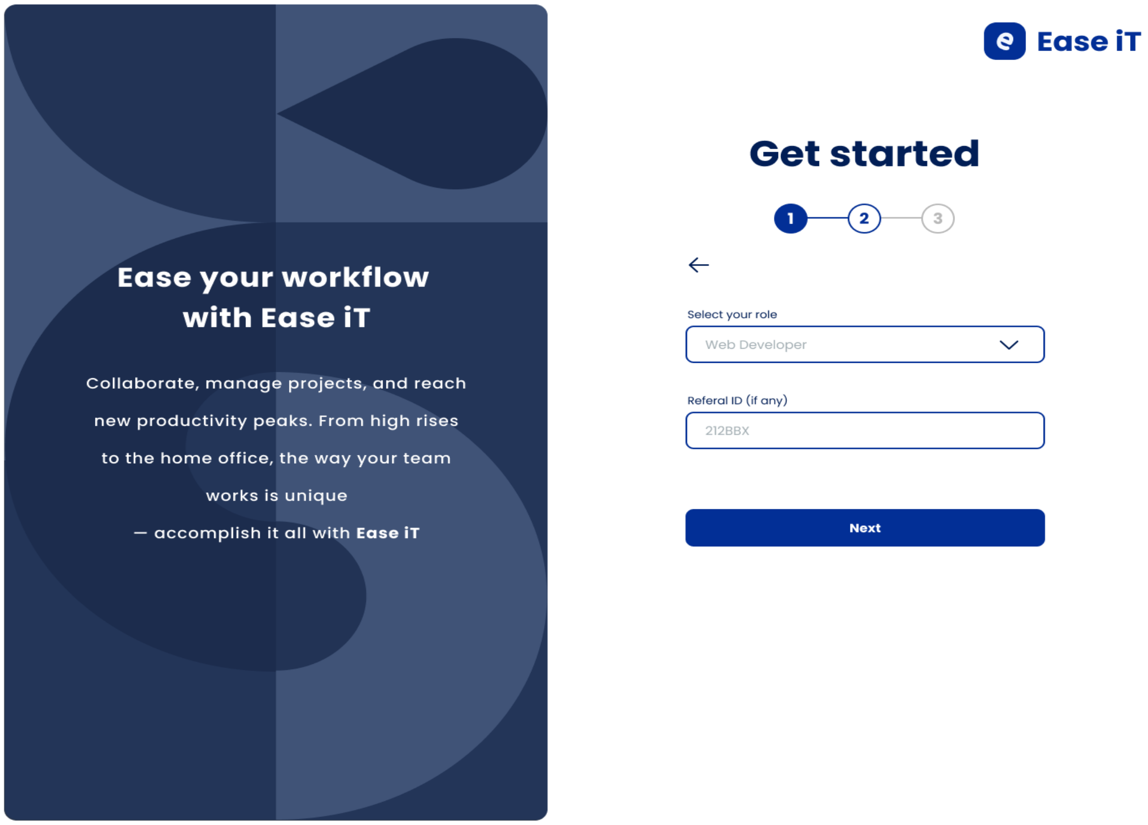
Ease your workflow with Ease iT

Collaborate, manage projects, and reach new productivity peaks. From high rises to the home office, the way your team works is unique — accomplish it all with **Ease iT**

Get started

Email ID: example@gmail.com

Password: *******



Ease iT

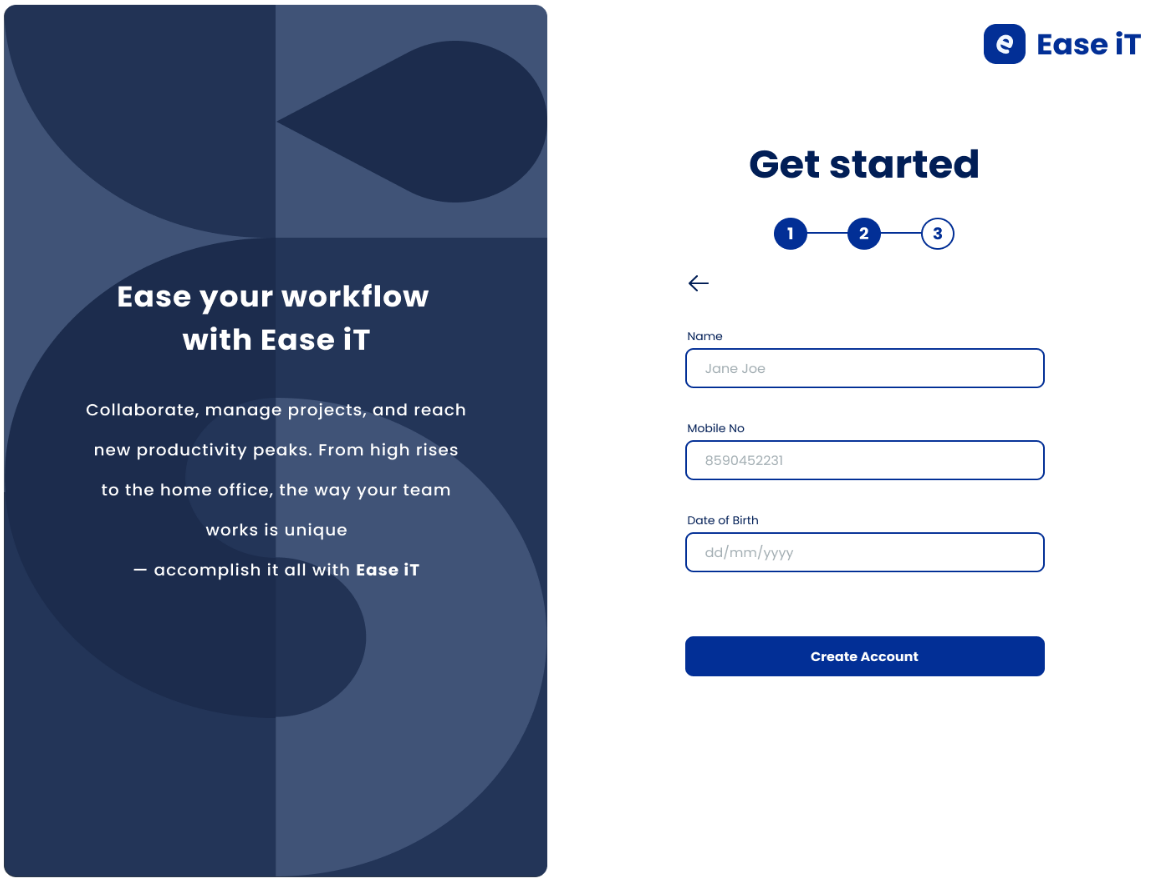
Ease your workflow with Ease iT

Collaborate, manage projects, and reach new productivity peaks. From high rises to the home office, the way your team works is unique — accomplish it all with **Ease iT**

Get started

Select your role: Web Developer

Referal ID (if any): 212BBX



Ease iT

Ease your workflow with Ease iT

Collaborate, manage projects, and reach new productivity peaks. From high rises to the home office, the way your team works is unique

— accomplish it all with **Ease iT**

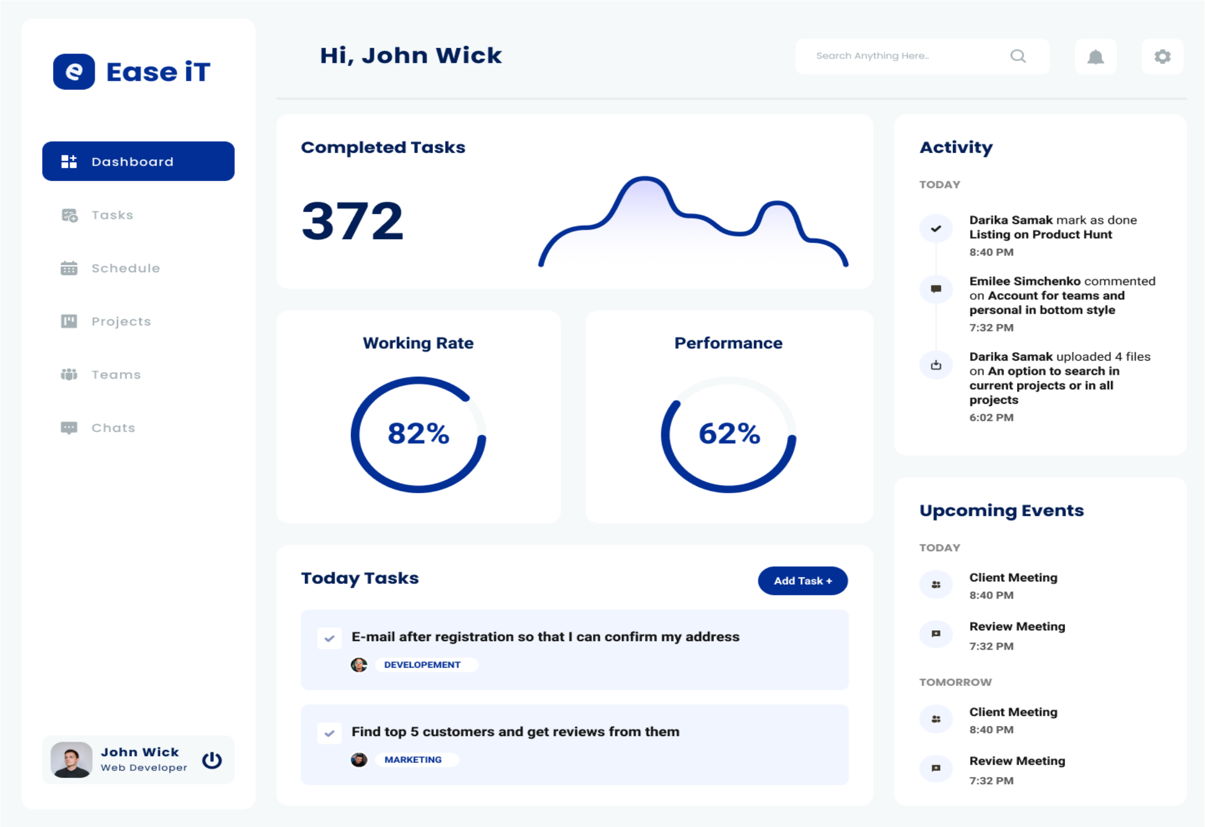
Get started

Name: Jane Joe

Mobile No: 8590452231

Date of Birth: dd/mm/yyyy

Dashboard



Ease iT

Hi, John Wick

Completed Tasks

372

Working Rate

82%

Performance

62%

Today Tasks

✓ E-mail after registration so that I can confirm my address

DEVELOPEMENT

✓ Find top 5 customers and get reviews from them

MARKETING

Activity

TODAY

Darika Samak mark as done **Listing on Product Hunt**
8:40 PM

Emilee Simchenko commented on **Account for teams and personal in bottom style**
7:32 PM

Darika Samak uploaded 4 files on **An option to search in current projects or in all projects**
6:02 PM

