BUILDTECH CONSTRUCTION MANAGEMENT SYSTEM

Project Report Submitted by

Sijimol Cyriac

Reg. No: AJC16MCA-I52

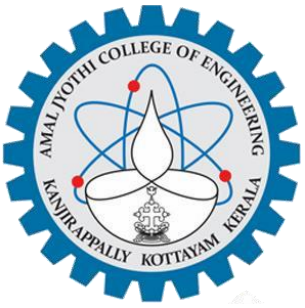
In Partial fulfillment for the award of the degree

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]

2016-2021

AMAL JYOTHI COLLEGE OF ENGINEERING

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

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project entitled **"BuildTech Construction Management System"** is a bonafide record of the work done by **Sijimol Cyriac AJC16MCA-I52,** during the academic year **2016-2021** carried out under our supervision. It is certified that all corrections/suggestions indicated for assessment have been incorporated in the report. The work report has been approved as it satisfies the academic requirements in respect of the project work prescribed by the university for the Integrated Master of Computer Applications Degree. Certified further, that to the best of our knowledge the exact work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this to any other candidate.

Fr. Dr. Rubin Thottupurathu Jose	**Ms. Sona Maria Sebastian**	**Ms. Meera Rose Mathew**
Head of the Department	Project Coordinator	Project Supervisor

External Expert appointed by the university

DECLARATION

I hereby declare that the project report **"BuildTech Construction Management System"** is a bonafide 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 (INMCA) from APJ Abdul Kalam Technological University, during the academic year 2016-2021.

Date..... **Sijimol Cyriac**

KANJIRAPPALLY **Reg. No: AJC16MCA-I52**

ACKNOWLEDGEMENT

First and foremost, I thank Almighty God for his gracious guidance through the project. I take this opportunity to express my gratitude to all those who have helped me in completing the project successfully.

It has been said that gratitude is the memory of the heart. I acknowledge my deep sense of gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** for providing all the infrastructural facilities for us, our Principal **Dr. Z. V. Lakaparampil** for providing good faculty for guidance.

I take the immense pleasure in expressing my thanks to Head of the Department of Master of Computer Applications, **Rev. Fr. Dr. Rubin Thottupurathu Jose**, for his kind patronages in making this project a successful one. I would like to extend my sincere thanks to our coordinator **Ms. Sona Maria Sebastian** and my project guide **Ms. Meera Rose Mathew** for their guidance and cooperation, without which this would not have been a success.

I am indebted to my beloved teachers whose cooperation and suggestions throughout the project which helped me a lot. I also thank all my friends and classmates for their interest, dedication and encouragement shown towards the project. I convey hearty thanks to parents for the moral support, suggestion and encouragement to make this venture a success.

Sijimol Cyriac

ABSTRACT

BuildTech Construction Management System is a web application to manage all day to day operations required by the customer who is constructing an individual house, building, flat etc. It provides information in a quick time according to the requirements that are to be fulfilled. It also provides a convenient method of online construction contracts. This system provides consistency of data and develops the user friendly and interactive website which will reduce the paper works, faster and easy work and save the time. The user can also reduce the time and effort in searching contractors by using this system. Labours can accept job requests from contractor and also contractors can easily find best labours.

In this system there are mainly four users: Admin, Contractor, Labours and Customer. Admin is the one who manages all the accounts and this system. Admin can approve contractors based on an active license and labours based on their aadhar-number. Customers can find the contractors nearby them and can send request to those contractors. Contractor can see the project details that are updated by the customer, he can study the plan and requirements of the customers, so that he can understand the actual requirements of the customer and prepare a detailed estimate of the project and send the estimate to the customer. When the contractor gets a contract he will evaluate whether the labours are available, and if they are available then the contractor assigns a worksite for the labours. This system is made to help the customers for an easy and convenient way of construction and also helps contractors to increase their work and income source. This system has all the features that are required by the customer who is constructing any building and also has all the functions that are needed by the contractors to complete their work fastly.

CONTENT

CONTENT

LIST OF ABBREVIATIONS

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modeling Language

INTRODUCTION

BuildTech Construction Management System is a web application to manage all day to day operations required by the customer who is constructing an individual house, buildings, offices etc. It provides information in a quick time according to the requirements that are to be fulfilled. It also provides a convenient method of online construction contracts. This system provides consistency of data and develops the user friendly and interactive website which will reduce the paper works, faster and easy work and save the time. The customer can also reduce the time and effort in searching contractors by using this system. Labours can accept job requests from contractors and also contractors can easily find best labours.

The version control system used during the development time was Git. Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

1. USING GIT AS A VERSION CONTROL SYSTEM

1.1 Introduction to GitHub

GitHub is a web-based version-control and collaboration platform for software developers. GitHub, which is delivered through a software-as-a-service (SaaS) business model, was started in 2008 and was founded on Git, an open source code management system created by Linus Torvalds to make software builds faster. And it is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers. GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Each public or private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private.

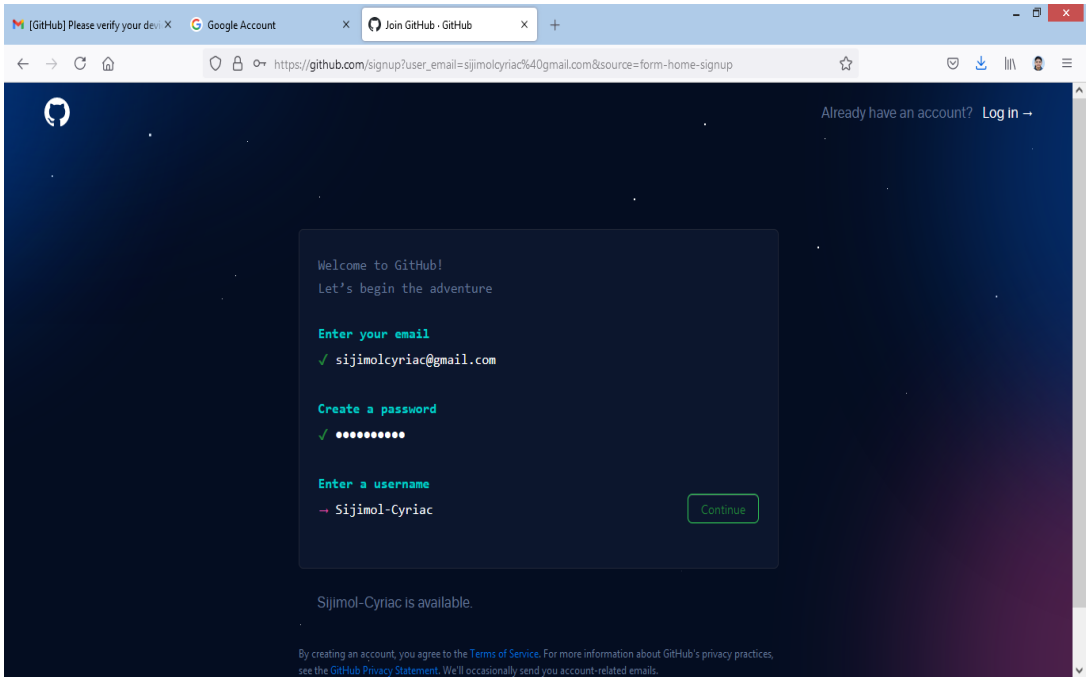
GitHub facilitates social coding by providing a web interface to the Git code repository and management tools for collaboration. GitHub can be thought of as a serious social networking site for software developers. Members can follow each other, rate each other's work, receive updates for specific projects and communicate publicly or privately.

GitHub products and features

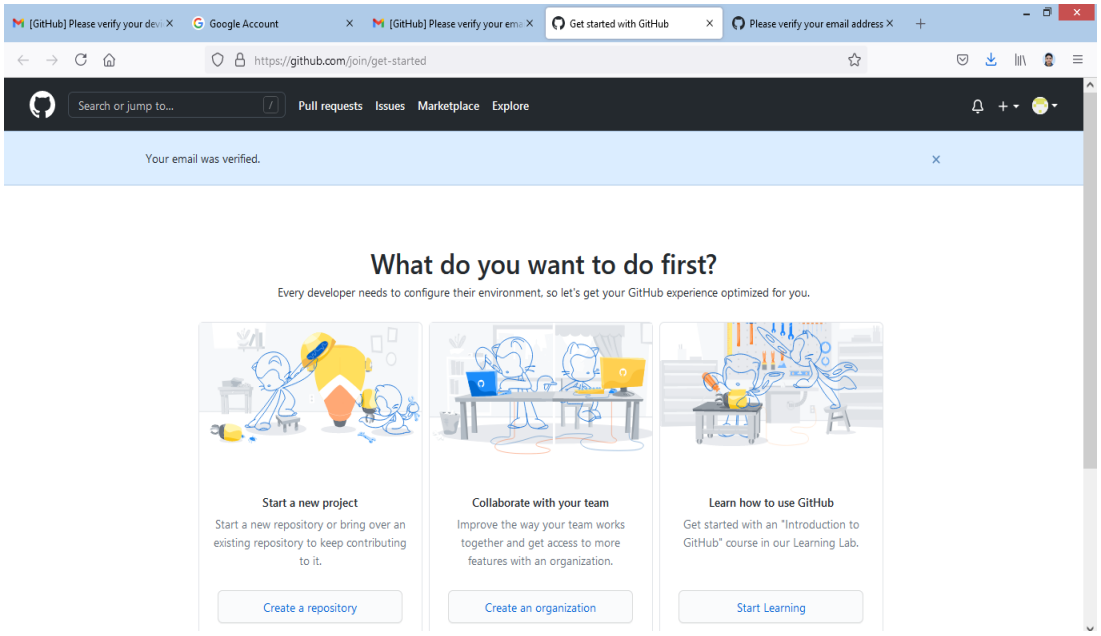
GitHub offers an on-premises version in addition to the well-known SaaS product. GitHub Enterprise supports integrated development environments and continuous integration tool integration, as well as a litany of third-party apps and services. It offers increased security and auditable than the SaaS version.

1.2 Working with Git

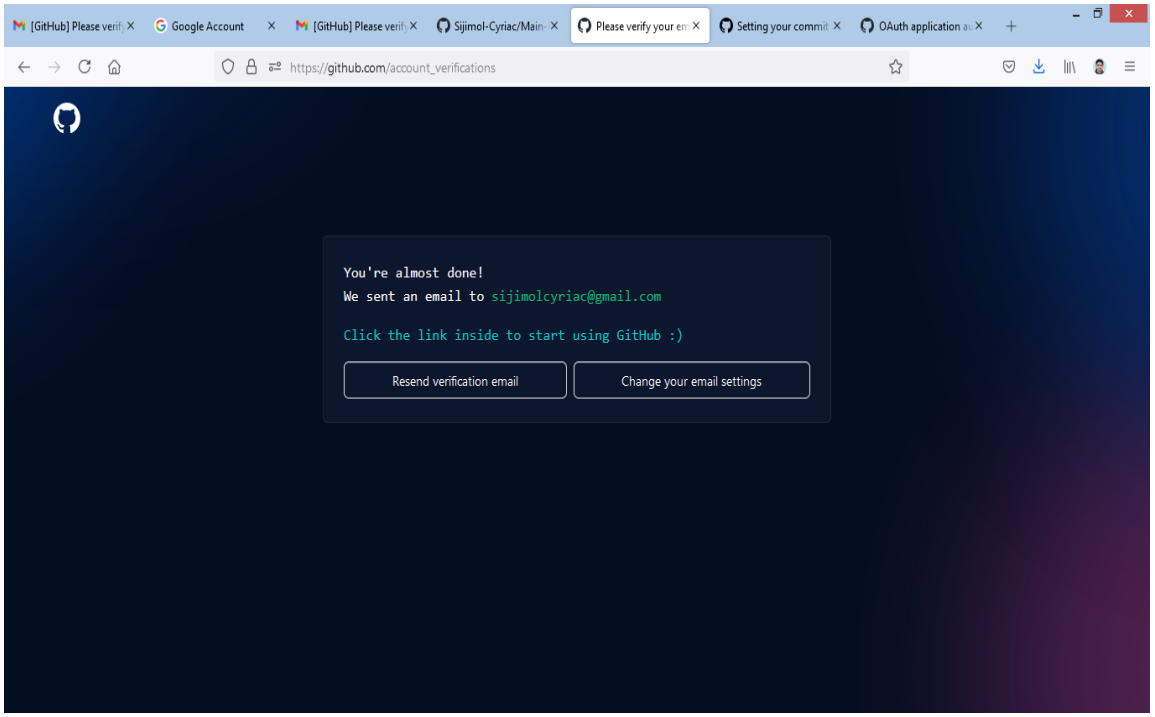
- Click on sign up button and create an account in GitHub.com



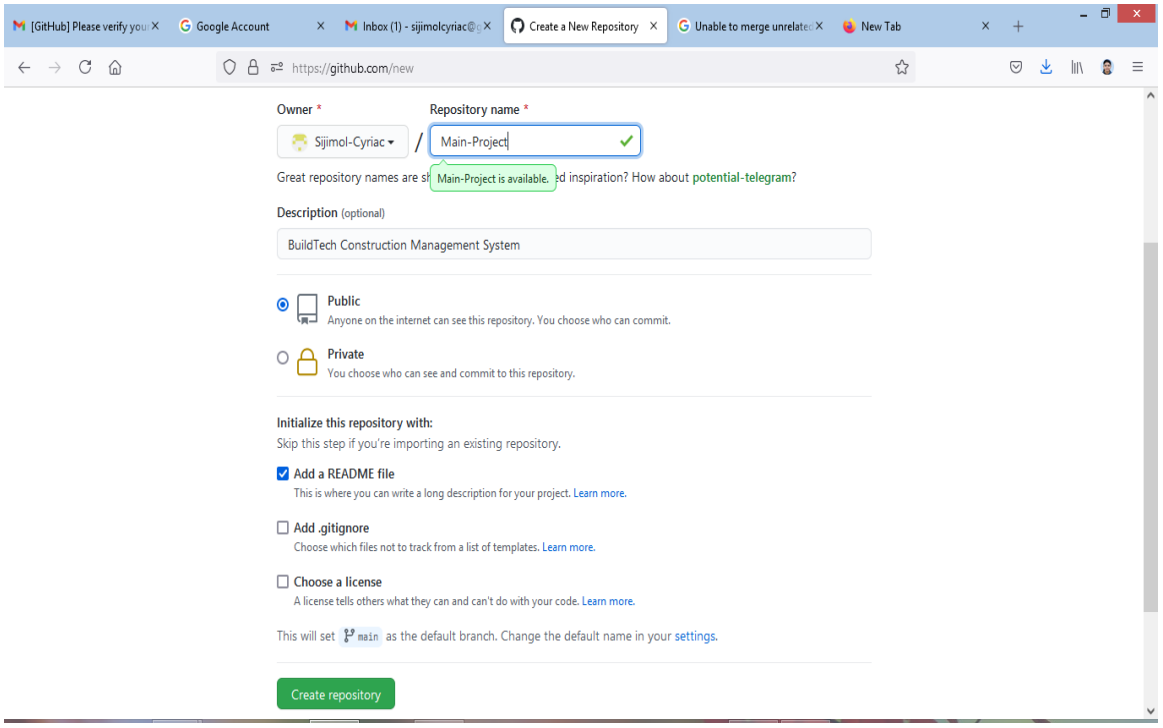
- Once successfully sign in, set up personal account, and choose your plan by selecting the options given below based on our purpose.



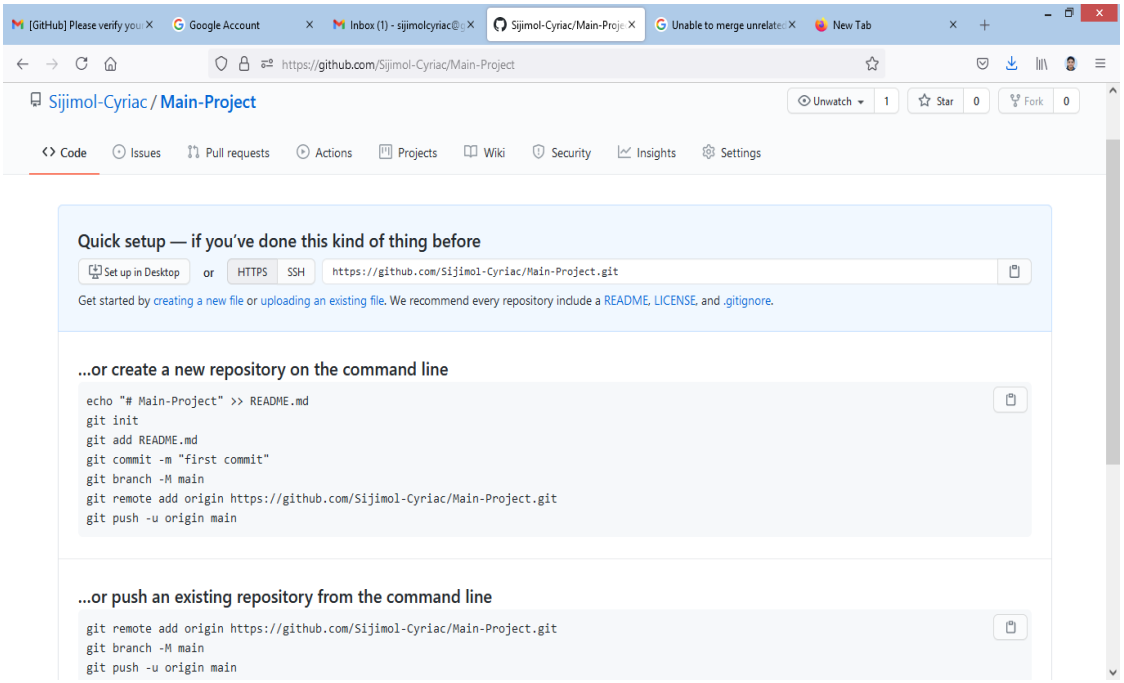
- Once this step is completed then verify the email address and you can access the home page