Upcoming Events

TODAY

Client Meeting
8:40 PM

Review Meeting
7:32 PM

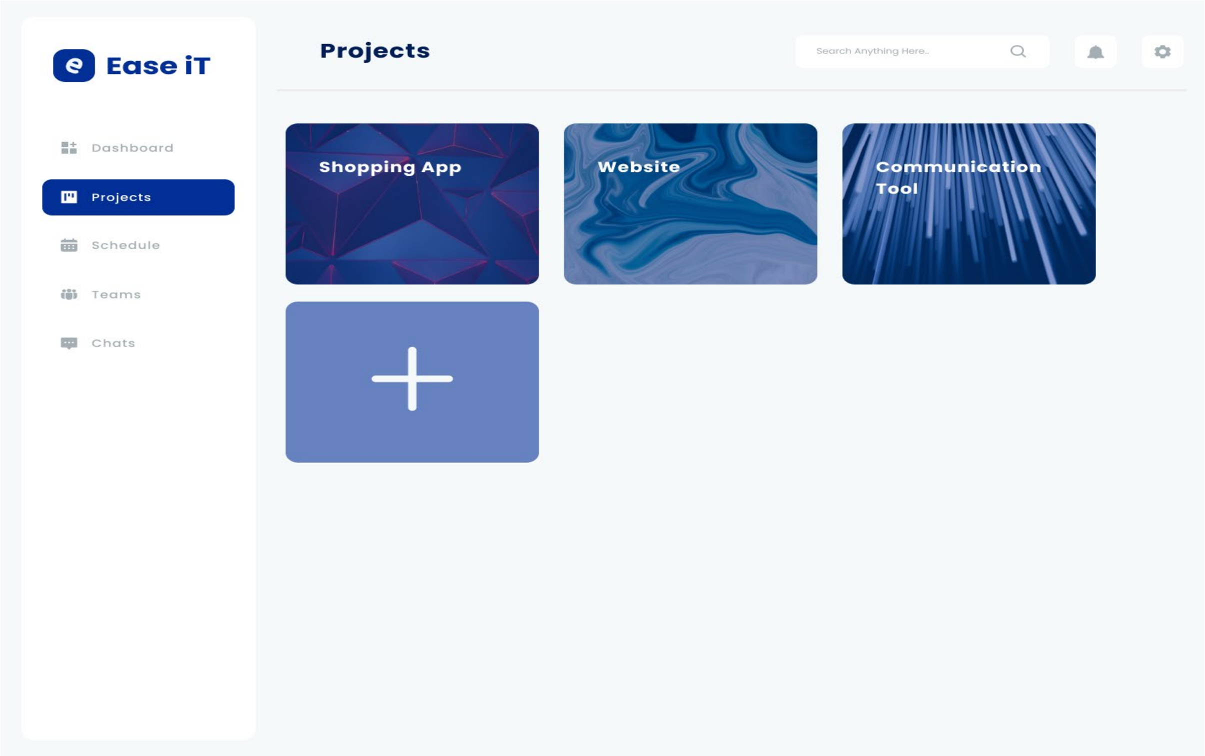
TOMORROW

Client Meeting
8:40 PM

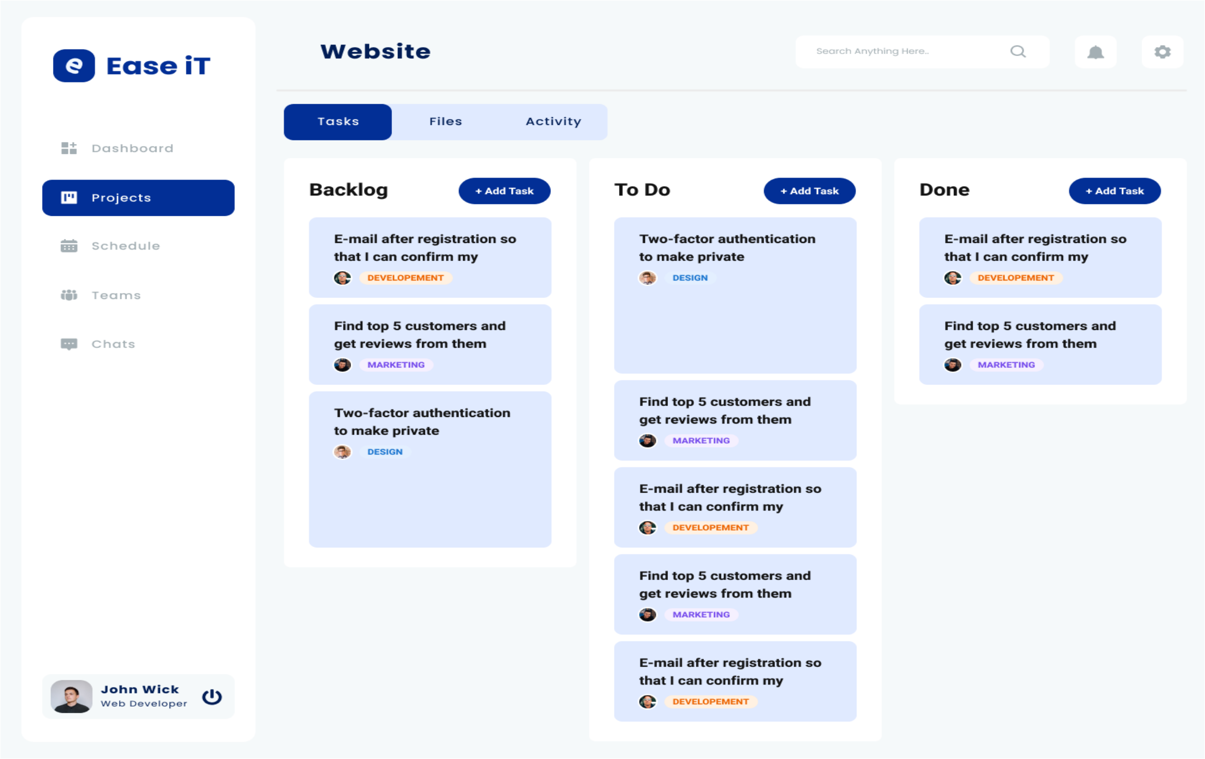
Review Meeting
7:32 PM

John Wick
Web Developer

Projects



Tasks



Backlog

E-mail after registration so that I can confirm my
DEVELOPEMENT

Find top 5 customers and get reviews from them
MARKETING

Two-factor authentication to make private
DESIGN

To Do

Two-factor authentication to make private
DESIGN

Find top 5 customers and get reviews from them
MARKETING

E-mail after registration so that I can confirm my
DEVELOPEMENT

Find top 5 customers and get reviews from them
MARKETING

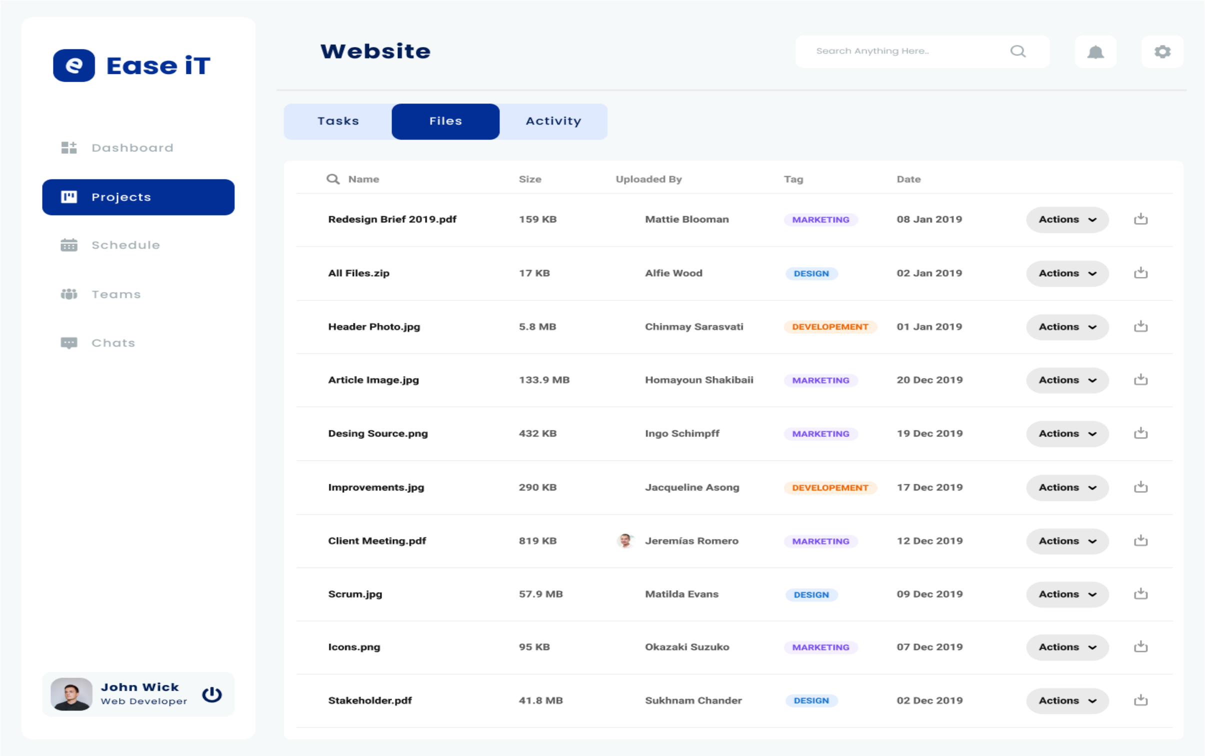
E-mail after registration so that I can confirm my
DEVELOPEMENT

Done

E-mail after registration so that I can confirm my
DEVELOPEMENT

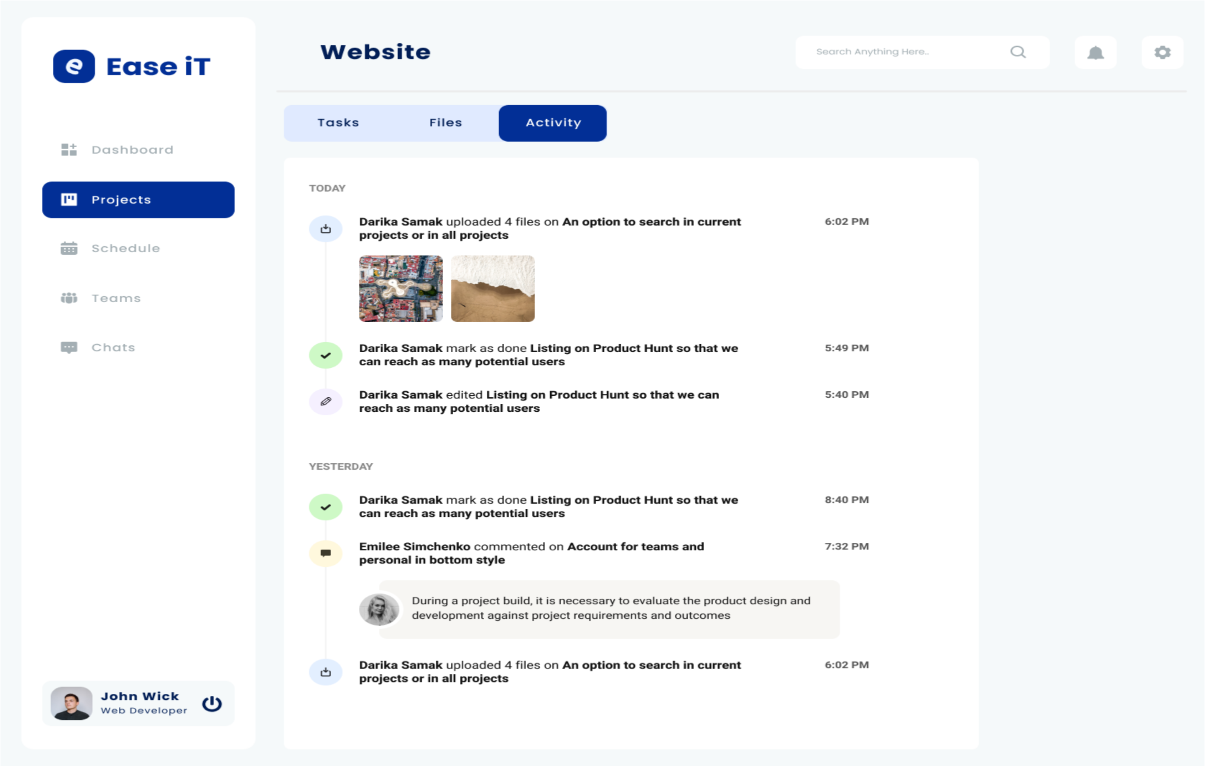
Find top 5 customers and get reviews from them
MARKETING

Files



Name	Size	Uploaded By	Tag	Date		
Redesign Brief 2019.pdf	159 KB	Mattie Blooman	MARKETING	08 Jan 2019	Actions	
All Files.zip	17 KB	Alfie Wood	DESIGN	02 Jan 2019	Actions	
Header Photo.jpg	5.8 MB	Chinmay Sarasvati	DEVELOPEMENT	01 Jan 2019	Actions	
Article Image.jpg	133.9 MB	Homayoun Shakibaii	MARKETING	20 Dec 2019	Actions	
Desing Source.png	432 KB	Ingo Schimpff	MARKETING	19 Dec 2019	Actions	
Improvements.jpg	290 KB	Jacqueline Asong	DEVELOPEMENT	17 Dec 2019	Actions	
Client Meeting.pdf	819 KB	Jeremías Romero	MARKETING	12 Dec 2019	Actions	
Scrum.jpg	57.9 MB	Matilda Evans	DESIGN	09 Dec 2019	Actions	
Icons.png	95 KB	Okazaki Suzuko	MARKETING	07 Dec 2019	Actions	
Stakeholder.pdf	41.8 MB	Sukhnam Chander	DESIGN	02 Dec 2019	Actions	

Activities



TODAY

Darika Samak uploaded 4 files on **An option to search in current projects or in all projects** 6:02 PM

Darika Samak mark as done **Listing on Product Hunt so that we can reach as many potential users** 5:49 PM

Darika Samak edited **Listing on Product Hunt so that we can reach as many potential users** 5:40 PM

YESTERDAY

Darika Samak mark as done **Listing on Product Hunt so that we can reach as many potential users** 8:40 PM

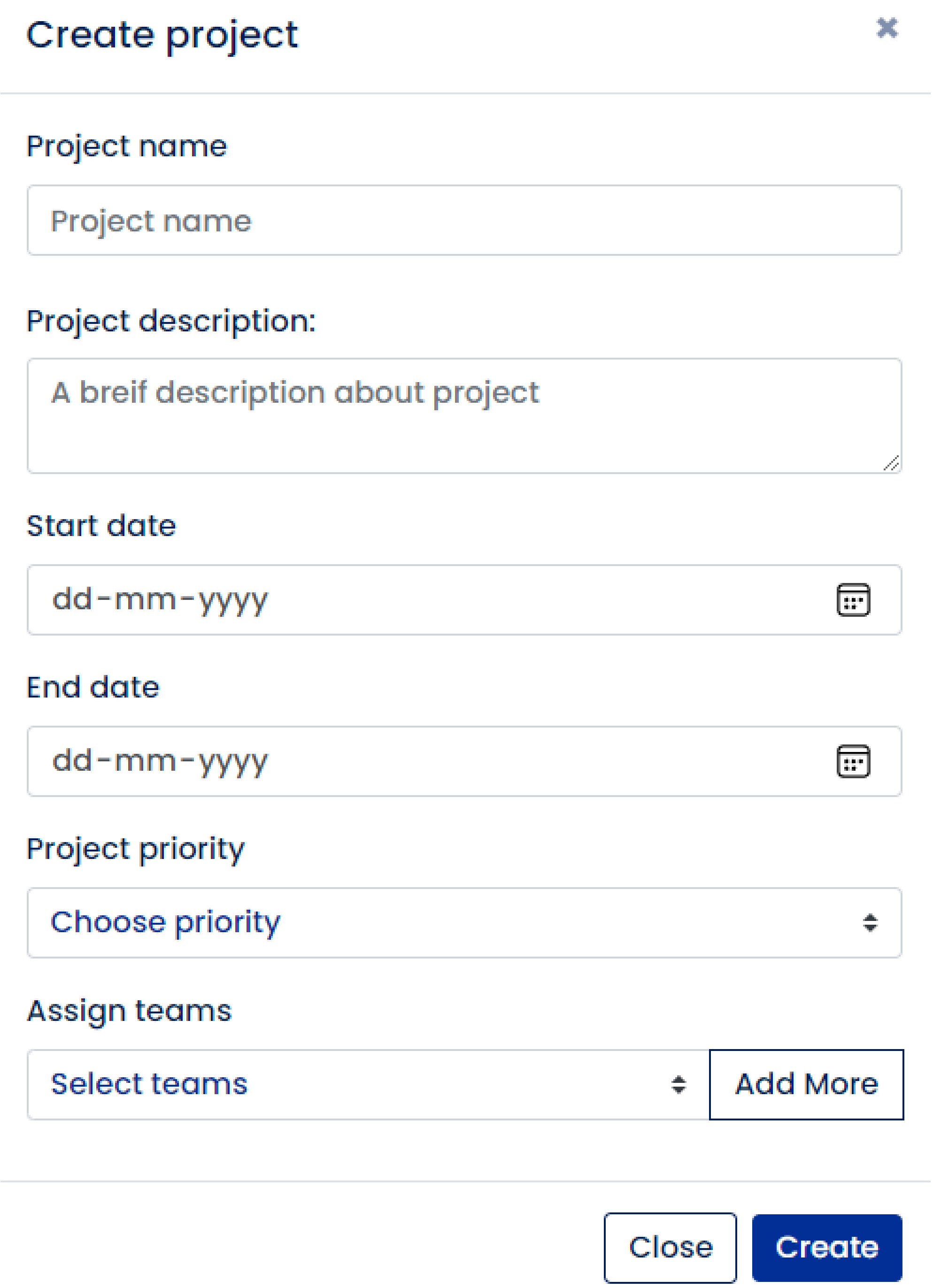
Emilee Simchenko commented on **Account for teams and personal in bottom style** 7:32 PM

During a project build, it is necessary to evaluate the product design and development against project requirements and outcomes

Darika Samak uploaded 4 files on **An option to search in current projects or in all projects** 6:02 PM

4.3.2 Form Designs

Form Name: Create Project



Create project

Project name

Project name

Project description:

A breif description about project

Start date

dd-mm-yyyy

End date

dd-mm-yyyy

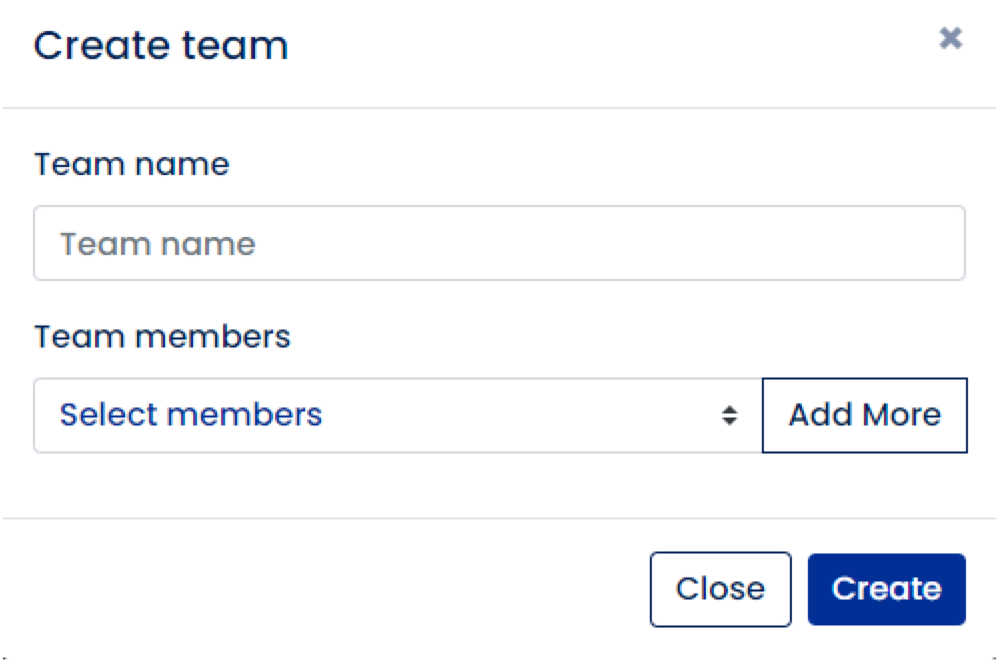
Project priority

Choose priority

Assign teams

Select teams

Form Name: Create Team



Create team

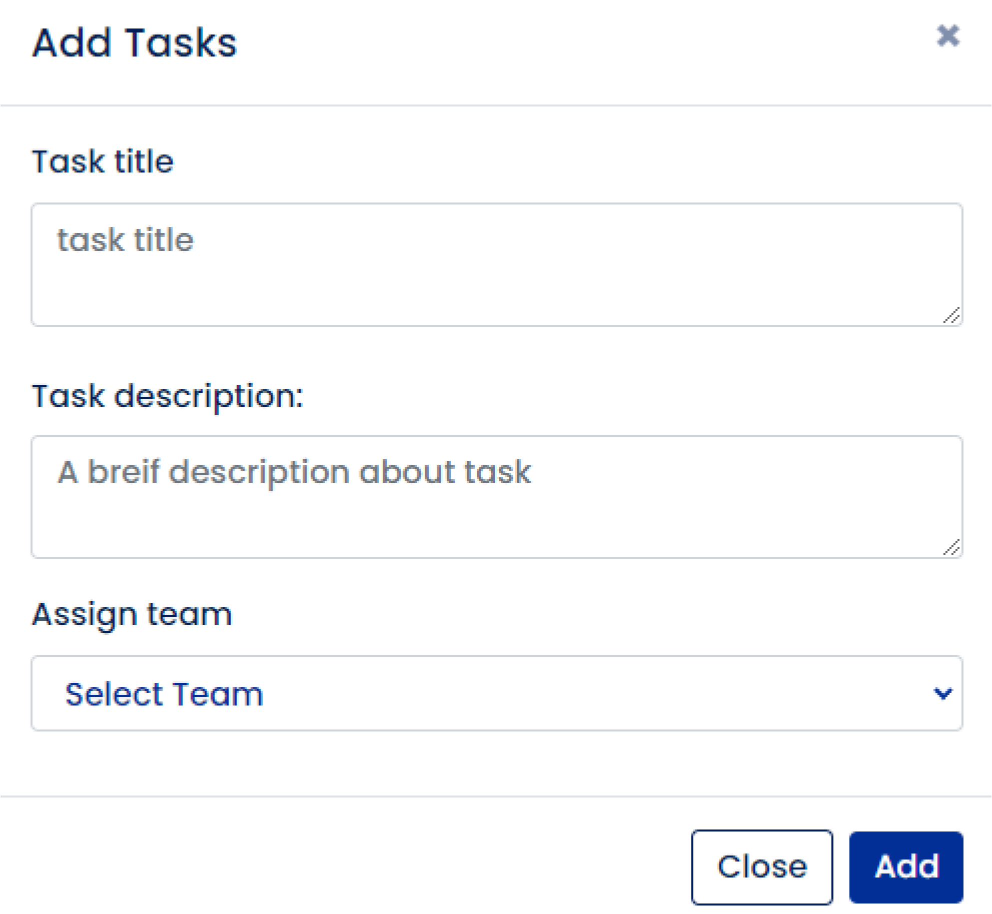
Team name

Team name

Team members

Select members

Form Name: Add Task



Add Tasks

Task title

task title

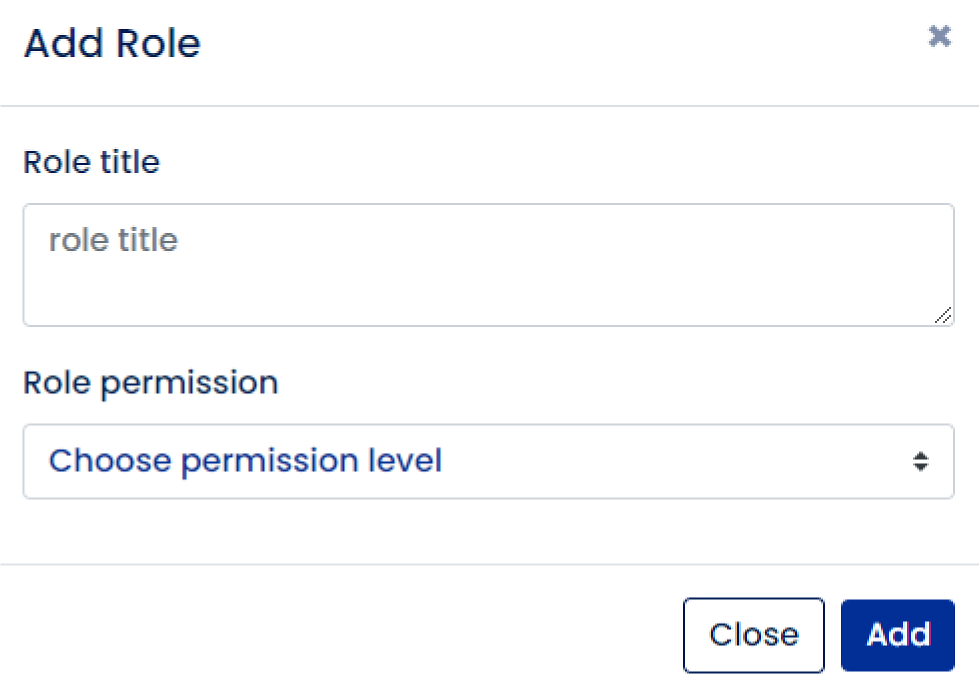
Task description:

A breif description about task

Assign team

Select Team

Form Name: Add User Roles



Add Role

Role title

role title

Role permission

Choose permission level

4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key is Super Key and Candidate Keys.

Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form. As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on another non-key attribute

Table Design

Table No: 01

Table Name: tbl_user

Primary Key: user_id

Foreign Key: role_id, team_id

Table Description: To store user information

Fieldname	**Data Type**	**Size**	**Description**
user_id	int	10	primary key
username	varchar	100	name of the user
mob	varchar	10	mobile number
email	varchar	50	email id
dob	varchar	8	date of birth
password	varchar	100	password
user_created_at	varchar	10	date of user created
user_status	int	2	status of user
role_id	int	10	foreign key of tbl_user_role
team_id	int	20	foreign key of tbl_teams

Table No: 02

Table Name: tbl_user_role

Primary Key: role_id

Foreign Key:

Table Description: To store user role types information

Fieldname	Data Type	Size	Description
role _id	int	10	primary key
role_name	varchar	50	user role name
role_permission	int	10	user permission

Table No: 03

Table Name: tbl_project

Primary Key: project_id

Foreign Key:

Table Description: To store details of projects

Fieldname	Data Type	Size	Description
project_id	int	15	primary key
project_name	varchar	50	name of the project
project_description	varchar	500	description of the project
project_created_date	varchar	10	date of project created
project_start_date	varchar	10	date which project will start
project_end_date	varchar	10	date which project will end
project_status	int	10	status of the project
project_priority	int	10	priority of project

Table No: 04

Table Name: tbl_teams

Primary Key: team_id

Foreign Key: manager_id

Table Description: To store details of teams

Fieldname	Data Type	Size	Description
team_id	int	10	primary key
team_title	varchar	50	team name
manager_id	int	10	foreign key of tbl_user

Table No: 05

Table Name: tbl_team_members

Primary Key: member_id

Foreign Key: user_id, team_id

Table Description: To store details of team members

Fieldname	Data Type	Size	Description
member_id	int	10	primary key
team_id	int	10	foreign key of tbl_teams
user_id	int	10	foreign key of tbl_user

Table No: 06

Table Name: tbl_team_allocation

Primary Key: team_allocation_id

Foreign Key: team_id, project_id, project_manager

Table Description: To store details of team allocation to projects

Fieldname	**Data Type**	**Size**	**Description**
team_allocation_id	int	10	primary key
team_id	int	10	foreign key of tbl_teams
project_id	int	10	foreign key of tbl_project
project_manager	int	10	foreign key of tbl_user

Table No: 07

Table Name: tbl_tasks

Primary Key: task_id

Foreign Key: project_id, team_id, task_added_by

Table Description: To store information about tasks

Fieldname	**Data Type**	**Size**	**Description**
task_id	int	15	primary key
task_title	varchar	300	title of the task
task_description	varchar	500	description of the task
team_id	int	15	foreign key of tbl_teams
task_added_by	int	15	foreign key of tbl_user
task_status	int	11	status of the task
project_id	int	15	foreign key of tbl_project
task_ceated_at	date		current_timestamp()

Table No: 08

Table Name: tbl_files

Primary Key: file_id

Foreign Key: project_id, team_id, uploaded_by_id

Table Description: To store uploaded file details

Fieldname	Data Type	Size	Description
file_id	int	15	primary key
file_name	varchar	50	name of the file
file_size	varchar	50	size of the file
uploaded_by_id	int	15	foreign key of tbl_user
uploaded_date	varchar		file uploaded date
team_id	int	15	foreign key of tbl_teams
project_id	int	15	foreign key of tbl_project

Table No: 09

Table Name: tbl_events

Primary Key: id

Foreign Key:

Table Description: To store schedules and event details

Fieldname	Data Type	Size	Description
id	int	10	primary key
title	varchar	255	title of the event
start	date		start date of event
end	date		end date of event

Table No: 10

Table Name: tbl_invitation

Primary Key: invitation_id

Foreign Key: email, team_id

Table Description: To store details of invited members

Fieldname	**Data Type**	**Size**	**Description**
invitation_id	int	10	primary key
email	varchar	100	foreign key of tbl_email
team_id	int	15	foreign key of tbl_teams
referral_id	varchar	100	referral id sends to user
invite_status	int	10	user invitation status

Table No: 11

Table Name: tbl_activity

Primary Key: activity_id

Foreign Key: activity_by, project_id

Table Description: To store all user activities

Fieldname	**Data Type**	**Size**	**Description**
activity_id	int	10	primary key

activity_desc	varchar	5000	Description of the activity
activity_by	int	15	foreign key of tbl_user
activity_date	datetime		date of which the activity added
project_id	int	10	foreign key of tbl_project
activity_type	varchar	500	the type of activity

Table No: 12

Table Name: tbl_chats

Primary Key: chat_id

Foreign Key: sender_id, receiver_id

Table Description: To store all chat details

Fieldname	**Data Type**	**Size**	**Description**
chat_id	int	10	primary key
sender_id	int	15	foreign key of tbl_user
sender_name	varchar	100	Name of the sender
receiver_id	int	15	foreign key of tbl_user
receiver_name	varchar	100	Name of the receiver
date_time	datetime		Timestamp of the chat

chat_text	varchar	1000	Actual chat message
chat_image	varchar	500	location of image attachments
chat_attachment	varchar	500	location of other document attachment

Table No: 13

Table Name: tbl_checklist

Primary Key: checklist_id

Foreign Key: task_id, project_id, user_id

Table Description: To store checklist details

Fieldname	**Data Type**	**Size**	**Description**
checklist_id	int	10	primary key
task_id	int	11	foreign key of tbl_tasks
project_id	int	11	foreign key of tbl_project
user_id	int	11	foreign key of tbl_user
checklist_title	varchar	500	title of the checklist
status	int	11	status of the checklist

Table No: 13

Table Name: tbl_comments

Primary Key: comment_id

Foreign Key: task_id, project_id, user_id

Table Description: To store comment details

Fieldname	**Data Type**	**Size**	**Description**
comment_id	int	10	primary key
task_id	int	11	foreign key of tbl_tasks
project_id	int	11	foreign key of tbl_project
user_id	int	11	foreign key of tbl_user
comment	varchar	500	Actual comment
comment_time	datetime	11	Timestamp of the comment

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as its vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appears to be working according to the specification, that performance requirement appears to have been met.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers are always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs
- Output Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Testing using Selenium

Selenium is one of the most widely used open-source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages.

Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

Selenium supports a variety of programming languages through the use of drivers specific to each language. Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.

Selenium can be used to automate functional tests and can be integrated with automation test tools such as Maven, Jenkins, & Docker to achieve continuous testing. It can also be integrated with tools such as TestNG, & JUnit for managing test cases and generating reports.

5.2.5.1 Login Page Testcase

Test Suite ID	TS001
Test Case ID	TC001
Test Case Summary	Login
Related Requirement	RS001
Prerequisites	1. Internet Connection 2. Pre-registered user
Test Procedure	1. Type in email id 2. Type in password

Test Data	1. mrudul@gmail.com 2. Enter password in alphanumeric (Eg: Mrudul@123)
Expected Result	1. Login Successful and redirected to dashboard. 2. Please enter valid alphanumeric password
Actual Result	1. Login Successful and redirected to dashboard. 2. Please enter valid alphanumeric password
Status	Pass
Remarks	Test case executed as expected.
Created By	Mrudul A Thakadiyel
Date of Creation	20/05/2022
Executed By	Mr. Jinson Devis
Date of Execution	21/05/2022
Test Environment	OS: Windows 10 Browser: Google Chrome

Code

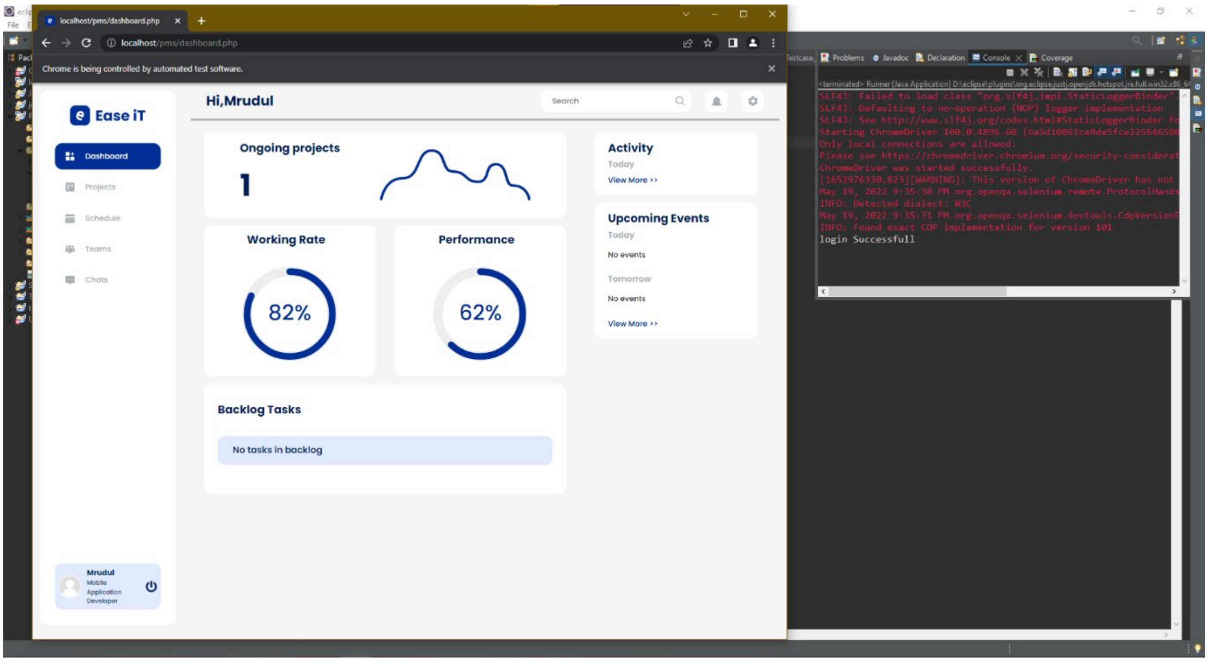
```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

import browserimplementation.DriverSetup;

public class Testcase {
        static WebDriver driver;

        // login
        public static void login() {
                driver = DriverSetup.getWebDriver("http://localhost/pms/");
                driver.findElement(By.name("email")).sendKeys("mrudul@gmail.com");
                driver.findElement(By.name("password")).sendKeys("Mrudul@123");
                driver.findElement(By.name("LoginSubmit")).click();
                WebElement                              logoutBtn                              =
driver.findElement(By.xpath("/html/body/div/div[1]/div/a[3]"));
                if (logoutBtn.isDisplayed()) {
                        System.out.println("login Successfull");
                } else {
                        System.out.println("login Failed");
                }
        }
public static void main(String[] args) throws InterruptedException {
                Testcase.login();
        }
}
```

Output:



```
<terminated> Runner [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder fo
Starting ChromeDriver 100.0.4896.60 (6a5d10861ce8de5fce225646580
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerat
ChromeDriver was started successfully.
[1652976330.823][WARNING]: This version of ChromeDriver has not
May 19, 2022 9:35:30 PM org.openqa.selenium.remote.ProtocolHands
INFO: Detected dialect: W3C
May 19, 2022 9:35:31 PM org.openqa.selenium.devtools.CdpVersionF
INFO: Found exact CDP implementation for version 101
login Successfull
```

5.2.5.2 Search Functionality Testcase

Test Suite ID	TS002
Test Case ID	TC002
Test Case Summary	Search Functionality
Related Requirement	RS001
Prerequisites	1. Internet Connection 2. User must be logged in
Test Procedure	1. Type in a search key 2. Click on the first option
Test Data	1. Cab
Expected Result	1. Find element Successfully and redirected to corresponding page.
Actual Result	1. Find element Successfully and redirected to corresponding page.
Status	Pass
Remarks	Test case executed as expected.
Created By	Mrudul A Thakadiyel
Date of Creation	20/05/2022
Executed By	Mr. Jinson Devis
Date of Execution	21/05/2022

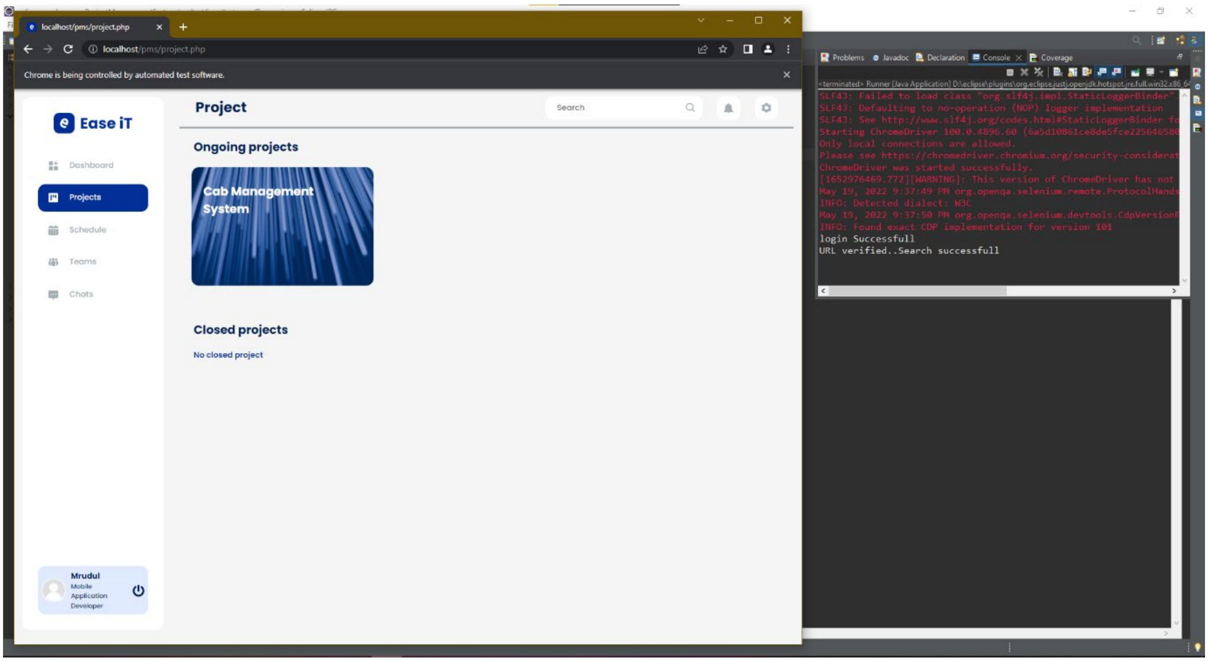
Test Environment	OS: Windows 10 Browser: Google Chrome

Code

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

import browserimplementation.DriverSetup;
public class Testcase {
        static WebDriver driver;
        // search
        public static void search() throws InterruptedException {
                driver.findElement(By.id("searchInput")).sendKeys("Cab");
                Thread.sleep(1000);
                driver.findElement(By.xpath("//*[@id=\"1\"]")).click();
                String currentUrl = driver.getCurrentUrl();
                String expectedUrl = "http://localhost/pms/project.php";
                if (currentUrl.equals(expectedUrl)) {
                        System.out.println("URL verified..Search successfull");
                } else {
                        System.out.println("verificaton is failed");
                }
        }
}
```

Output:



```
<terminated> Runner [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder fo
Starting ChromeDriver 100.0.4896.60 (6a5d10861ce8de5fce225646580
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerat
ChromeDriver was started successfully.
[1652976469.772][WARNING]: This version of ChromeDriver has not
May 19, 2022 9:37:49 PM org.openqa.selenium.remote.ProtocolHands
INFO: Detected dialect: W3C
May 19, 2022 9:37:50 PM org.openqa.selenium.devtools.CdpVersionF
INFO: Found exact CDP implementation for version 101
login Successfull
URL verified..Search successfull
```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The aim of the project was to make a complete, fully working web-based project management system for the company. Requirements from the company has been gathered and taken into account. In web-based project management system there has been used an already implemented system to improve company's everyday use and to increase performance, productivity and efficiency. As a good project management system, it has a possibility to upload, download and delete files and uniformly gives change for developers to be in constant contact with the customer requirements and expectations for the project. User management tool in web-based project management system is a good appliance for keeping eye on the project and for giving rights to different users by system administrator in company. This all makes a complete and good communication system inside company, all data and material will be accessible from one place, to facilitate the solution of a project and contact communication with a client. Finally, the whole system has been tested to ensure that everything functions correctly before the system processes actual data and produces information that people will rely on.

7.2 FUTURE SCOPE

- Notification and alert about project and task deadlines can be added
- Can generate a consolidated project report

CHAPTER 8

BIBLIOGRAPHY

REFERENCES

- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
- Roger S Pressman, “Software Engineering”, 1994.
- Pankaj Jalote, “Software engineering: a precise approach”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES

- jQuery AJAX Introduction (w3schools.com)
- ClickUp
- Trello
- Jira | Issue & Project Tracking Software | Atlassian
- Bitrix24

CHAPTER 9

APPENDIX

9.1 SAMPLE CODE

Admin

adminDashboard.php

```
<?php

include('./config/connect.php');

session_start();

if (isset($_SESSION["pmsSession"]) != session_id()) {

   header("Location: ./index.php");

   die();

} else {

   $team_id=$_SESSION["currentUserTeamId"];

?>

   <!DOCTYPE html>

   <html>

   <head>

      <meta charset="utf-8" />

      <meta http-equiv="X-UA-Compatible" content="IE=edge" />

      <title></title>

      <meta name="description" content="" />

      <meta name="viewport" content="width=device-width, initial-scale=1" />

      <link rel="stylesheet" href="./stylesheets/css/style.css" />

      <!-- Bootstrap CSS -->

      <link rel="stylesheet" href="./stylesheets/css/bootstrap.min.css" />

      <link rel="icon" href="./assets/images/logo2.png" type="image/icon type" />
```

```
</head>
<body class="dashboard-body">
  <div class="dashboard-container">
    <!--sidebar goes here-->
    <?php include_once("./layouts/adminSidebar.php"); ?>
    <!--sidebar end-->
    <!--header starts-->
    <?php include_once("./layouts/header.php"); ?>
    <!--header ends-->
    <!--Dashboard contents-->
    <div class="dashboard-contents">
      <div class="row">
        <!--col 1 start-->
        <div class="col-8">
          <div class="d-flex">
            <div class="col">
              <div class="completed-task">
              <div class="d-flex flex-column">
                  <h1 class="content-heading">Total Project</h1>
                  <?php
                  $userid = $_SESSION['userId'];
                  //select from tbl_team_allocation table
                    $sqlQuery = "SELECT count(DISTINCT project_id) AS proCount
FROM tbl_team_allocation";
                  $result2 = mysqli_query($connect, $sqlQuery);
```

```
                $count2 = mysqli_num_rows($result2);
                if($count2>0){
                  $row2 = mysqli_fetch_assoc($result2);
                  $project_count=$row2['proCount'];
                  echo '<h2 class="stats">'.$project_count.'</h2>';
                }
                else{
                  echo '<h2 class="stats">0</h2>';
                }
                ?>
              </div>
              <img class="chart-img" src="./assets/images/Chart.svg" alt="" />
            </div>
          </div>
        </div>
        <div class="d-flex">
          <div class="col-6">
          <div class="completed-task mt-3">
            <div class="d-flex flex-column">
                <h1 class="content-heading">Total Teams</h1>
                <?php
                $userid = $_SESSION['userId'];
                //select from tbl_team table
$sqlQuery3 = "SELECT count(DISTINCT team_id) AS teamCount FROM tbl_teams";
```

```
                    $result3 = mysqli_query($connect, $sqlQuery3);
                    $count3 = mysqli_num_rows($result3);
                    if($count3>0){
                      $row3 = mysqli_fetch_assoc($result3);
                      $teamCount=$row3['teamCount'];
                      echo '<h2 class="stats">'.$teamCount.'</h2>';
                    }
                    else{
                      echo '<h2 class="stats">0</h2>';
                    }
                    ?>
                  </div>
                </div>
              </div>
              <div class="col-6">
              <div class="completed-task mt-3">
                <div class="d-flex flex-column">
                    <h1 class="content-heading">Total users</h1>
                    <?php
                    $userid = $_SESSION['userId'];
                    //select from tbl_user table
      $sqlQuery4 = "SELECT count(DISTINCT user_id) AS userCount FROM tbl_user";
                    $result4 = mysqli_query($connect, $sqlQuery4);
                    $count4 = mysqli_num_rows($result4);
                    if($count2>0){
                        $row4 = mysqli_fetch_assoc($result4);
                        $userCount=$row4['userCount'];
                        echo '<h2 class="stats">'.$userCount.'</h2>';
                    }
                    else{
                        echo '<h2 class="stats">0</h2>';
                    }
                    ?>
                </div>
            </div>
        </div>
    </div>
</div>
<!--col 1 end-->
<!--col 2 start-->
<div class="col-4">
    <div class="d-flex flex-column">
        <div class="dashboard-card">
            <h1 class="content-heading">Activity</h1>
            <h3 class="sub-title">Today</h3>
            <ul class="activity-container">
                <li class="items">
                    <img src="./assets/icons/tick-dark-ico.svg" alt="" />
                    <div class="card-text">
                      <p>Darika Samak mark as done Listing on Product Hunt</p>
                      <div class="time-stamp">8:40pm</div>
                    </div>
                  </li>
                  <li class="items">
                    <img src="./assets/icons/comment-ico.svg" alt="" />
                    <div class="card-text">
                      <p>Darika Samak mark as done Listing on Product Hunt</p>
                      <div class="time-stamp">8:40pm</div>
                    </div>
                  </li>
                  <li class="items">
                    <img src="./assets/icons/upload-ico.svg" alt="" />
                    <div class="card-text">
                      <p>Darika Samak mark as done Listing on Product Hunt</p>
                      <div class="time-stamp">8:40pm</div>
                    </div>
                  </li>
                </ul>
              </div>
            </div>
          </div>
          <!--col 2 end-->
```

```
            </div>
          </div>
        </div>
        <script src="//code.jquery.com/jquery-3.1.1.slim.min.js"></script>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
       <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
fQybjgWLrvvRgtW6bFlB7jaZrFsaBXjsOMm/tB9LTS58ONXgqbR9W8oWht/amnpF"
crossorigin="anonymous">
        </script>
        <script src="./js/app.js"></script>
        <script>
          $(document).ready(function() {
            //checklist task
            $(".styled-checkbox").change(function() {
              var $this = $(this);
              let taskStatus = 4;
              let checkParent = $this.parent();
              let checkSibling = checkParent.next();
              let title = checkSibling.find('.task-title');
              if ($this.is(":checked")) {
                let taskId = $this.attr("id");
                $.ajax({
                  url: './server/updateTask.php',
                  type: 'POST',
                    data: {
                        taskId: taskId,
                        taskStatus: taskStatus
                    },
                    success: function(data) {
                        title.addClass('strike');
                        setTimeout(function() {
                            location.reload();
                        }, 2000);
                    }
                });
            } else {
                alert($this.attr("id") + " is unchecked");
            }
        });
    });
    </script>
  </body>
  </html>
<?php
}
?>
```

Manageusers.php

```
<?php
include('./config/connect.php');
session_start();
if (isset($_SESSION["pmsSession"]) != session_id()) {
  header("Location: ./index.php");
  die();
} else {
?>
  <!DOCTYPE html>
  <html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <title></title>
    <meta name="description" content="" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="./stylesheets/css/style.css" />
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="./stylesheets/css/bootstrap.min.css" />
    <link rel="icon" href="./assets/images/logo2.png" type="image/icon type" />
  </head>
  <body class="dashboard-body">
    <div class="dashboard-container">
```

```
<!--sidebar goes here-->
<?php include_once("./layouts/adminSidebar.php"); ?>
<!--sidebar end-->
<!--header starts-->
<?php include_once("./layouts/header.php"); ?>
<!--header ends-->
<!--Dashboard contents-->
<div class="dashboard-contents">
  <div class="row">
    <!--col 2 start-->
    <div class="col-12">
      <div class="d-flex flex-column">
        <div class="files-card">
          <div class="d-flex justify-content-between">
            <h1 class="content-heading">Team Members</h1>
                <button data-toggle='modal' data-target='#addMemberModal' class="add-task-
btn">Add Members +</button>
          </div>
          <!---table start-->
          <table class="table files-table">
            <thead>
              <tr class="t-head">
                <th>Name</th>
                <th>Email ID</th>
                <th>Mobile No</th>
                <th>Role</th>

                <th>Status</th>

              </tr>

            </thead>

            <tbody id="filesContainer">

            </tbody>

          </table>

          <!---table end-->

          <h3 class="view-more-btn">View More >></h3>

        </div>

      </div>

    </div>

    <!--col 2 end-->

  </div>

 </div>

</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

 <script  src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
Piv4xVNRyMGpqkS2by6br4gNJ7DXjqk09RmUpJ8jgGtD7zP9yug3goQfGII0yAns"
crossorigin="anonymous"></script>

<script src="./js/app.js"></script>

<script>

 $(document).ready(function() {

  let memberCount = 4;
```

```
        $("#filesContainer").load("./server/loadUsers.php", {
          memberCount: memberCount
        });
        $(".view-more-btn").click(function() {
          memberCount += 4;
          $("#filesContainer").load("./server/loadUsers.php", {
            memberCount: memberCount
          });
        });
      });
    </script>
  </body>
  </html>
<?php
}
?>
```

manageUserRole.php

```
<?php
include('./config/connect.php');
session_start();
if (isset($_SESSION["pmsSession"]) != session_id()) {
  header("Location: ./index.php");
  die();
} else {
```

```
?>
 <!DOCTYPE html>
 <html>
 <head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title></title>
  <meta name="description" content="" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="./stylesheets/css/style.css" />
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="./stylesheets/css/bootstrap.min.css" />
  <link rel="icon" href="./assets/images/logo2.png" type="image/icon type" />
 </head>
 <body class="dashboard-body">
  <div class="dashboard-container">
   <!--sidebar goes here-->
   <?php include_once("./layouts/adminSidebar.php"); ?>
   <!--sidebar end-->
   <!--header starts-->
   <?php include_once("./layouts/header.php"); ?>
   <!--header ends-->
   <!--Dashboard contents-->
   <div class="dashboard-contents">
```