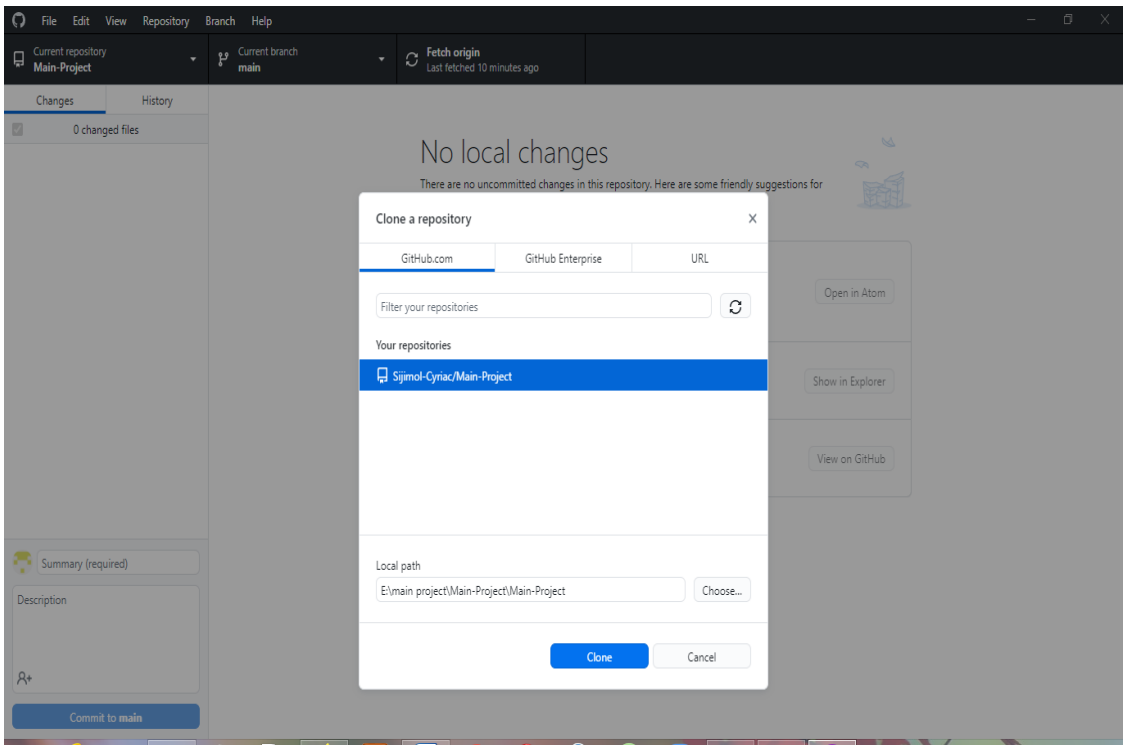
- Create a repository in GitHub



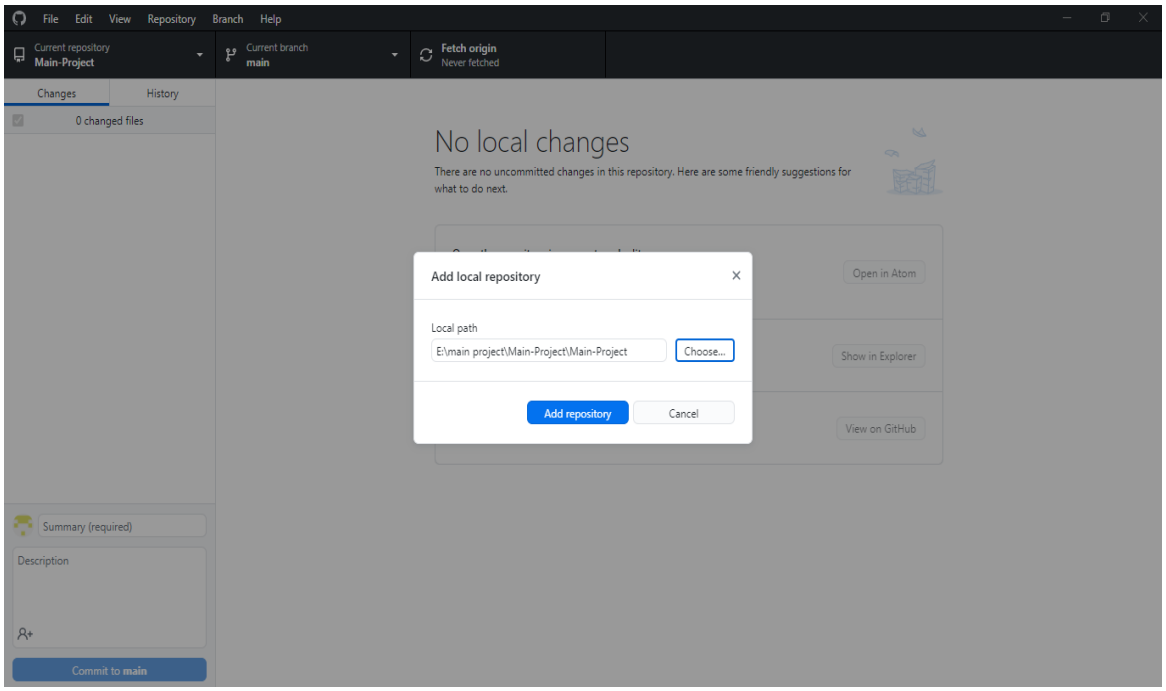
❖ Once repository is completed, you can setup the repository



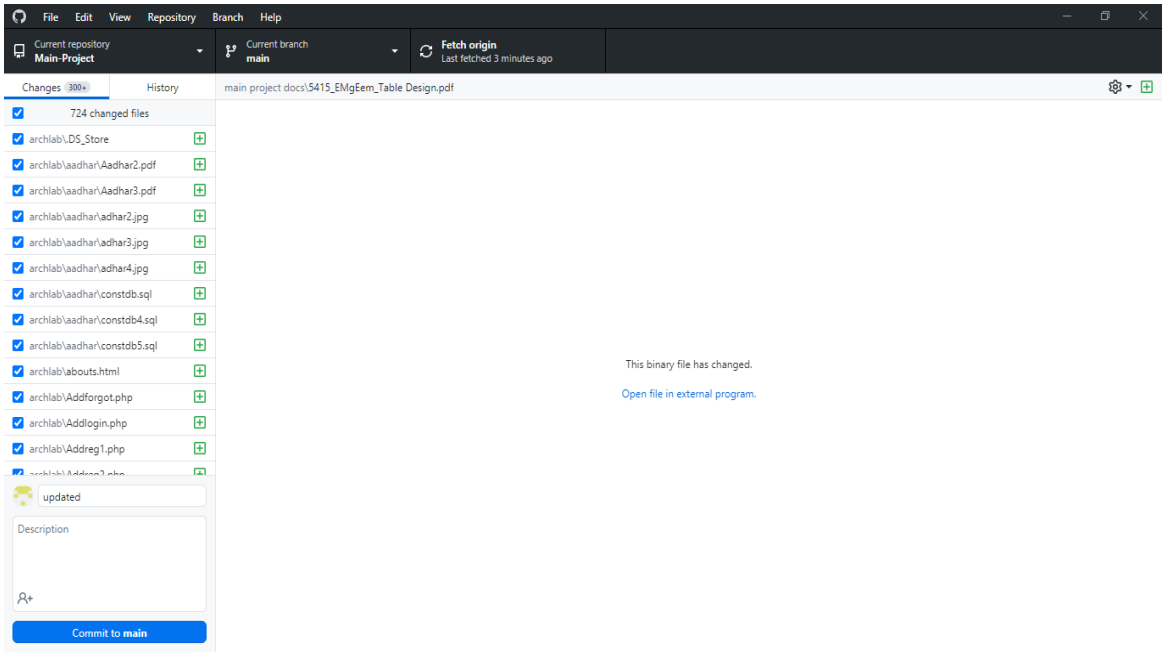
❖ Download and install GitHub desktop application. Once installed Go to Repository > Clone repository, and select your repository which was created in GitHub.com or Select a local system folder.



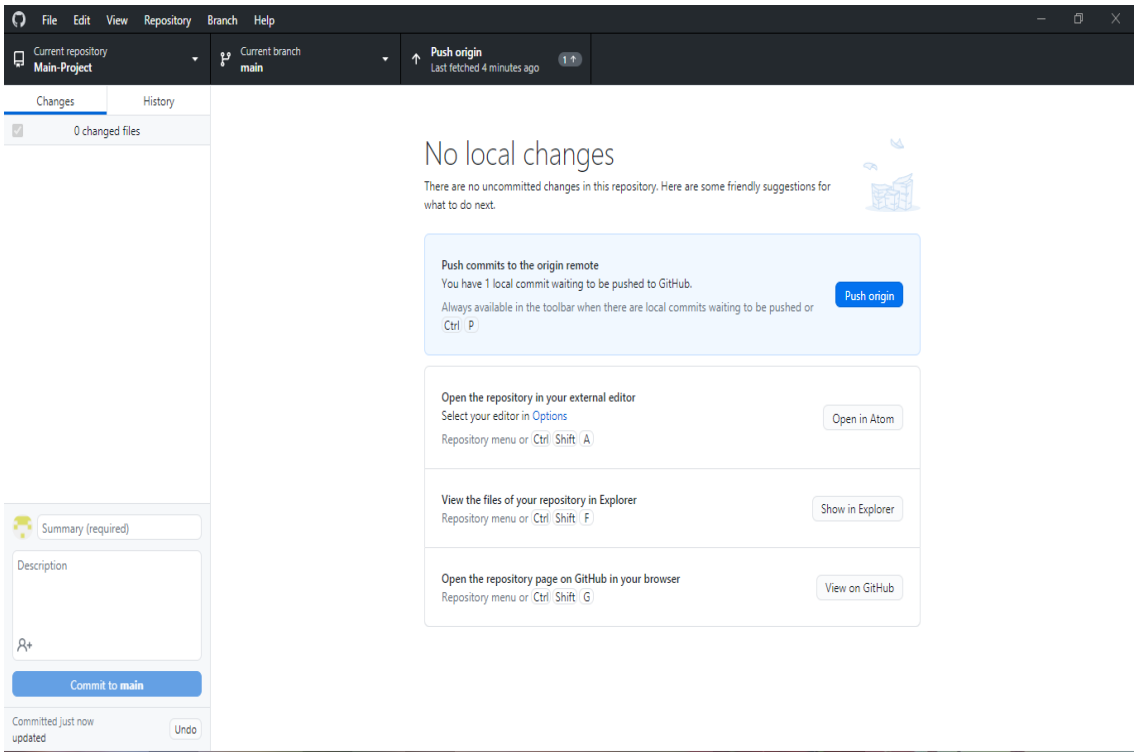
❖ Add local system folder, then create a local repository with folder name.



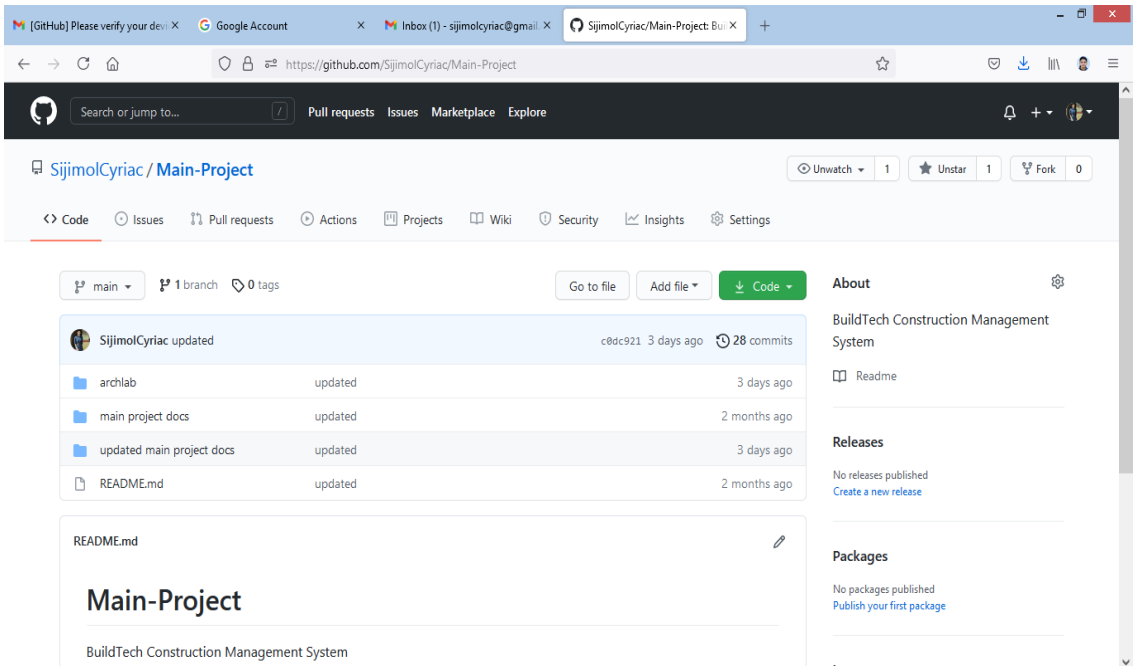
❖ Once repository is created, make changes on the file which is stored inside the local repository and commit to main.



❖ Push the local repository to origin



❖ Refresh GitHub.com repository to fetch commits



2. PROJECT DOCUMENTATION

2.1 INTRODUCTION

2.1.1 Project Overview

"BUILDTECH CONSTRUCTION MANAGEMENT SYSTEM" is a web application which is meant to help the customers to make their construction easy. The customer can also reduce the time and effort in searching contractors by using this system. Labours can accept job requests from contractors and also contractors can easily find best labours. The proposed system includes four users they are administrator, customer, contractor and labour. Registered customers can login to the site and can upload project details he wants to construct and can also make the payment to the contractor through online. The customer can also access the daily progress of the work through weekly progress report option. Contractor can see the project details that are updated by the customer, he can study the plan and requirements of the customers, so that he can understand the actual requirements of the customer and prepare a detailed estimate of the project and send the estimate to the customer. The administrator has the central control over the whole system. When the contractor gets a contract he will evaluate whether the labours are available, and if they are available then the contractor assigns a worksite for the labours. Contractors add wages to labours. Administrators do the verification of contractor and labour identities, approve contractor and labour, add labour categories, services, state, district etc.