```
    <div class="row">
     <!--col 2 start-->
     <div class="col-12">
       <div class="d-flex flex-column">
         <?php
         if (isset($_SESSION['addRoleStatus'])) {
           echo '<div class="alert alert-primary alert-dismissible fade show" role="alert">
           <strong>' . $_SESSION['addRoleStatus'] . '</strong>
           <button type="button" class="close" data-dismiss="alert" aria-label="Close">
             <span aria-hidden="true">&times;</span>
           </button>
         </div>';
           unset($_SESSION['addRoleStatus']);
         }
         ?>
         <div class="files-card">
           <div class="d-flex justify-content-between">
             <h1 class="content-heading">User Roles</h1>
                   <button  data-toggle='modal'  data-target='#addRoleModal'  class="add-task-btn">Add Role +</button>
           </div>
           <!---table start-->
           <table class="table files-table">
             <thead>
               <tr class="t-head">
```

```
                    <th>Role ID</th>

                    <th>Role Name</th>

                  </tr>

                </thead>

                <tbody id="filesContainer">

                </tbody>

              </table>

              <!---table end-->

              <h3 class="view-more-btn">View More >></h3>

            </div>

          </div>

        </div>

        <!--col 2 end-->

      </div>

    </div>

  </div>

  <!-- add role Modal starts-->

            <div class="modal fade" id="addRoleModal" tabindex="-1" aria-labelledby="addRoleModalLabel" aria-hidden="true">

    <div class="modal-dialog">

      <div class="modal-content">

                  <form id="addRoleForm" class="modal-form-container" method="post" action="./server/addUserRole.php">

          <div class="modal-header">

            <h5 class="modal-title" id="addtaskjectModalLabel">Add Role</h5>
            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
              <span aria-hidden="true">&times;</span>
            </button>
          </div>
          <div class="modal-body">
          <div class="alert alert-warning alert-dismissible fade show" role="alert" id="alertMsg"
     style="display:none;">
              <p class="msg">Please fill all the field !</p>
              <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                <span aria-hidden="true">&times;</span>
              </button>
            </div>
            <div class="form-group">
              <label for="roleName">Role title</label>
             <textarea name="roleName" id="roleName" class="form-control" placeholder="role
     title" autocomplete="off" required></textarea>
            </div>
            <div class="form-group">
              <label for="rolePermission">Role permission</label>
                  <select class="custom-select" name="rolePermission" id="rolePermission" aria-
     label="Example select with button addon">
                <option disabled selected>Choose permission level</option>
                <option value="2">Manager level</option>
                <option value="3">User level</option>
              </select>
```

```
           </div>
          </div>
          <div class="modal-footer">
                    <button type="button" class="btn btn-secondary modal-btn" data-
dismiss="modal">Close</button>
           <button id="addRoleBtn" name="addRoleBtn" type="submit" class="btn btn-primary
modal-btn-submit">Add</button>
          </div>
         </form>
        </div>
       </div>
     </div>
     <!--add role  Modal ends-->
     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
      <script  src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
Piv4xVNRyMGpqkS2by6br4gNJ7DXjqk09RmUpJ8jgGtD7zP9yug3goQfGII0yAns"
crossorigin="anonymous"></script>
     <script src="./js/app.js"></script>
     <script>
      $(document).ready(function() {
        let memberCount = 4;
        $("#filesContainer").load("./server/loadRole.php", {
          memberCount: memberCount
        });
        $(".view-more-btn").click(function() {
```

```
     memberCount += 4;

     $("#filesContainer").load("./server/loadRole.php", {

       memberCount: memberCount

     });

    });

   });

  </script>

 </body>

 </html>

<?php

}

?>
```

Manager/Team members

Project.php

```
<?php

include('./config/connect.php');

session_start();

if (isset($_SESSION["pmsSession"]) != session_id()) {

  header("Location: ./index.php");

  die();

} else {

  $managerId = $_SESSION['userId'];

?>

  <!DOCTYPE html>
```

```
<html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title></title>
  <meta name="description" content="" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="./stylesheets/css/style.css" />
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="./stylesheets/css/bootstrap.min.css" />
  <link rel="icon" href="./assets/images/logo2.png" type="image/icon type" />
</head>
<body class="dashboard-body">
  <div class="dashboard-container">
    <!--sidebar goes here-->
    <?php include_once("./layouts/sidebar.php");
    ?>
    <!--sidebar end-->
    <!--header starts-->
    <?php include_once("./layouts/header.php");
    ?>
    <!--header ends-->
    <!--Dashboard contents-->
    <div class="dashboard-contents">
```

```
        <div class="project-container">
          <?php
          //select items from tbl_project if type_id is Manager in tbl_user
          if ($_SESSION['currentUserTypeId'] == '2') {
              $query = "SELECT * FROM tbl_project WHERE project_id IN (SELECT
distinct project_id FROM tbl_team_allocation aloc JOIN tbl_user u ON aloc.project_manager
= u.user_id WHERE u.user_id = '" . $_SESSION['userId'] . "')";
            $result = mysqli_query($connect, $query);
            if (mysqli_num_rows($result) > 0) {
              while ($row = mysqli_fetch_assoc($result)) {
                $proId = $row['project_id'];
                $proName = $row['project_name'];
      echo "<a href='./tasks.php?id=$proId' class='projects pro' id='$proId'>$proName</a>";
              }
              echo "<div class='projects' id = 'createPro' data-toggle='modal'></div>";
            } else {
              echo "<div class='projects' id = 'createPro' data-toggle='modal'></div>";
            }
          } else {
            //not manager
              $query = "SELECT * FROM tbl_project WHERE project_id IN (SELECT
distinct project_id FROM tbl_team_allocation aloc JOIN tbl_user u ON aloc.team_id =
u.team_id WHERE u.team_id = '" . $_SESSION['currentUserTeamId'] . "')";
            $result = mysqli_query($connect, $query);
            if (mysqli_num_rows($result) > 0) {
                    while ($row = mysqli_fetch_assoc($result)) {
                        $proId = $row['project_id'];
                        $proName = $row['project_name'];
echo "<a href='./tasks.php?id=$proId' class='projects pro' id='$proId'>$proName</a>";
                    }
                } else {
                    echo '
                    <div class="alert alert-success" role="alert">
                        <h4 class="alert-heading">No Project Yet Added!</h4>
                        <p>Please contact your manager for any queries.</p>
                        <hr>
    <p class="mb-0">For any technical assistance contact system administrator.</p>
                    </div>
                    ';
                }
            }
            ?>
        </div>
        <!-- Modal starts-->
                <div class="modal fade" id="addProjectModal" tabindex="-1" aria-labelledby="addProjectModalLabel" aria-hidden="true">
            <div class="modal-dialog">
                <div class="modal-content">
                    <form id="createProForm" class="modal-form-container" method="post">
                        <input type="hidden" name="assign-count" value="1" id="assign-count">
                <div class="modal-header">
                <h5 class="modal-title" id="addProjectModalLabel">Create project</h5>
      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                  </button>
                </div>
                <div class="modal-body">
                  <div class="alert alert-success alert-dismissible fade show" role="alert" id="success" style="display:none;">
                    <div id="message"></div>
                           <button id="alertClose" type="button" class="close" data-dismiss="alert" aria-label="Close">
                      <span aria-hidden="true">&times;</span>
                    </button>
                  </div>
                  <div class="form-group">
                    <label for="pro-name">Project name</label>
                     <input type="text" name="pro-name" id="pro-name" class="form-control" placeholder="Project name" required autocomplete="off" />
                  </div>
                  <div class="form-group">
     <label for="pro-description" class="col-form-label">Project description:</label>
                  <textarea id="pro-description" name="pro-description" placeholder="A breif description about project" class="form-control"></textarea>
                  </div>
```

```
<div class="form-group">
    <label for="pro-start-date">Start date</label>
        <input type="date" name="pro-start-date" id="pro-start-date" class="form-control" placeholder="Project name" required autocomplete="off" />
</div>
<div class="form-group">
    <label for="pro-end-date">End date</label>
        <input type="date" name="pro-end-date" id="pro-end-date" class="form-control" placeholder="Project name" required autocomplete="off" />
</div>
<div class="form-group">
    <label for="pro-priority">Project priority</label>
    <select class="custom-select" name="pro-priority" id="pro-priority" aria-label="Example select with button addon">
        <option disabled selected>Choose priority</option>
        <option value="1">Top level</option>
        <option value="2">Medium level</option>
        <option value="3">Low level</option>
    </select>
</div>
<div id="team-select" class="form-group">
    <label for="pro-teams">Assign teams</label>
    <div id="duplicater" class="input-group mb-3">
        <select class="custom-select" name="pro-team" id="pro-team1" aria-label="Example select with button addon">
```

```
                                    <option disabled selected>Select teams</option>

                                    <?php

    $sql = "SELECT * FROM tbl_teams WHERE `manager_id`='$managerId'";

                                    $result = mysqli_query($connect, $sql);

                                    while ($row = mysqli_fetch_assoc($result)) {

                                      $team_id = $row['team_id'];

                                      $team_title = $row['team_title'];

                                    echo '<option value="' . $team_id . '">' . $team_title . '</option>';

                                    }

                                    ?>

                                  </select>

                                  <div class="input-group-append">

     <button id="addBtn" class="btn btn-outline-secondary modal-btn" type="button">Add

                                      More</button>

                                  </div>

                                  <div class="input-group-append">

                                   <button style="display:none;" id="delBtn" class="btn btn-outline-
secondary modal-btn" type="button">Delete

                                    </button>

                                  </div>

                                </div>

                              </div>

                            </div>

                            <div class="modal-footer">
```

```
                        <button type="button" class="btn btn-secondary modal-btn" data-dismiss="modal">Close</button>
                     <button id="createProBtn" type="button" class="btn btn-primary modal-btn-submit">Create</button>
                    </div>
                  </form>
               </div>
           </div>
        </div>
        <!-- Modal ends-->
        <!--Confirmation Modal start-->
        <div class="modal fade" id="confirmationModal" tabindex="-1" role="dialog" aria-labelledby="confirmationModalLabel" aria-hidden="true">
          <div class="modal-dialog" role="document">
            <div class="modal-content">
              <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Delete Task</h5>
           <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                  <span aria-hidden="true">&times;</span>
                </button>
              </div>
              <div class="modal-body">
                Are you sure you want to delete this project?
              </div>
              <div class="modal-footer">
```

```
    <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
  <button type="button" id="projectDeleteBtn" class="btn btn-danger">Delete</button>
                </div>
              </div>
            </div>
          </div>
          <!--Confirmation Modal ends-->
        </div>
    </div>
    <script src="./js/app.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
fQybjgWLrvvRgtW6bFlB7jaZrFsaBXjsOMm/tB9LTS58ONXgqbR9W8oWht/amnpF"
crossorigin="anonymous">
    </script>
    <script>
      //add new project
      $(document).ready(function() {
        $('#createPro').click(function() {
          $('#addProjectModal').modal('show');
        });
        $('#createProBtn').on('click', function() {
          var proName = $('#pro-name').val();
          var proDescription = $('#pro-description').val();
            var proStartDate = $('#pro-start-date').val();
            var proEndDate = $('#pro-end-date').val();
            var proPriority = $('#pro-priority').val();
            var proTeam = $('#pro-team').val();
            var proTeamCount = $('#assign-count').val();
            var proTeamArray = [];
            for (var i = 1; i <= proTeamCount; i++) {
              proTeamArray.push($('#pro-team' + i).val());
            }
    if (proName != '' && proDescription != '' && proStartDate != '' && proEndDate != '' &&
              proPriority != '' && proTeam != '', proTeamArray[0] != null) {
              $("#createProBtn").attr("disabled", "disabled");
              $.ajax({
                url: './server/createProject.php',
                type: 'POST',
                data: {
                  proName: proName,
                  proDescription: proDescription,
                  proStartDate: proStartDate,
                  proEndDate: proEndDate,
                  proPriority: proPriority,
                  // proTeam: proTeam,
                  proTeamArray: proTeamArray,
                  proStatus: 1
```

```
                            },
                            success: function(data) {
                                $("#createProBtn").removeAttr("disabled");
                                $('#createProForm').find('input:text').val(");
                                $('#success').show();
                                $('#message').html('Project created successfully !');
                                setTimeout(function() {
                                    $('#success').hide();
                                    $('#addProjectModal').modal('hide');
                                    window.location.reload();
                                }, 1000);
                            }
                        });
                    } else {
                        alert('Please fill all the field !');
                    }
                });
            });
        </script>
    </body>
    </html>
<?php
}
?>
```

addTeam.php

```
<?php
include('../config/connect.php');
extract($_POST);
session_start();
$manager_id = $_SESSION['userId'];
//insert into team table
$sql = "INSERT INTO `tbl_teams`(`team_title`, `manager_id`) VALUES
('$teamName','$manager_id')";
$result = mysqli_query($connect, $sql);
$teamId  = mysqli_insert_id($connect);
if (sizeof($teamMemberArr) > 0 && $teamMemberArr[0] != null) {
   foreach ($teamMemberArr as $memberID) {
      //insert into team_members table
           $sql2 = "INSERT INTO `tbl_team_members`(`team_id`, `user_id`) VALUES
('$teamId','$memberID')"; //not needed, but just in case
      $result2 = mysqli_query($connect, $sql2);
      $sql3 = "UPDATE `tbl_user` SET `team_id`='$teamId' WHERE `user_id`='$memberID'";
      $result3 = mysqli_query($connect, $sql3);
   }
}
if($result){
   echo "success";
}else{
   echo "error";
}
```

addTeamMembers.php

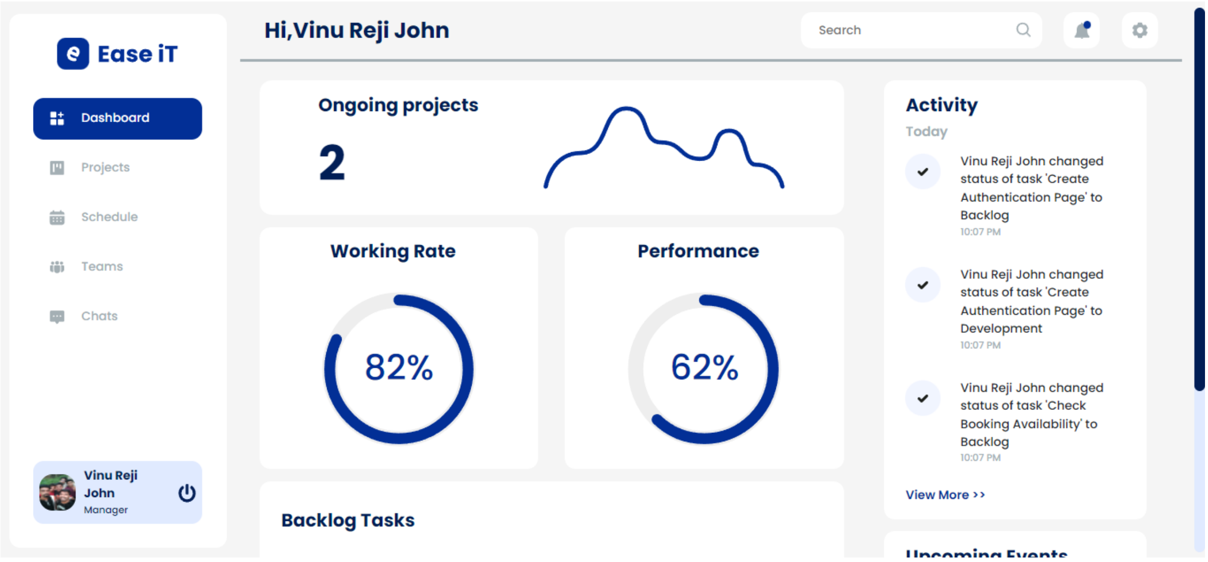
```
<?php
include('../config/connect.php');
extract($_POST);
session_start();
$manager_id = $_SESSION['userId'];
$postStatus = 0;
if (sizeof($teamMemberArr) > 0 && $teamMemberArr[0] != null) {
  foreach ($teamMemberArr as $memberID) {
    //insert into team_members table
        $sql2 = "INSERT INTO `tbl_team_members`(`team_id`, `user_id`) VALUES ('$team_id','$memberID')"; //not needed, but just in case
    $result2 = mysqli_query($connect, $sql2);
    if(!$result2){
      echo "Error: " . $sql2 . "<br>" . mysqli_error($connect);
    }
    else{
      $postStatus = 1;
    }
        $sql3 = "UPDATE `tbl_user` SET `team_id`='$team_id' WHERE `user_id`='$memberID'";
    $result3 = mysqli_query($connect, $sql3);
    if(!$result3){
      echo "Error2: " . $sql3 . "<br>" . mysqli_error($connect);
```

```
        }
        else{
            $postStatus = 1;
        }
    }
}
if($postStatus == 1){
    echo "success";
}
else{
    echo "fail";
}
```

9.2 SCREENSHOTS

Manager/Team members

Dashboard

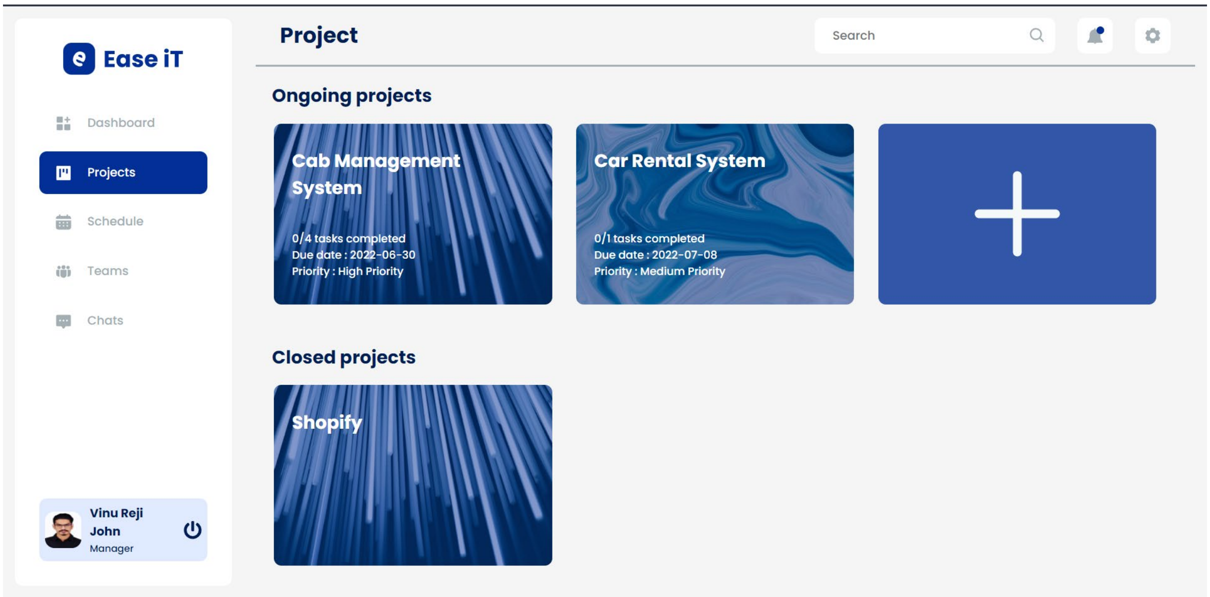


Vinu Reji John changed status of task 'Create Authentication Page' to Backlog
10:07 PM

Vinu Reji John changed status of task 'Create Authentication Page' to Development
10:07 PM

Vinu Reji John changed status of task 'Check Booking Availability' to Backlog
10:07 PM

Project



Ongoing projects

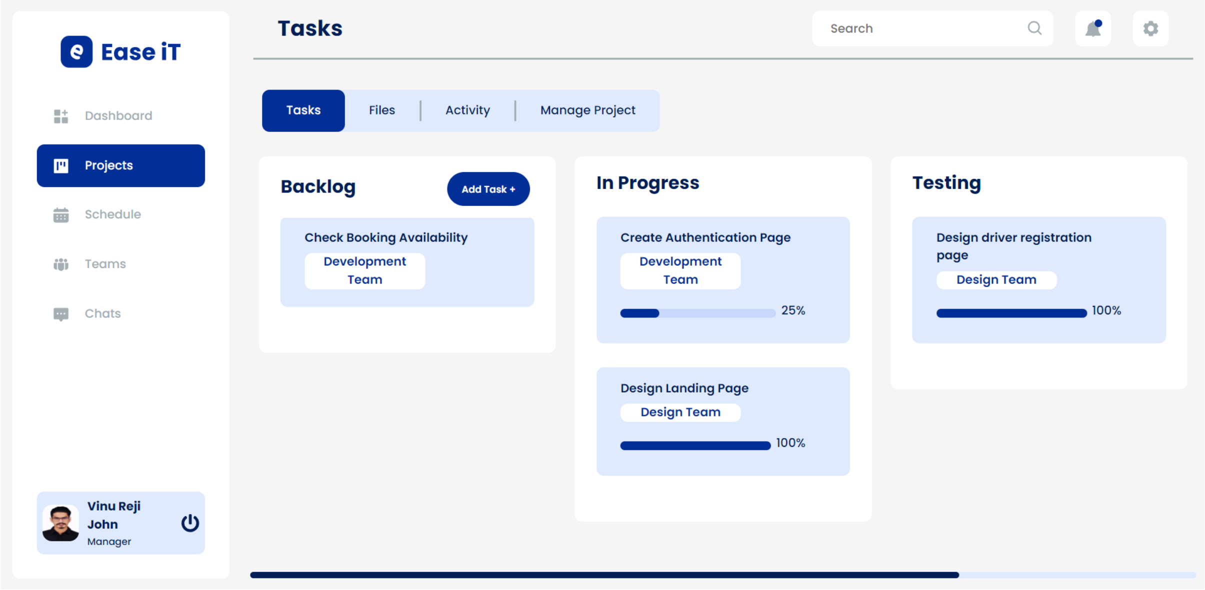
Cab Management System
0/4 tasks completed
Due date : 2022-06-30
Priority : High Priority

Car Rental System
0/1 tasks completed
Due date : 2022-07-08
Priority : Medium Priority

Closed projects

Shopify

Tasks



Backlog

Check Booking Availability
Development Team

In Progress

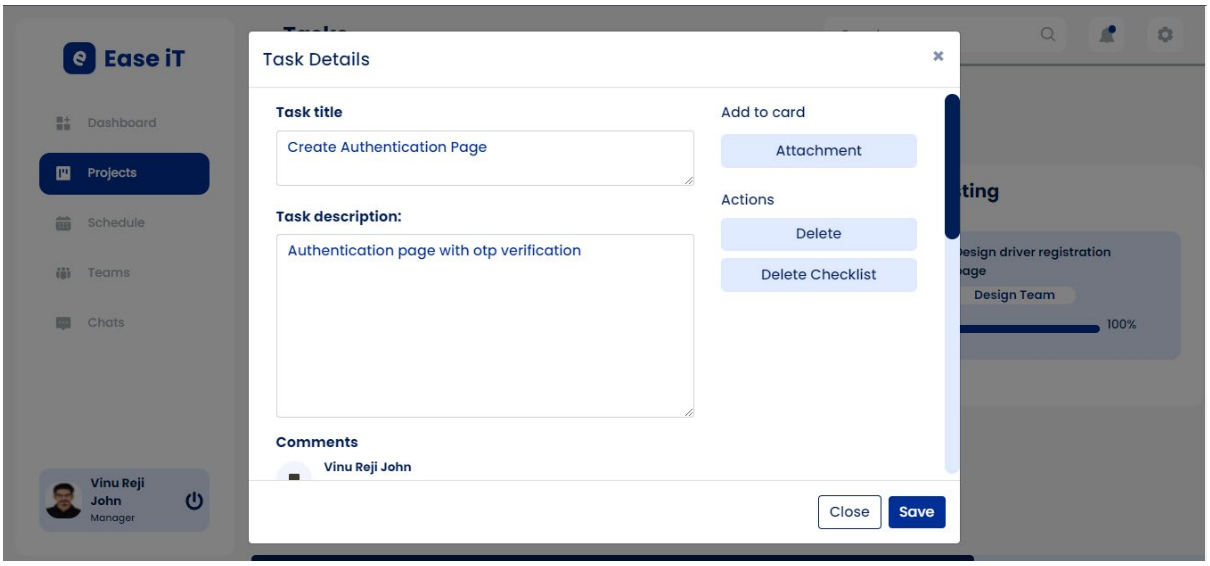
Create Authentication Page
Development Team
25%

Design Landing Page
Design Team
100%

Testing

Design driver registration page
Design Team
100%

Task Details



Task Details

Task title

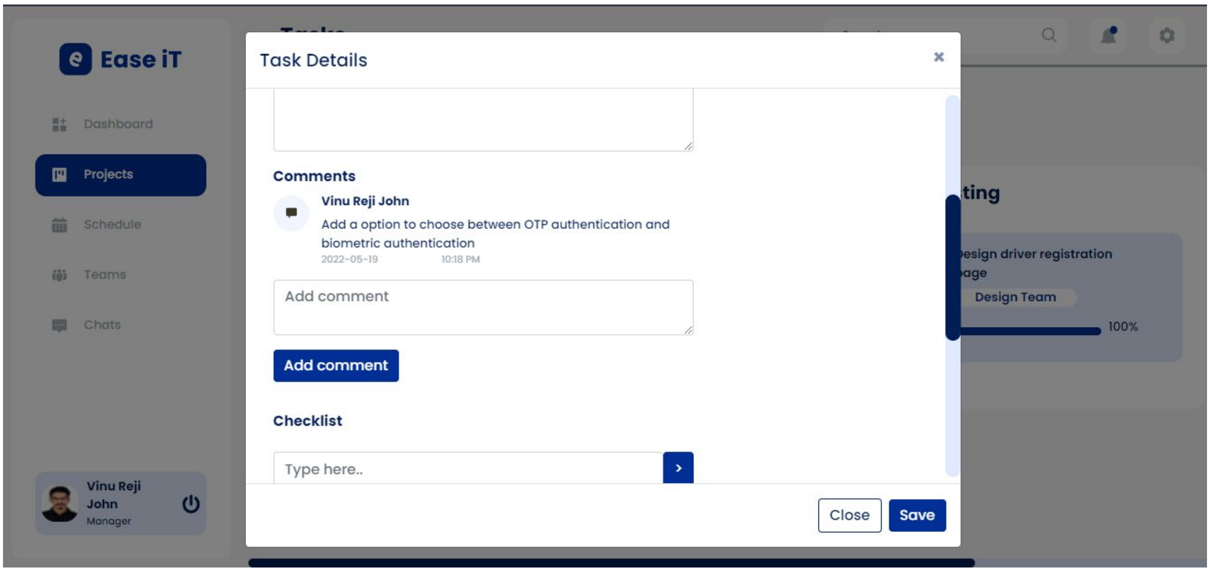
Create Authentication Page

Task description:

Authentication page with otp verification

Comments

Vinu Reji John



Task Details

Comments

Vinu Reji John

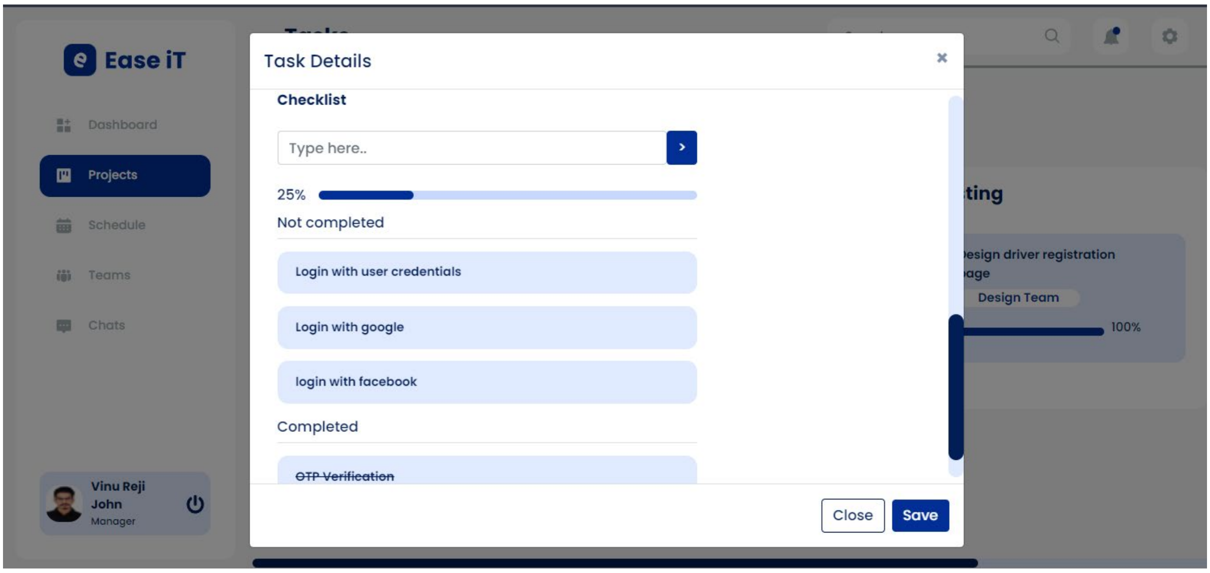
Add a option to choose between OTP authentication and biometric authentication

2022-05-19 10:18 PM

Add comment

Checklist

Type here..



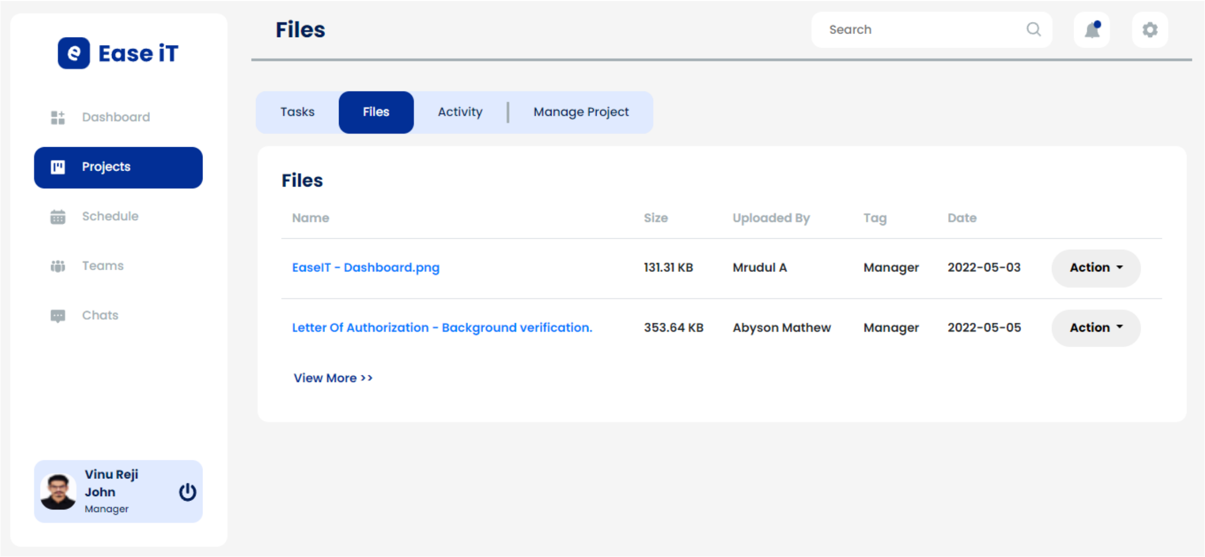
Not completed

- Login with user credentials
- Login with google
- login with facebook

Completed

- ~~OTP Verification~~

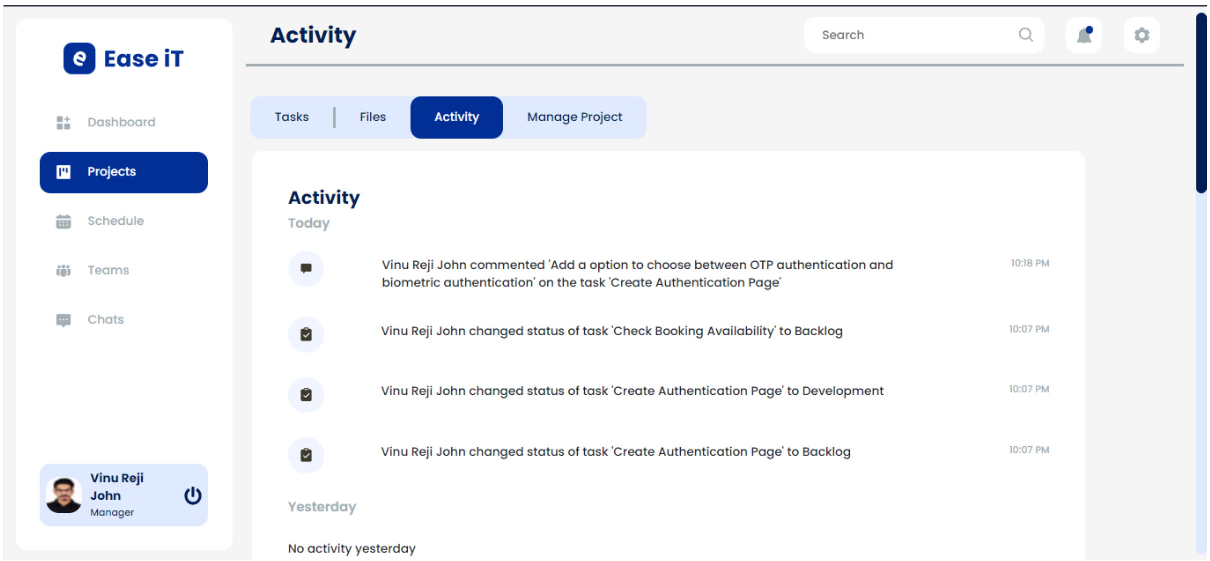
Files



Files

Name	Size	Uploaded By	Tag	Date	