2.1.2 Project Specification

The proposed system is made to help the customers for an easy and convenient way of constructing a house and also helps contractor's to increase their work and income source. We will also provide users to view the contractor details, payment details, complaint details etc.

The system includes 4 modules. They are:

1. **Admin Module**

   The site admin have an overall control on the website. He can perform the functionalities like,

   - Login to the Application
   - Add/View/Update/Delete labour category and service
   - Add/View/Update/Delete state and district

- View registered contractors, labours and customers
- Activate/Deactivate registered contractors and labours
- Manage profile and password

2. Customer Module

Customer can perform functionalities like,

- Registration/Login and View/Manage profile/Change password.
- Search contractors nearby them
- Upload project to those contractors
- View/Approve estimation details
- View/Approve weekly progress report details
- Post complaints to the contractors
- Payment with Razorpay gateway

3. Contractor Module

A contractor can manage their construction work effectively by this application. They can perform the functionalities like,

- Registration/Login and View/Manage profile/change password
- View/Approve/Reject project of customers
- Prepare detailed estimation of project
- Allocating duties for labours
- View/Approve leave and attendance of labours
- Add wages to labours
- View/Replay complaints from customer.
- Generate weekly progress report based on a given from and to date.

4. Labour Module

Labour can register into this system based on their labour category. After registration, he will get notification from contractor about worksite duties and get paid by the contractor. They can perform the functionalities like,

- Registration/Login and View/Manage profile/change password
- View/Accept/Reject contractor request
- Add attendance and leave

2.2 SYSTEM STUDY

2.2.1 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 analyzed. 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, analyzing 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.

2.2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Means, all the records of about customers, contractors, labours, etc. are stored manually into the file registers, which is too lengthy and tedious. They have to maintain that record for long duration, but maintaining all these records as manually is very difficult task when the organization is very big. Modifying the records is not very easy, since for a minute change the whole record would be required to re-enter. To make and maintain the different reports about construction work is also very tough and tedious job. Information is stored into registers, which requires large storage space. Searching particular record is very tedious job, since as a time passes the condition of the registers becomes bad. Every year, they have to maintain new registers. The current system is less convenient in managing stakeholders including plan approval, less transparency, no standardized packages, project often delayed with no progress visibility, unskilled labour force and unmonitored quality of material and not all technology based. The current system has too much manual work and increases burden on workers but doesn't provide good result. The proposed system rectifies the drawbacks of the present system. It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can upload their project to contractors by viewing the profile details, year of experience etc. of the contractors.

Drawbacks of existing system

- Less convenient in managing project details including plan approval, less transparency, no standardized packages.
- Project often delayed with no progress visibility.
- Unskilled labour force and unmonitored quality of material and not all technology based.
- Human effort is needed.
- Customers do not get a chance to identify the quality of materials used.

2.2.3 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth; on such consideration the system is proposed. In our proposed system there is admin who can view all the contractors, labours and customers. It allows customers to upload their projects and do their transactions by using online payment method .Users of this proposed system are admin, customer, contractor and labour. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Our website is multifunctional which includes customer introduction, contractor introduction, project details etc. The master file that is file which contains all the details of the data's which are kept for long time is contractor master and customer master. The contractor master contains all the details of completed projects, on-going projects, rates and details of different categories of house, daily progress report, payment options etc. The customer master contains personal details of the customer like customer id, customer name, profession, residential address, etc. This system is designed to help the customers to make the construction easy. The proposed system provides consistency of data and reduces the paper work. Also, the customer can easily search the contractor by selecting state and district. This system helps the customer to fulfill his dream project construction with the help of a contractor through online. This system helps the customer to get its project done smoothly, efficiently and in less time and it helps the contractor to increase the work and income source. This system is made to help both the customer and the contractor.

Advantages of proposed system

The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- *You are able to get creative and innovative project plans:-*

  Our customers will get creative project plans with affordable price and they have the option to view contractor profile and can send feedbacks. Also contractors can able to do proper construction management and can also improve and increase their work and income source.

- *Better security: -*

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

- *Ensure data accuracy: -*

The proposed system eliminates the manual errors while entering the details of the users during the registration.

- *Better service: -*

The system will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

- *User friendliness and interactive: -*

The proposed system's interface helps the users to perform their operations without any confusions or difficulties. A customer can easily search contractors nearby them and can upload project to them. A contractor can get notification when a new project request, or complaint arrives and which helps the contractor to immediately take an action on it.

- *Minimum time required: -*

The customers can search contractors by selecting the state and district and view their profile which helps to upload a new project with minimum time. Similarly contractors can manage their construction work and other operations easily using this website.

2.3 REQUIREMENT ANALYSIS

2.3.1 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.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

2.3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, BuildTech Construction Management System was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

2.3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input , output , programs and procedures. Having identified an outline system, the

investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

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

The project should be developed such that the necessary functions and performance are achieved within the constraints. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using HTML, CSS in front end and PHP, MySQL in server back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB.

2.3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

BuildTech Construction Management System, GUI is user friendly so that users can easily use it without any training.

2.4 Requirement Modeling

2.4.1 UML diagram

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling 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. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Activity diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

1. UML Activity Diagram

Activity Diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not.

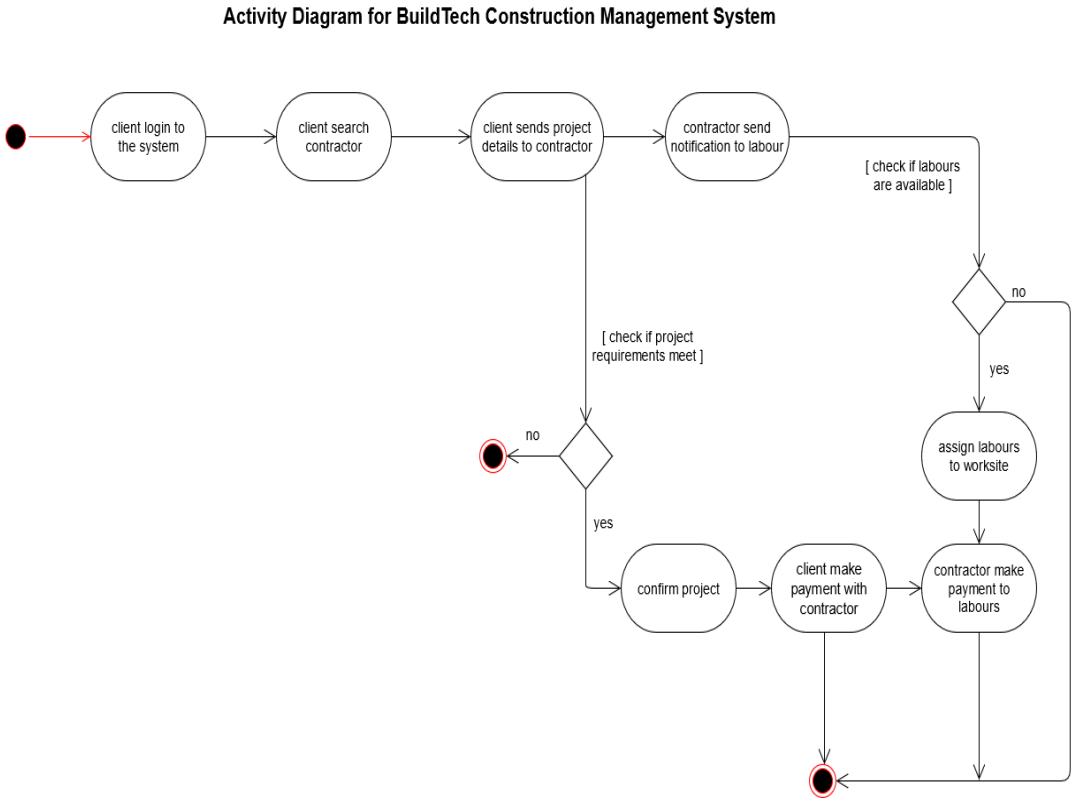


Fig. Activity diagram for BuildTech Construction Management System

2. UML 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 Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. 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 the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

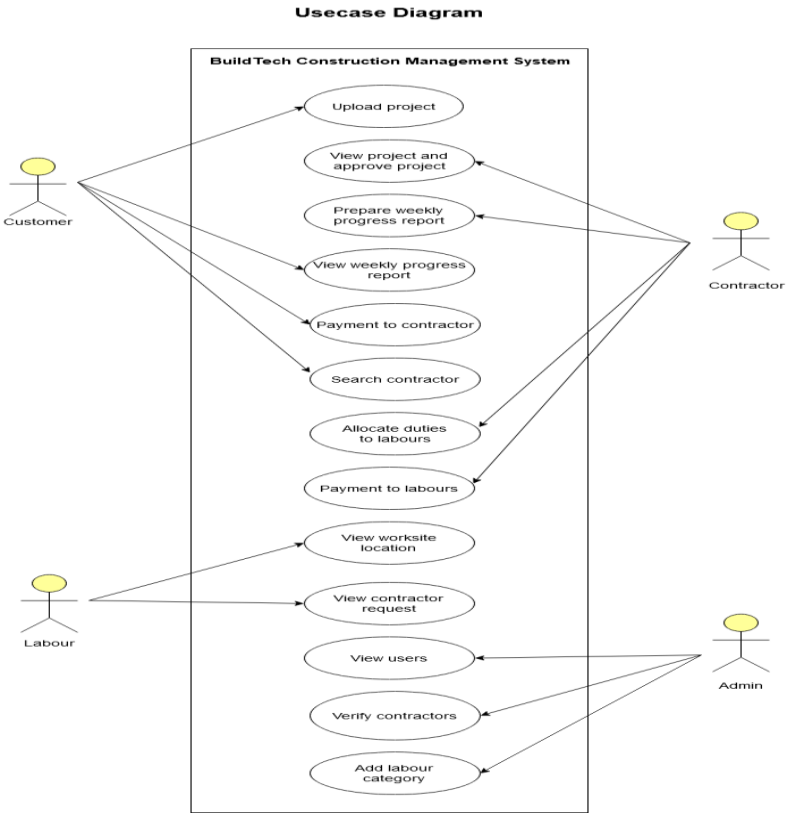


Fig. Use case diagram for BuildTech Construction Management System

3. UML Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

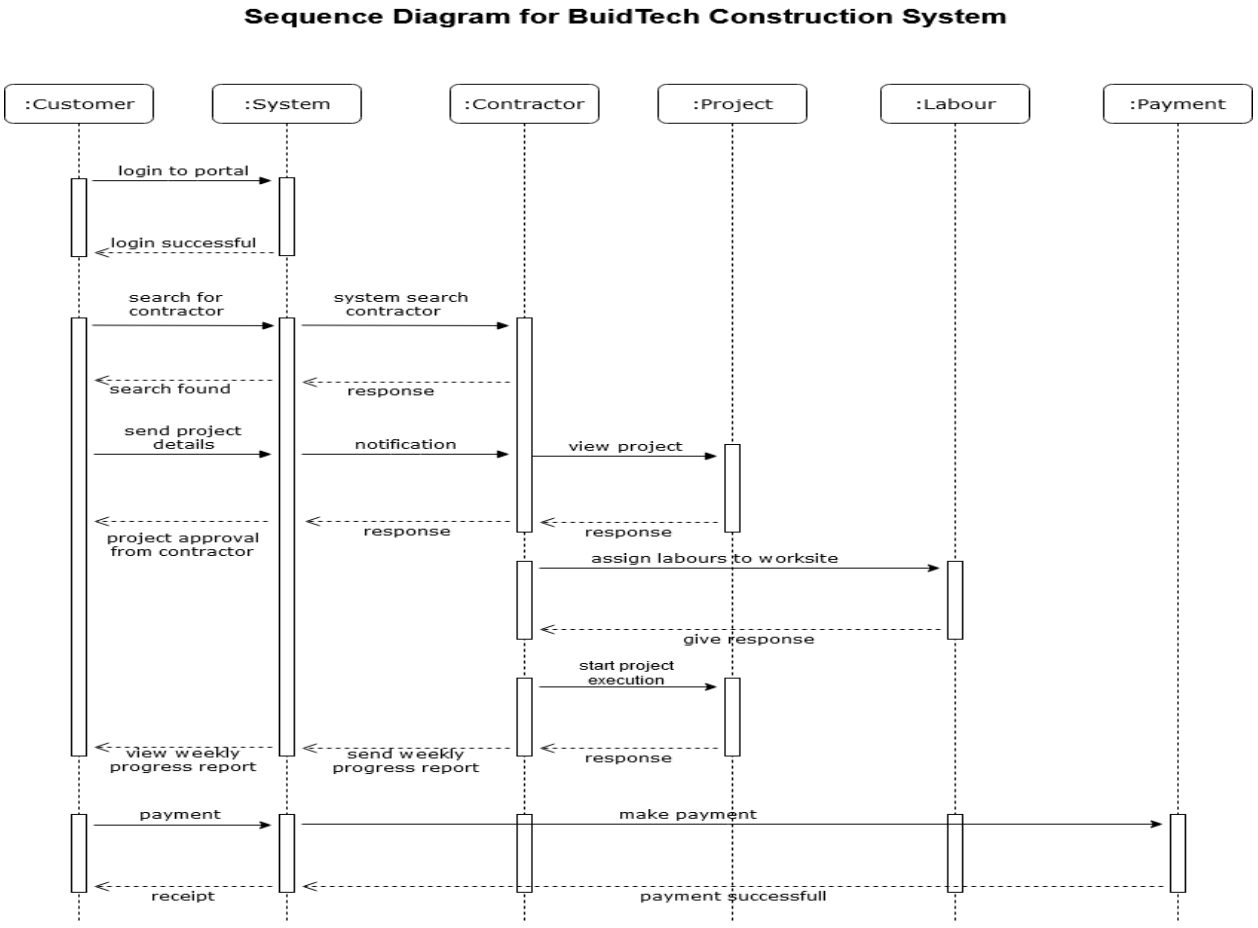


Fig. Sequence diagram for BuildTech Construction Management System

4. UML 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.

Notations of a Collaboration Diagram

Following are the components of a component diagram that are enlisted below:

1. **Objects:** The representation of an object is done by an object symbol with its name and class underlined, separated by a colon. In the collaboration diagram, objects are utilized in the following ways:
    - The object is represented by specifying their name and class.
    - It is not mandatory for every class to appear.
    - A class may constitute more than one object.
    - In the collaboration diagram, firstly, the object is created, and then its class is specified.
    - To differentiate one object from another object, it is necessary to name them.
2. **Actors:** In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.
3. **Links:** The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.
4. **Messages:** It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases.

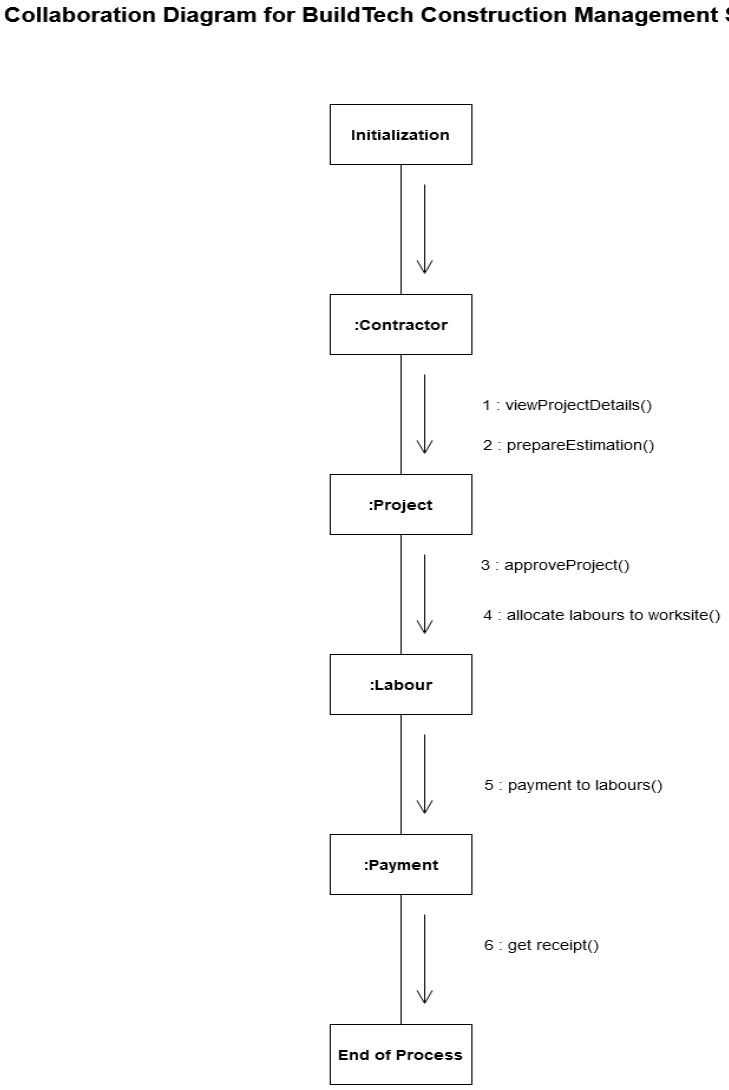


Fig. Collaboration Diagram for BuildTech Construction Management System

5. UML State Chart Diagram

It describes different states of a component in a system. The states are specific to a component/object of a system. A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the

reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Statechart 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 Statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart 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.

Statechart Diagram for BuildTech Construction Management System

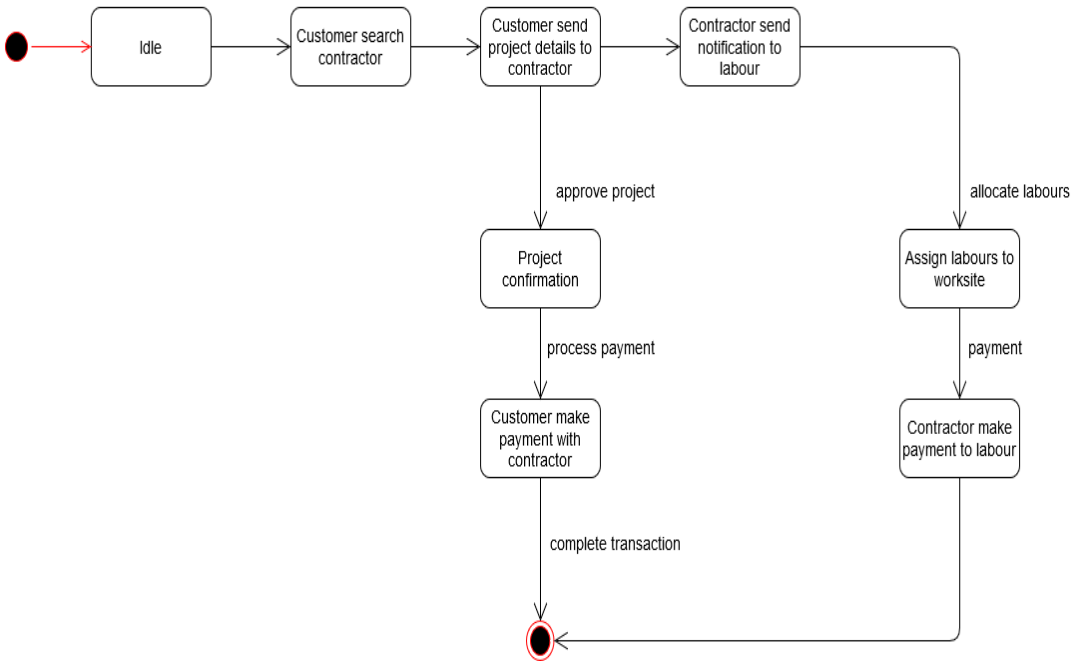


Fig. Statechart Diagram for BuildTech Construction Management System

6. UML 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 modeling 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 purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