EaseIT - Dashboard.png	131.31 KB	Mrudul A	Manager	2022-05-03	Action ▾
Letter Of Authorization - Background verification.	353.64 KB	Abyson Mathew	Manager	2022-05-05	Action ▾

Activity



Activity

Today

Vinu Reji John commented 'Add a option to choose between OTP authentication and biometric authentication' on the task 'Create Authentication Page' 10:18 PM

Vinu Reji John changed status of task 'Check Booking Availability' to Backlog 10:07 PM

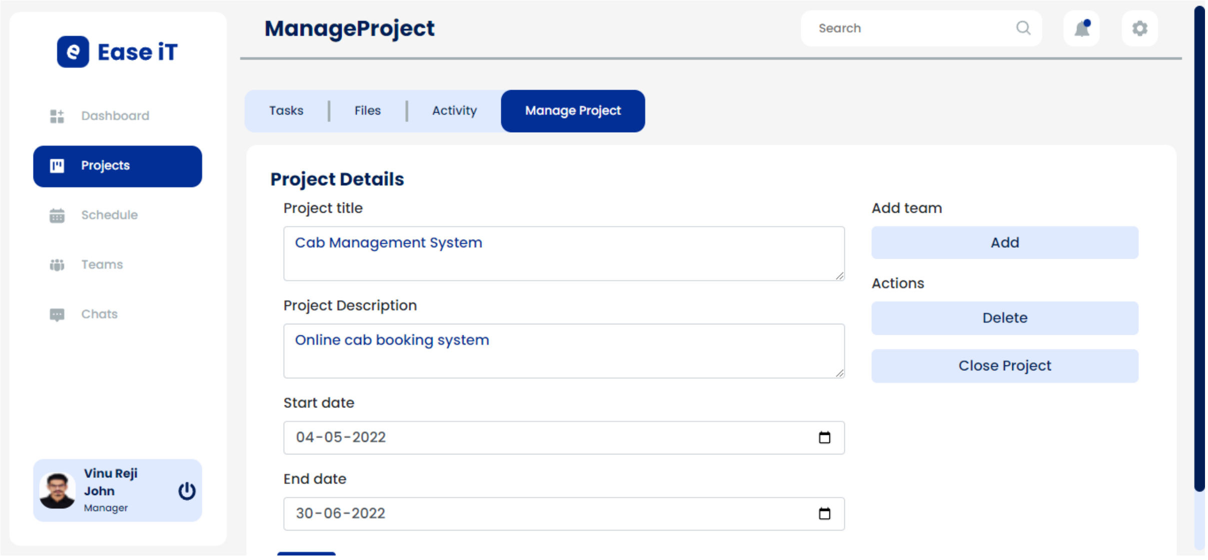
Vinu Reji John changed status of task 'Create Authentication Page' to Development 10:07 PM

Vinu Reji John changed status of task 'Create Authentication Page' to Backlog 10:07 PM

Yesterday

No activity yesterday

Manage Project



Project Details

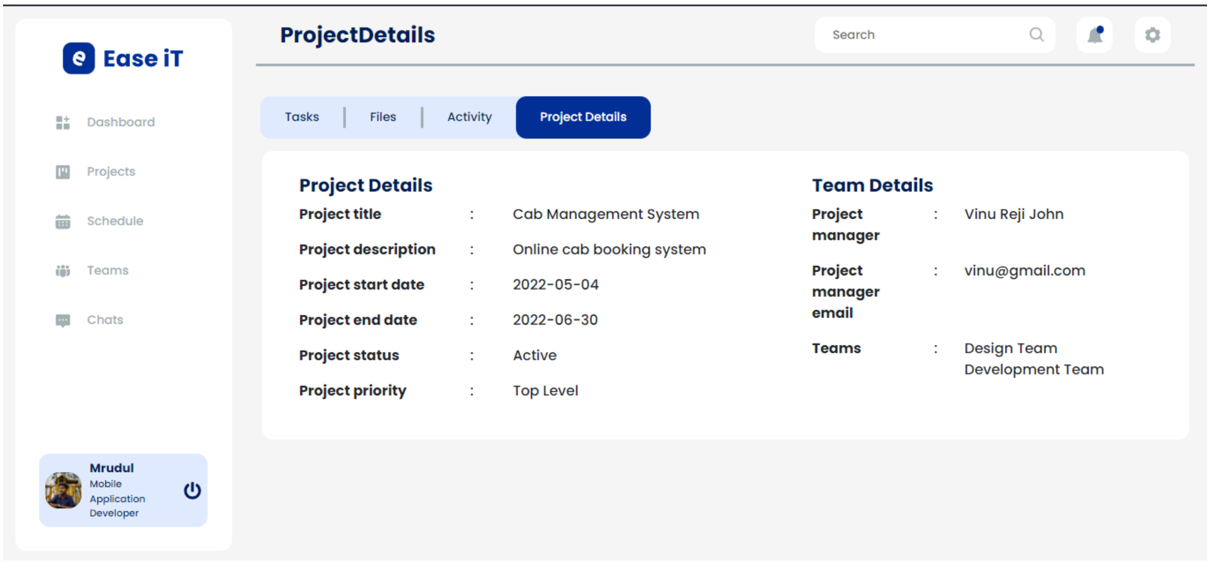
Project title: Cab Management System

Project Description: Online cab booking system

Start date: 04-05-2022

End date: 30-06-2022

Project Details



ProjectDetails

Project Details

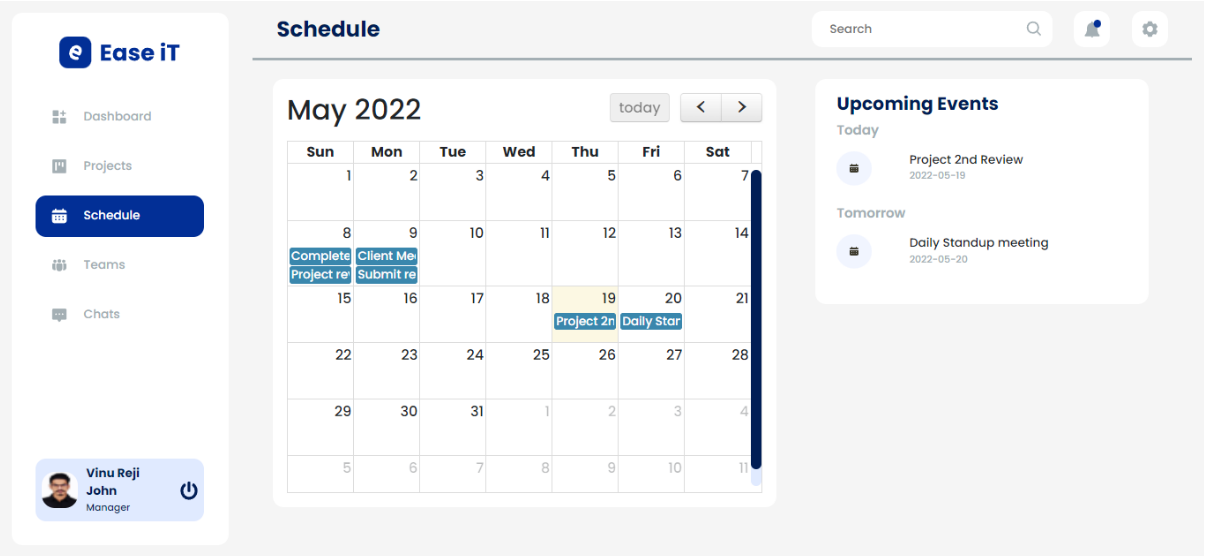
Project title	:	Cab Management System
Project description	:	Online cab booking system
Project start date	:	2022-05-04
Project end date	:	2022-06-30
Project status	:	Active
Project priority	:	Top Level

Team Details

Project manager	:	Vinu Reji John
Project manager email	:	vinu@gmail.com
Teams	:	Design Team Development Team

Mrudul
Mobile Application Developer

Schedule



Schedule

May 2022

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8 Complete Project re	9 Client Me Submit re	10	11	12	13	14
15	16	17	18	19 Project 2n	20 Daily Star	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Upcoming Events

Today

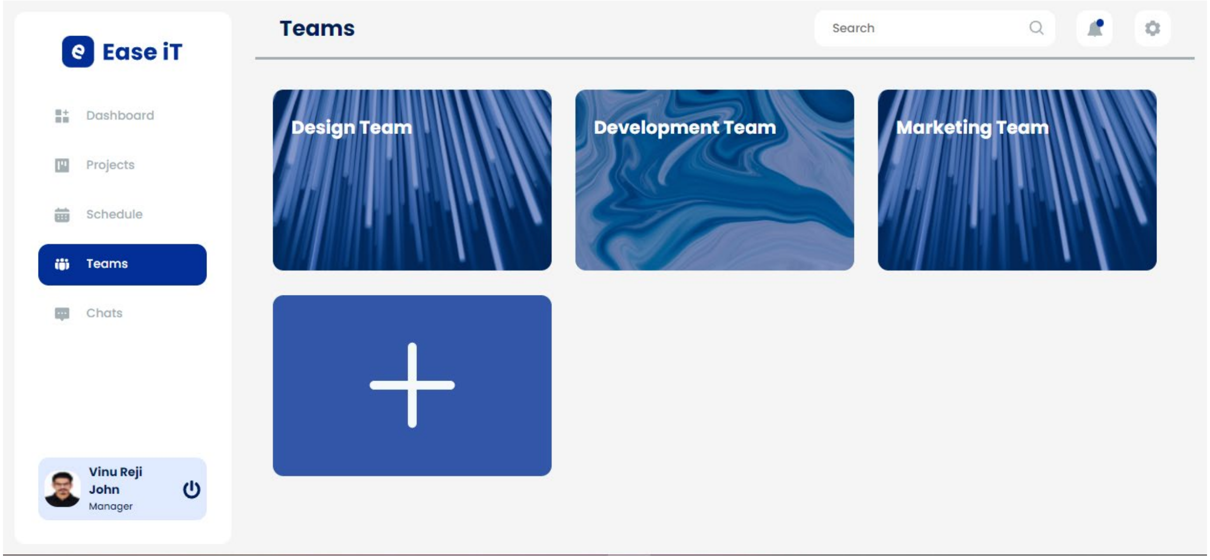
Project 2nd Review
2022-05-19

Tomorrow

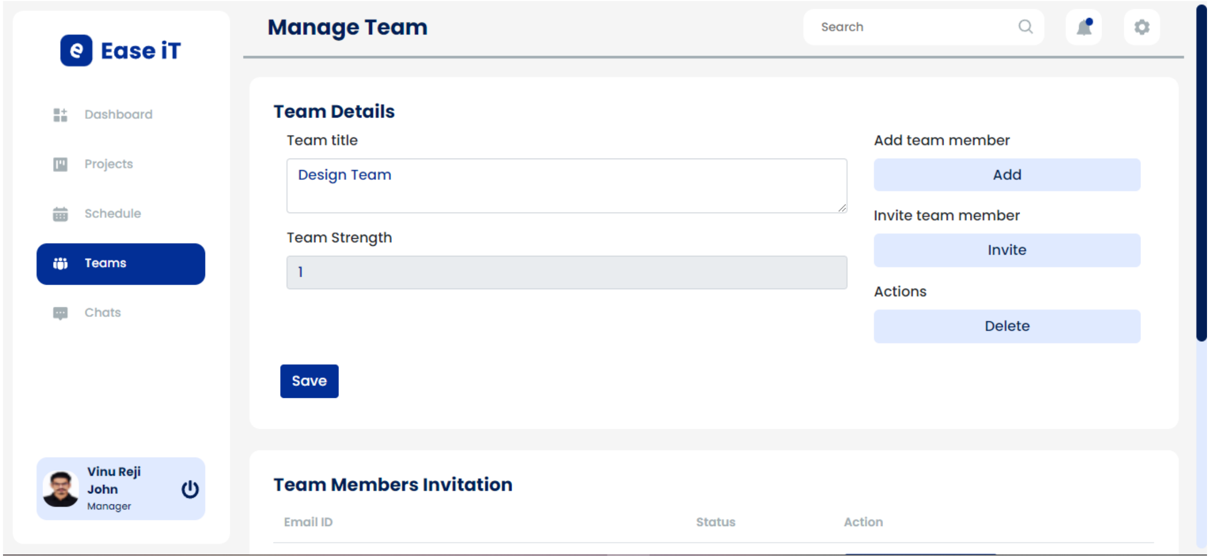
Daily Standup meeting
2022-05-20

Vinu Reji John
Manager

Teams



Manage Team



Manage Team

Team Details

Team title: Design Team

Team Strength: 1

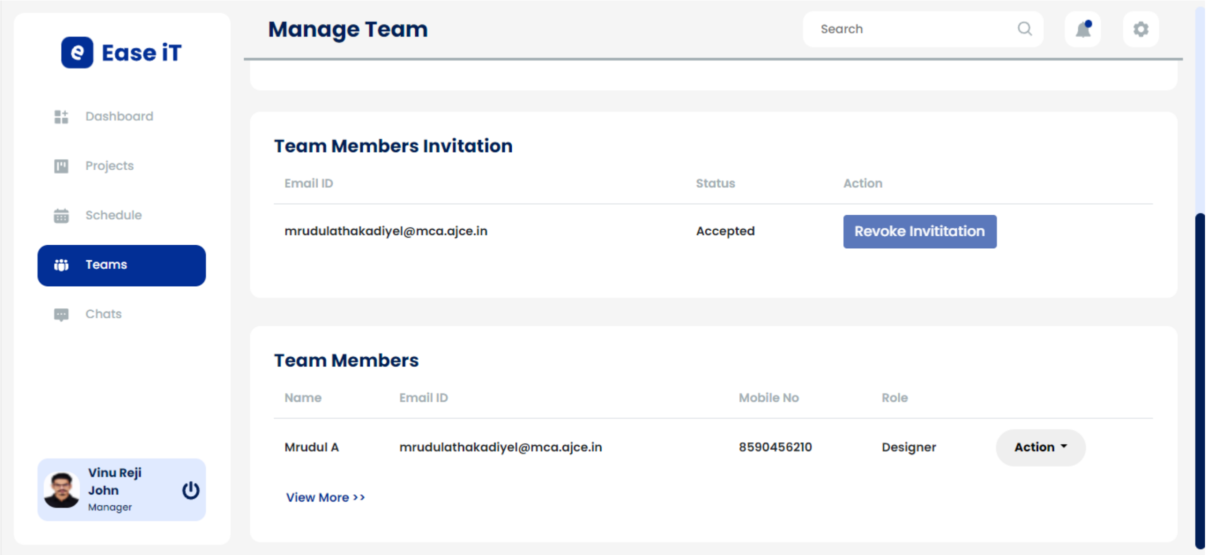
Add team member

Invite team member

Actions

Team Members Invitation

Email ID	Status	Action



Manage Team

Team Members Invitation

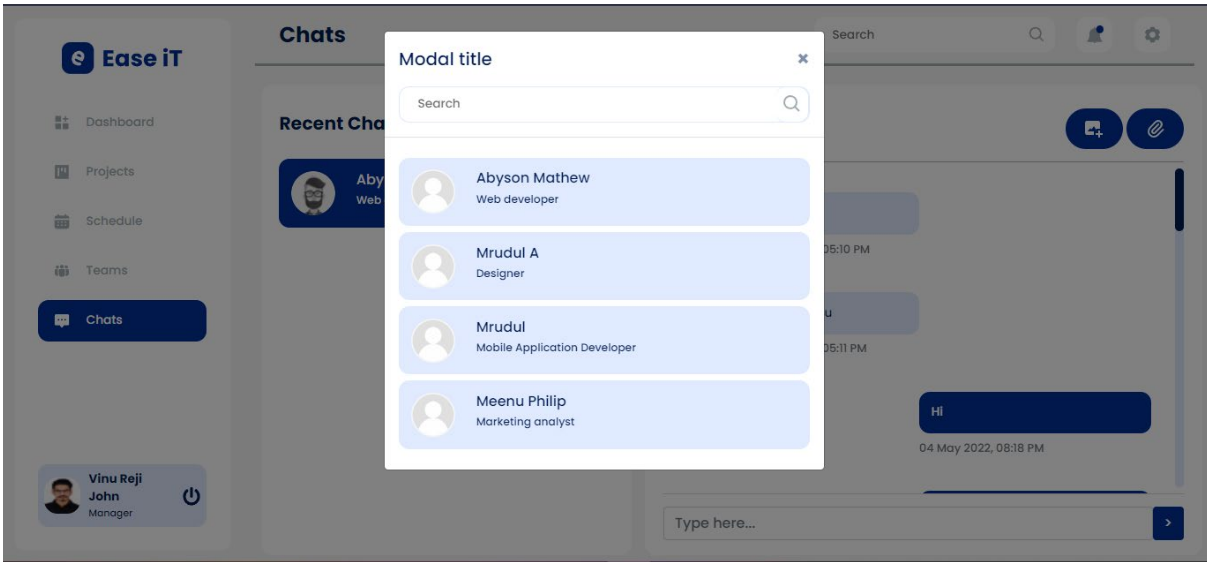
Email ID	Status	Action
mrudulathakadiyel@mca.ajce.in	Accepted	Revoke Invititation

Team Members

Name	Email ID	Mobile No	Role	
Mrudul A	mrudulathakadiyel@mca.ajce.in	8590456210	Designer	Action

View More >>

Chats



Modal title ×

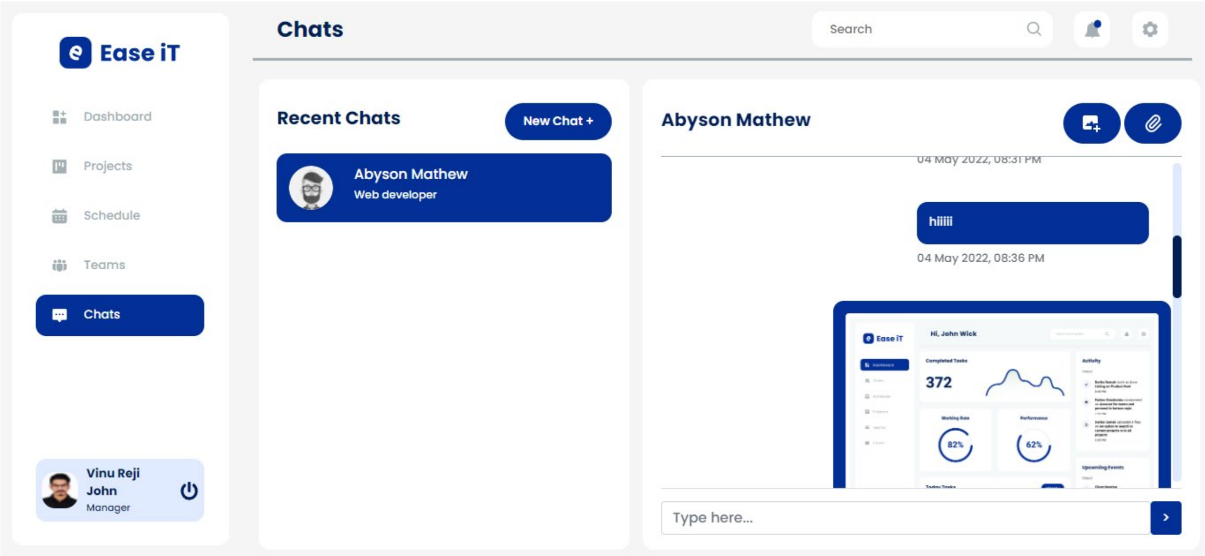
Search

Abyson Mathew
Web developer

Mrudul A
Designer

Mrudul
Mobile Application Developer

Meenu Philip
Marketing analyst

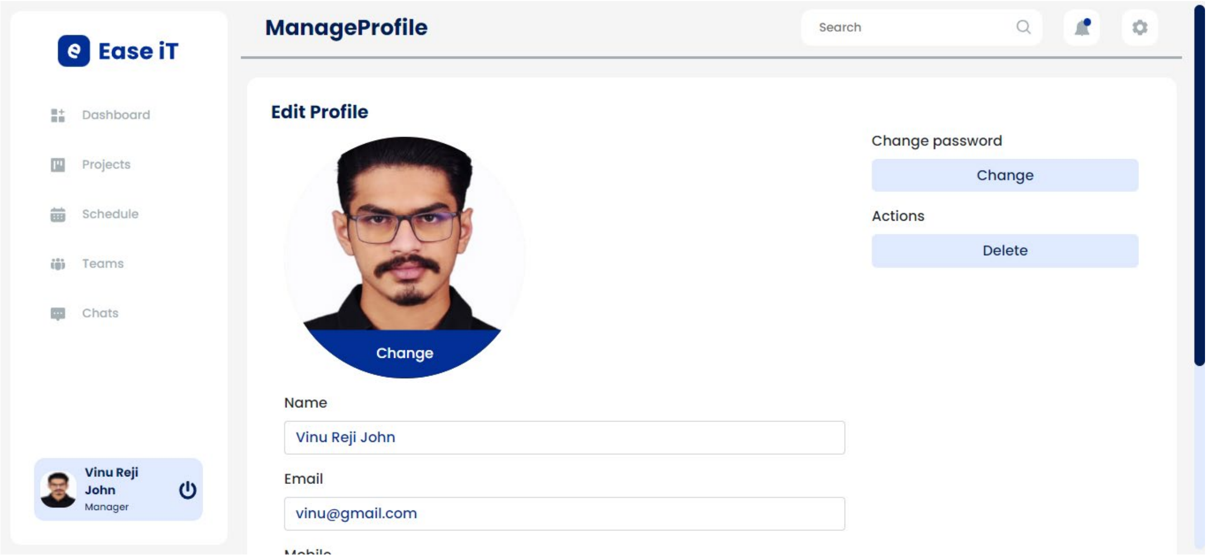


04 May 2022, 08:31 PM

hiiiii

04 May 2022, 08:36 PM

Manage profile



Name

Vinu Reji John

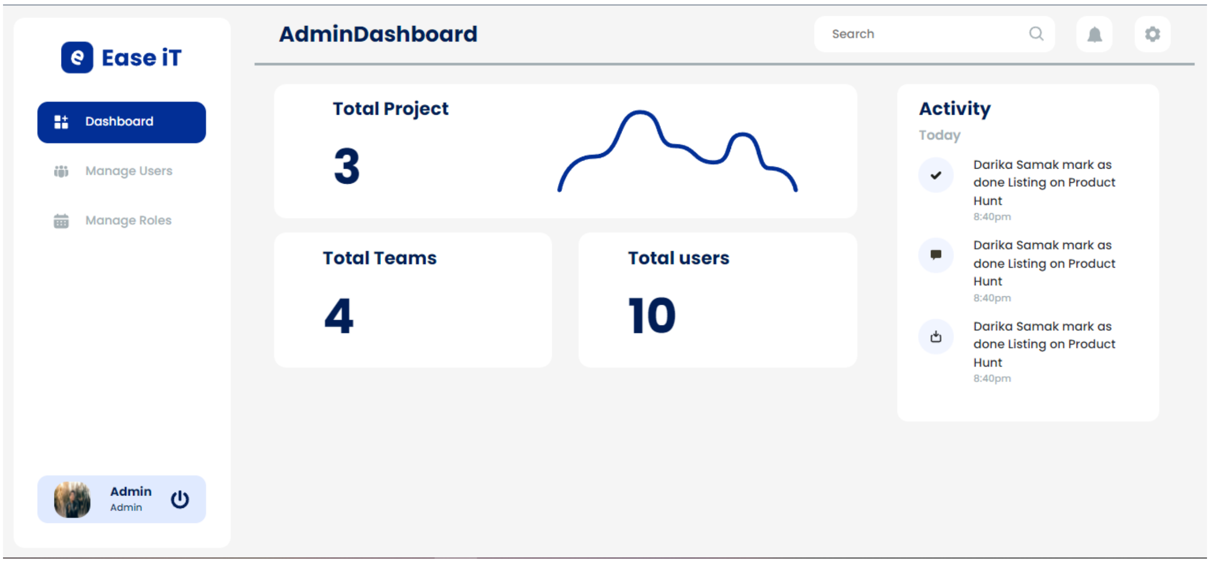
Email

vinu@gmail.com

Mobile

Admin Pages

Admin Dashboard



Total Project

3

Total Teams

4

Total users

10

Activity

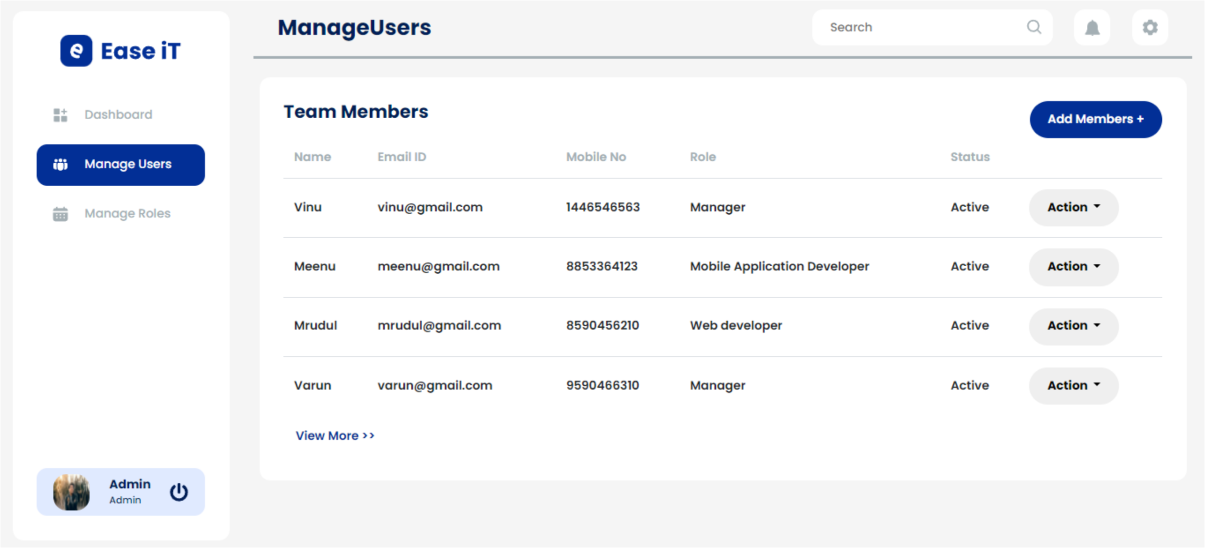
Today

Darika Samak mark as done Listing on Product Hunt
8:40pm

Darika Samak mark as done Listing on Product Hunt
8:40pm

Darika Samak mark as done Listing on Product Hunt
8:40pm

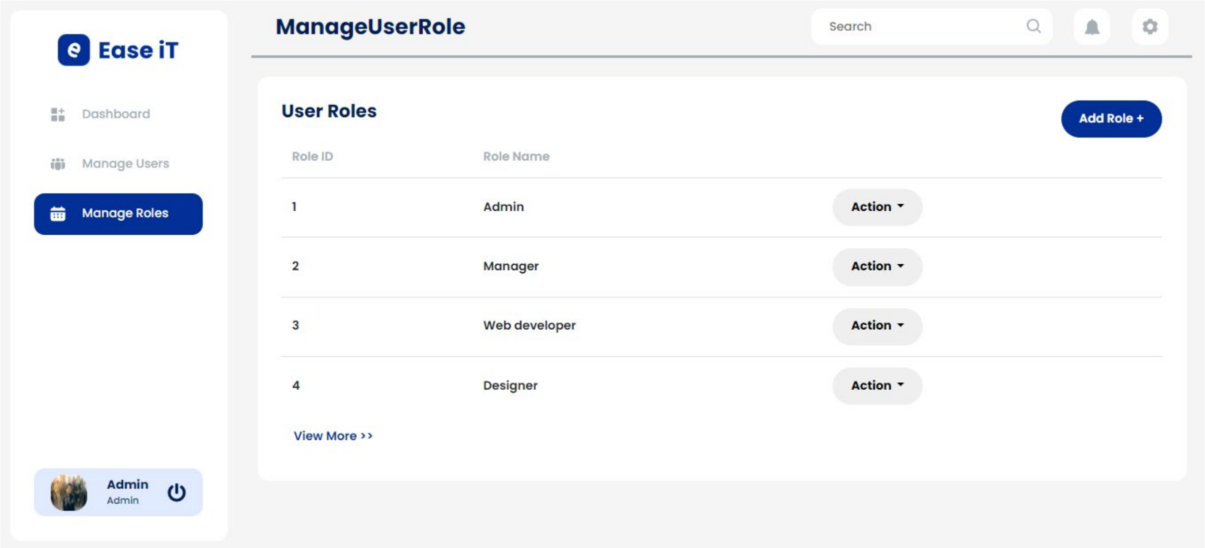
Manage Users



Team Members

Name	Email ID	Mobile No	Role	Status	
Vinu	vinu@gmail.com	1446546563	Manager	Active	Action
Meenu	meenu@gmail.com	8853364123	Mobile Application Developer	Active	Action
Mrudul	mrudul@gmail.com	8590456210	Web developer	Active	Action
Varun	varun@gmail.com	9590466310	Manager	Active	Action

Manage User Role



User Roles

Role ID	Role Name	
1	Admin	Action
2	Manager	Action
3	Web developer	Action
4	Designer	Action