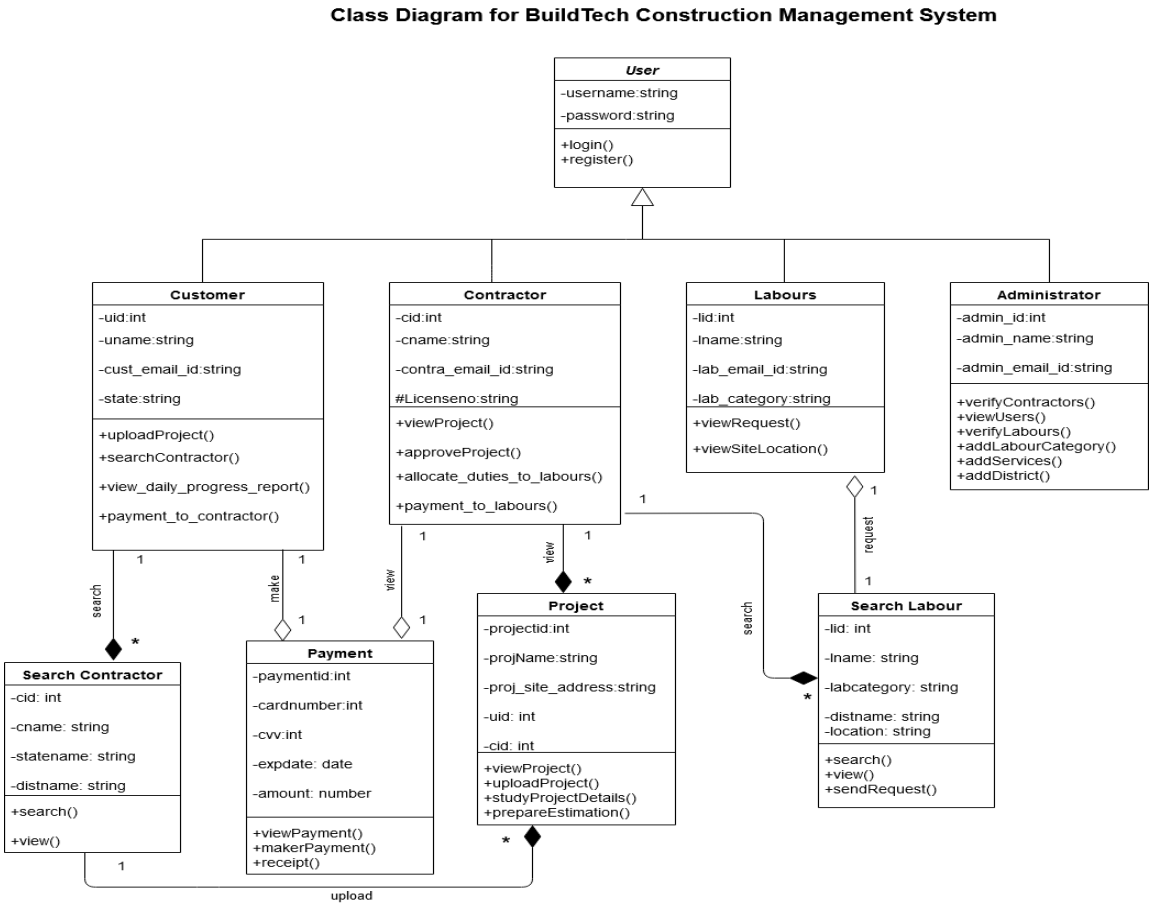


Fig. Class Diagram for BuildTech Construction Management System

7. UML Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams.

Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

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 behavior and their relationship from practical perspective

Object Diagram for BuildTech Construction Management System

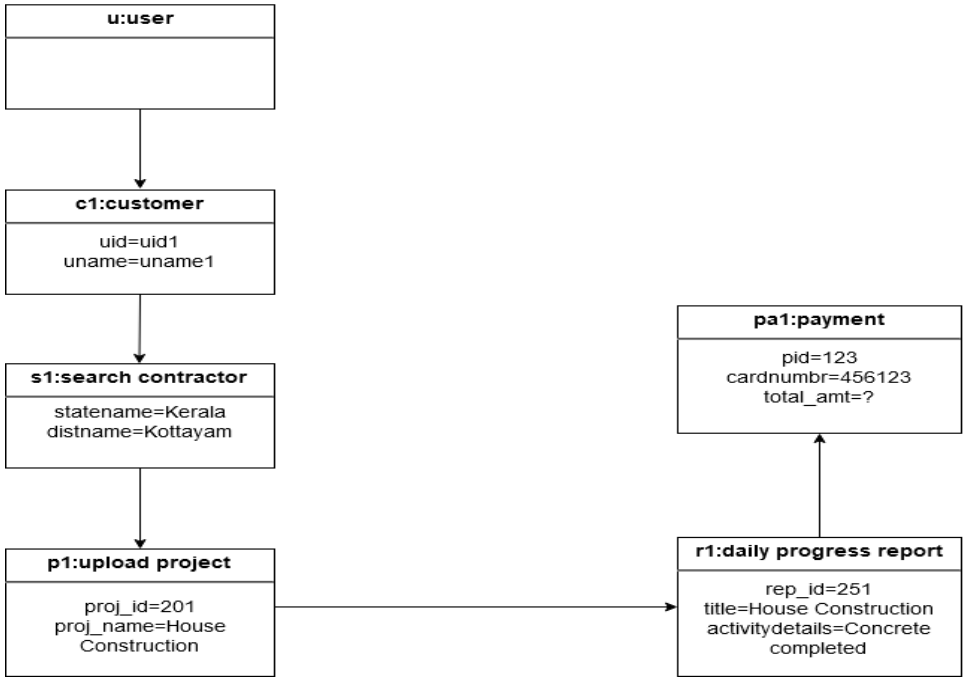


Fig. Object diagram for BuildTech Construction Management System

8. UML Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as –

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.

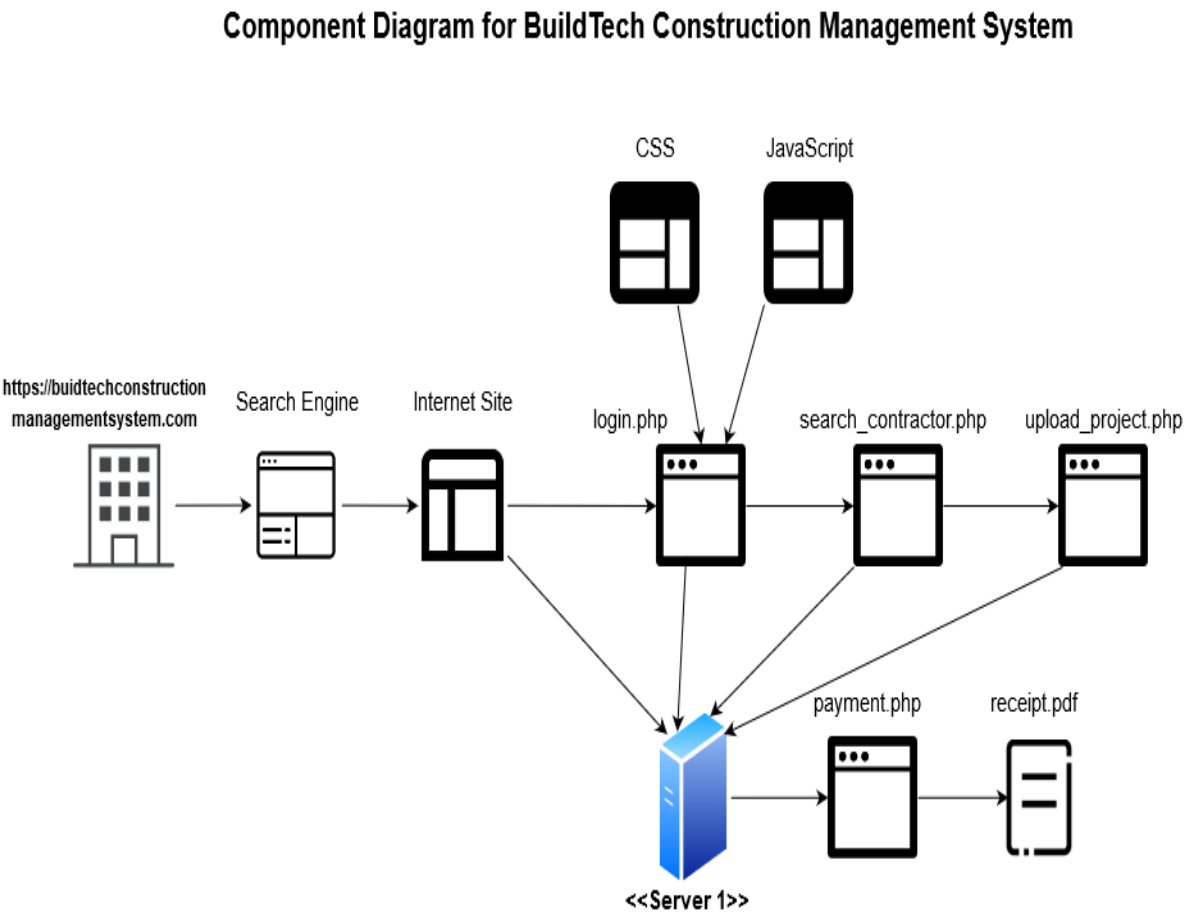


Fig. Component diagram for BuildTech Construction Management System

9. UML 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.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

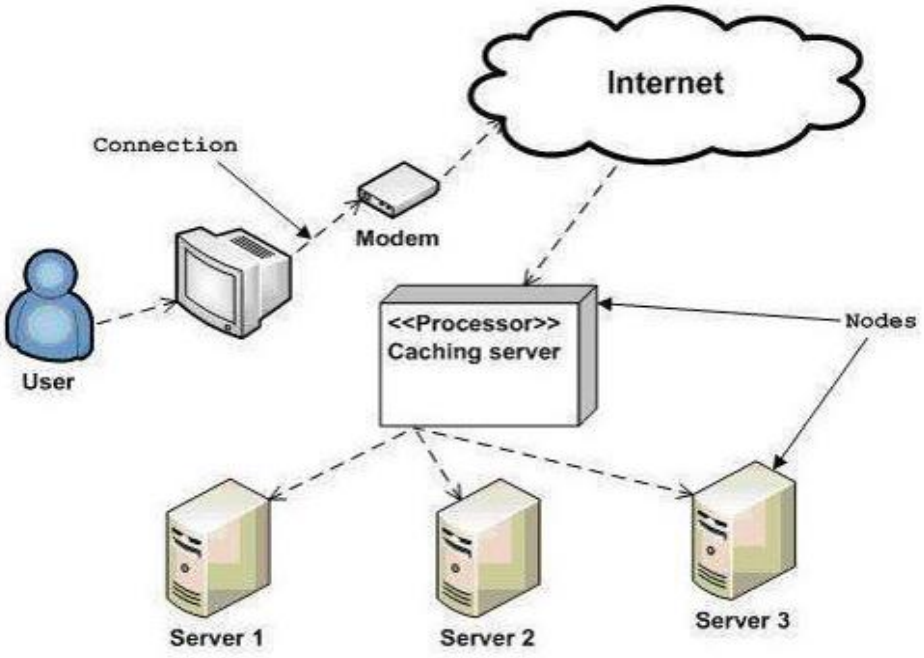


Fig. Deployment diagram for BuildTech Construction Management System

2.5 System Specification

2.5.1 Hardware Specification

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

2.5.2 Software Specification

Front End	-	HTML, CSS
Back End	-	PHP
Database	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, JQuery, PHP, CSS

2.6 Software Description

2.6.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP: Hypertext Preprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

2.6.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

❖ **MySQL is a database management system**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

❖ **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL92" refers to the standard released in 1992,"SQL: 1999" refers to the standard released in 1999, and "SQL: 2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

❖ **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

❖ **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

❖ **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

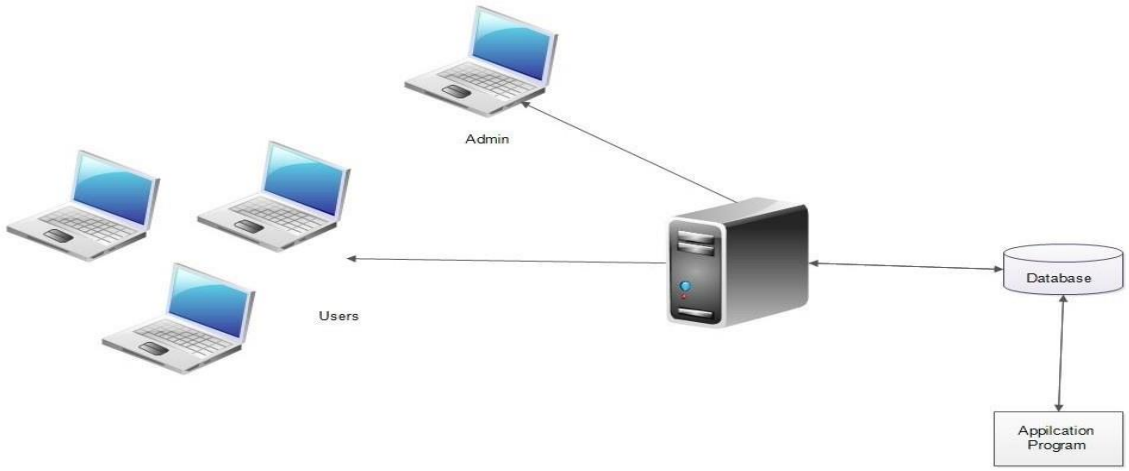
❖ **A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

2.7 System Design

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.

2.7.1 Architectural Design



The registered customer, contractor, labour and admin can accesses the website through internet using their Laptop, Smart Phone, Tablet or Desktop Computer. The System’s application program processes the user’s request and provides the required services by taking data from the system database.

2.7.2 Module Design

Admin Module

The administrator of the company is allowed to access all the services in the system. Admin has the overall control of the system. Admin can add or update labour categories, state and district details etc. Admin can view all the registered users, can able to activate or deactivate contractors and labours.

View/Manage registered user details	Deactivate/Activate the registered users.
Add/View/Update labour category, services, state and district	Manage profile/Change password

Registered Customer Module

After registration customer can search contractors nearby them and they can upload their project details to those contractors and do secure online payment. Customer can also add complaints to them.

Customer registration, login	Search contractors
Upload project to contractor	View/Approve estimation details
View/Approve weekly progress report	Add complaints to contractors

Registered Contractor Module

The contractor can register into this web site and they can see all the projects that are updated by the customers for construction. So that, the contractor can studies the plan details and requirements of the customer. Contractor can allocate duties to labours and can add wages to them. They can do construction management using this web site effectively.

Contractor registration, login	View/Approve/Reject project of customer
Prepare detailed estimation	Allocate duties to labours
View/Replay to Complaints from customer	Generate weekly progress report

Registered Labour Module

Labour can register into this system based on their labour category. After registration, he will get notification from contractor about worksite duties and get paid by the contractor.

Labour registration, login	View/Accept/Reject contractor request
Add daily attendance	Apply leave to contractor
View/Solve site complaints from contractor	View daily wages from contractor

2.7.3 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.

Every value in a relation is atomic, that is not decomposable.

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 are 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 attributes 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 other non-key attribute.

TABLES

Table No 01 : tbl_login

Primary Key : login_id

Field Name	Type	Size	Description
login_id	Int	10	Primary key of login table
username	Varchar	20	Unique username to login
password	Varchar	20	Password to login
user_type	Varchar	20	Type of users
status	Int	10	Status of the user

Table No 02 : tbl_state

Primary Key : state_id

Field Name	Type	Size	Description
state_id	Int	10	Primary key of state table
state_name	Varchar	20	To store state names
status	Int	10	Status of state

Table No 03 : tbl_district

Primary Key : district_id

Foreign Key : state_id

Field Name	Type	Size	Description
district_id	Int	10	Primary key of district table

state_id	Int	10	Foreign key with reference from state table
district_name	Varchar	30	To store district names
status	Int	10	Status of district

Table No 04 : tbl_ contractor_reg

Primary Key : contractor_id

Foreign Key : login_id

Field Name	Type	Size	Description
contractor_id	Int	10	Primary key of contractor table
contractor_name	Varchar	20	Name of contractor
email_id	Varchar	50	Email id of contractor
phone_no	Varchar	20	Phone number of contractor
licenceNo	Varchar	20	Licence number of contractor
licenceProof	Varchar	50	Licence proof of contractor
companyName	Varchar	50	Company name of contractor
dist_name	Varchar	10	District name of contractor
state_name	Varchar	20	State name of contractor
login_id	Int	10	Foreign key with reference from login table
status	Int	10	Status of the contractor

Table No 05 : tbl_labours_reg

Primary Key : lid

Foreign Key : login_id

Field Name	Type	Size	Description
lid	Int	10	Primary key of labour table
lab_name	Varchar	20	Name of labour
phoneno	Varchar	20	Phone number of labour
email_id	Varchar	20	Email id of labour
dist_name	Varchar	20	District name of labour
state_name	Varchar	20	State name of labour
adharCardNo	Number	20	Aadhar-card number of labour
login_id	Int	10	Foreign key with reference from login table
category_name	Varchar	20	Category name of labour
status	Int	10	Status of the labour

Table No 06 : tbl_customer_reg

Primary Key : cust_id

Foreign Key : login_id

Field Name	Type	Size	Description
cust_id	Int	10	Primary key of customer table
cust_name	Varchar	20	Name of customer
address	Varchar	50	Address of customer
postOffice	Varchar	20	Post office of customer
PIN Code	Number	20	Pin-Code of customer
phno	Number	20	Phone number of customer
email_id	Varchar	20	Email id of customer

dist_name	Varchar	20	District name of customer
state_name	Varchar	20	State name of customer
login_id	Int	10	Foreign key with reference from login table
status	Int	10	Status of the customer

Table No 07 : tbl_project

Primary Key : proj_id

Foreign Key : cust_id, contractor_id

Field Name	Type	Size	Description
proj_id	Int	10	Primary key of project table
yur_service	Varchar	20	Name of project
site_address	Varchar	50	Site address of project
project_plan	Varchar	50	Project plan details
package	Varchar	20	Different packages of project
no_of_floors	Int	10	No of floors
sqfeet	Int	100	Square feet details
cust_id	Int	10	Foreign key with reference from customer table
contractor_id	Int	10	Foreign key with reference from contractor table
status	Int	10	Status of the project

Table No 08 : tbl_weekly_progress_report

Primary Key : report_id

Foreign Key : proj_id

Field Name	Type	Size	Description
report_id	Int	10	Primary key of weekly report table
proj_id	Int	10	Foreign key with reference from project table
description	Varchar	50	Work description
activity_details	Varchar	20	Activities performed that week
fdate	Date	10	From date
tdate	Date	10	To date
status	Int	10	Status of the report

Table No 09 : tbl_labour_category

Primary Key : category_id

Field Name	Type	Size	Description
category_id	Int	10	Primary key of labour category table
category_name	Varchar	20	Category name of labour
status	Int	10	Status of the category

Table No 10 : tbl_complaint

Primary Key : comp_id

Foreign Key : login_id, to_login_id

Field Name	Type	Size	Description
comp_id	Int	10	Primary key of complaint table
login_id	Int	10	Foreign key with reference from customer table

to_login_id	Int	10	Foreign key with reference from contractor table
complaint	Varchar	100	Brief summary of the issue
status	Int	10	Status of the complaint

Table No 11 : tbl_daily_wages

Primary Key : wageid

Foreign Key : attnd_id

Field Name	Type	Size	Description
wageid	Int	10	Primary key of daily wages table
attnd_id	Int	10	Foreign key with reference from attendance table
wages	Int	100	Daily wage amount
status	Int	10	Status of the daily wages

Table No 12 : tbl_payment

Primary Key : pay_tid

Foreign Key : proj_id

Field Name	Type	Size	Description
pay_id	Int	10	Primary key of payment table
proj_id	Int	10	Foreign key with reference from project table
cdate	Timestamp	10	Current date of payment
amount	Number	20	Amount transferred
status	Int	10	Status of the payment

Table No 13 : tbl_postoffice

Primary Key : pid

Field Name	Type	Size	Description
pid	Int	10	Primary key of post office table
post_office	Varchar	20	Post office name
status	Int	10	Status of the post office

Table No 14 : tbl_attendance

Primary Key : attnd_id

Foreign Key : proj_id,lab_name

Field Name	Type	Size	Description
attnd_id	Int	10	Primary key of attendance table
proj_id	Int	10	Foreign key with reference from project table
cdate	Timestamp	10	Current date of attendance
lab_name	Varchar	20	Foreign key with reference from labour table
attnd	Varchar	20	Attendance of the labour
status	Int	10	Status of the attendance

Table No 15 : tbl_services

Primary Key : serv_id

Field Name	Type	Size	Description
serv_id	Int	10	Primary key of service table
service	Varchar	20	Name of service
status	Int	10	Status of the service

Table No 16 : tbl_site_location

Primary Key : sid

Foreign Key : proj_id, labour_name

Field Name	Type	Size	Description
sid	Int	10	Primary key of site location table
proj_id	Int	10	Foreign key with reference from project table
labour_name	Varchar	20	Foreign key with reference from labour table
fdate	Date	10	From date
tdate	Date	10	To date
status	Int	10	Status of the work completion

Table No 17 : tbl_leave

Primary Key : leave_id

Foreign Key : contractor_id, lid

Field Name	Type	Size	Description
leave_id	Int	10	Primary key of leave table
contractor_id	Int	10	Foreign key with reference from contractor table
lid	Varchar	20	Foreign key with reference from labour table
leave_date	Date	10	Date of leave
reason	Varchar	20	Reason for leave
status	Int	10	Status of the leave

Table No 18 : tbl_comp_assignlab

Primary Key : assign_id

Foreign Key : comp_id, proj_id,labour_name

Field Name	Type	Size	Description
assign_id	Int	10	Primary key of complaint assign labour table
comp_id	Int	10	Foreign key with reference from complaint table
proj_id	Int	10	Foreign key with reference from project table
labour_name	Varchar	100	Foreign key with reference from labour table
status	Int	10	Status of the complaint

Table No 19 : tbl_estimation

Primary Key : est_id

Foreign Key : proj_id

Field Name	Type	Size	Description
est_id	Int	10	Primary key of estimation table
proj_id	Int	10	Foreign key with reference from project table
total_cost	Varchar	100	Total estimated cost
concrete	Varchar	100	Cost of concrete work
brick	Varchar	100	Cost of brick
electrical	Varchar	100	Cost of electricals
status	Int	10	Status of the estimation

2.8 System Testing

2.8.1 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 terms 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 behavior 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 it 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 appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

2.8.2 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 is 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

2.8.2.1 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.

2.8.2.2 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. Moreover differences in program structures were removed and a unique program structure was evolved.

2.8.2.3 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 thatwill 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.

2.8.2.4 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.

2.8.3 Test Case

Test Case 1

Project Name: BuildTech Construction Management System					
Login Test Case					
Test Case ID: Fun_1			**Test Designed By:** SIJIMOL CYRIAC		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 09-04-2021		
Module Name :Login Screen			**Test Executed By :** Ms. MEERA ROSE MATHEW		
Test Title : Verify login with valid username and password			**Test Execution Date:** 12-04-2021		
Description: Test the Login Page					
Pre-Condition : User has valid user name and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to Login Page		Login Page should be	Login page displayed	Pass
2	Provide Valid User name	User Name: Admin	User should be able to Login	User Logged in and navigated to Admin Dashboard with records	Pass
3	Provide Valid Password	Password: Admin@2			
4	Click on Login button				
5	Provide Invalid User Name or password	User Name: Admin12 Password: admin*2	User should not be able to Login	Message for enter valid user or password displayed	Pass
6	Provide Null User Name or Password	User Name:null Password: null			
7	Click on Login button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Test Case 2

Project Name: BuildTech Construction Management System					
User Registration Test Case					
Test Case ID: Fun_2			**Test Designed By:** SIJIMOL CYRIAC		
Test Priority(Low/Medium/High): Medium			**Test Designed Date:** 09-04-2021		
Module Name :User Registration			**Test Executed By :**Ms. MEERA ROSE MATHEW		
Test Title : Verify new user registration			**Test Execution Date:** 12-04-2021		
Description: Test the User registration Page					
Pre-Condition : User should not be already registered					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to Registration Page		Registration Page should be displayed	Registration page displayed	Pass
2	Provide null information	Name: null	Validation message should be displayed	*Mandatory field message displayed	Pass
3	Provide Valid Details of user	All the mandatory registration details of the user	User should be able to register successfully	Registration successful and login page displayed	Pass
4	Click on Register button				
Post-Condition: User is validated with database and successfully registered in the application. The Account session details are logged in database					

2.9 Implementation

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.

2.9.1 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.

2.9.2 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 theirrole 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.

2.9.3 Operational Document

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.

2.9.4 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".

2.10 Conclusion and Future Enhancements

2.10.1 Future Enhancement

- Develop a Mobile Application for the website.
- Provide option to add contractor work gallery, so that customers will get better idea about the construction work and its quality.
- Include a chat-box option for the communication between customer and contractor.
- Include an option for labours to get paid online.
- Include an option for contractors to select and buy raw materials from vendors
- Provide more security.

2.10.2 Conclusion

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to upload projects by viewing profile of contractors. Provides lots of advantages like search contractor, view profile of contractors, enhanced user interface, payment options, add complaint, upload daily progress report options and may more.

2.11 Bibliography

Books/References:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- PankajJalote, "So*ftware engineering*: a precise approach", 2006.
- James lee and Brent ware Addison, "Open source web development with LAMP", 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

Websites:

- www.w3schools.com
- www.jquery.com
- https://www.lovelycoding.org/construction-work-management-system/
- https://www.slideshare.net/chiragbarasiya/construction-management-system-final-year-report
- https://app.diagrams.net
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

2.12 APPENDIX

2.12.1 SAMPLE CODE

Connection code

```
<?php
  $con=mysqli_connect("localhost","root","","constdb") or die("COULDN'T CONNECT");
?>
```

Admin activate contractor code

```
        <?php
session_start();
include("DbConne.php");
if(!empty($_SESSION['uname']))
{
    $temp=$_SESSION['uname'];
    if(isset($_REQUEST['x']))
    {
                $a=intval($_GET['x']);
                $sql="update tbl_contractor_reg set status='1' where login_id='$a'";
                $sqli="update tbl_login set status='1' where login_id='$a'";
                mysqli_query($con,$sql);
                mysqli_query($con,$sqli);
    }
    if(isset($_REQUEST['y']))
    {
                $a=intval($_GET['y']);
                $sql="update tbl_contractor_reg set status='0' where login_id='$a'";
                $sqli="update tbl_login set status='0' where login_id='$a'";
                mysqli_query($con,$sql);
                mysqli_query($con,$sqli);
    }
    ?>
<!DOCTYPE html>
<html lang="en">
```

```
<?php include("header.php");
?>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>Admin</title>
    <link href="css/styles.css" rel="stylesheet" />
    <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/js/all.min.js"
crossorigin="anonymous"></script>
                <style>
                    table, th, td {
                                                                    text-
align:center;
                                                                    min-
width: 150px;
                    }
    </style>
  </head>
  <body>
    <div id="layoutSidenav">
      <div id="layoutSidenav_nav">
        <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
          <div class="sb-sidenav-menu">
            <div class="nav">
              <div class="sb-sidenav-menu-heading"></div>
              <a class="nav-link" href="index.php">
                <div class="sb-nav-link-icon"><i class="fas fa-fw fa-home"></i></div>
                Dashboard
              </a>
              <div class="sb-sidenav-menu-heading">Activities</div>
              <a class="nav-link collapsed" href="#" data-toggle="collapse" data-
target="#collapseLayouts"

                  aria-expanded="false" aria-controls="collapseLayouts">
                <div class="sb-nav-link-icon"><i class="fas fa-fw fa-users"></i></div>
                Users
```

```
<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
                    </a>
                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne" data-parent="#sidenavAccordion">
                      <nav class="sb-sidenav-menu-nested nav">
                        <a class="nav-link" href="viewcust.php">Manage Customer</a>
                         <a class="nav-link" href="viewcontra.php">Manage Contractor</a>

                                                                    <a class="nav-link" href="viewlab.php">Manage Labour</a>
                      </nav>
                    </div>

                                    <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseLayouts"

                                    aria-expanded="false" aria-controls="collapseLayouts">
                      <div class="sb-nav-link-icon"><i class="fas fa-list-alt"></i></div>
                      Category
                      <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
                    </a>
                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingTwo" data-parent="#sidenavAccordion">
                      <nav class="sb-sidenav-menu-nested nav">
                        <a class="nav-link" href="viewcat.php">Manage Category</a>
                      </nav>
                    </div>

                                    <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseLayouts"

                                    aria-expanded="false" aria-controls="collapseLayouts">

                                                                    <div class="sb-nav-link-icon"><i class=" fas fa-tasks"></i></div>

                                                                Services

                                                                    <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>

                                                                </a>
```

```
                                                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingTwo" data-parent="#sidenavAccordion">
                                                        <nav class="sb-sidenav-menu-nested nav">
                                                            <a class="nav-link" href="viewservice.php">Manage Services</a>
                                                        </nav>
                                                    </div>
                                                    <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapsePages"
                                                        aria-expanded="false" aria-controls="collapsePages">
                                                        <div class="sb-nav-link-icon"><i class=" fa fa-map-marker"></i></div>
                                                        Locations
                                                        <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
                                                    </a>
                                                    <div class="collapse" id="collapsePages" aria-labelledby="headingThree" data-parent="#sidenavAccordion">
                                                        <nav class="sb-sidenav-menu-nested nav accordion" id="sidenavAccordionPages">
                                                            <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#pagesCollapseAuth"
                                                                aria-expanded="false" aria-controls="pagesCollapseAuth">
                                                                State
                                                                <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
```

```
</a>
                                                                    <div
class="collapse" id="pagesCollapseAuth" aria-labelledby="headingOne"
                                                                    data-
parent="#sidenavAccordionPages">
    <nav class="sb-sidenav-menu-nested nav">
                        <a class="nav-link" href="viewstate.php">Manage State</a>
    </nav>
                                                                    </div>
                                                                    <a  class="nav-link
collapsed" href="#" data-toggle="collapse" data-target="#pagesCollapseError"
                                                                    aria-
expanded="false" aria-controls="pagesCollapseError">
    District
    <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
                                                                    </a>
                                                                    <div
class="collapse" id="pagesCollapseError" aria-labelledby="headingOne"
                                                                    data-
parent="#sidenavAccordionPages">
    <nav class="sb-sidenav-menu-nested nav">
                        <a class="nav-link" href="viewdist.php">Manage District</a>
```

```
</nav>
                                                                    </div>
                                                                    <a class="nav-link
collapsed" href="#" data-toggle="collapse"
                                                                    data-
target="#pagesCollapseError" aria-expanded="false" aria-controls="pagesCollapseError">
                                                                        Post
Office
    <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
                                                                    </a>
                                                                    <div
class="collapse" id="pagesCollapseError" aria-labelledby="headingOne"
                                                                    data-
parent="#sidenavAccordionPages">
    <nav class="sb-sidenav-menu-nested nav">
                        <a class="nav-link" href="viewpost.php">Manage Post Office</a>
                                                                    </nav>
                                                                    </div>
                                                                    </nav>
                                                                    </div>
            </div>
            </div>
            <div class="sb-sidenav-footer">
               <div class="small">Logged in as:</div>
               BuildTech Construction
            </div>
```

```
</nav>
        </div>
        <div id="layoutSidenav_content">
          <main>
            <div class="container-fluid">
              <h2 class="mt-4">Contractor</h2>
              <ol class="breadcrumb mb-4">
                <li class="breadcrumb-item"><a href="index.php">Dashboard</a></li>
                <li class="breadcrumb-item active">Contractor</li>
              </ol>
              <div class="card mb-4">
                <div class="card-body">

                                                <div class="table-responsive">

                 <form action="Addreg2.php" method="POST">
                        <table      style="background-color:#ccd4e0"      class="table      table-bordered"
id="dataTable"

                                                width="100%" cellspacing="0">
                  <?php
                 $query = "select * from tbl_contractor_reg";
                 $results = mysqli_query($con,$query);
                        echo "<h2><center>Contractor Details</center></h2>";
                 echo  "<tr><th>Sl  No</th><th>Contractor  Name</th><th>Email  Id</th><th>Phone
No</th><th>License No</th>

                                                <th>License                    Proof</th><th>Company
Name</th><th>Specialization</th><th>District Name</th>

                                                <th>State Name</th><th>Status</th></tr>";
                 while($fin=mysqli_fetch_array($results))
                 {
                 echo "<tr>";
                 echo "<td>".$fin['contractor_id']."</td><td>"

                                                .$fin['contractor_name']."</td><td>"
                                .$fin['email_id']."</td><td>"
                        .$fin['phone_no']."</td><td>"
                       .$fin['licenseNo']."</td><td>";
```

```
                                                                    echo "<a
href='viewproof.php?x=" .$fin['login_id']." ' target='_blank'>proof details</a></td><td>"
                    .$fin['companyName']."</td><td>"

                                                        .$fin['spec']."</td><td>"

                                                          .$fin['dist_name']."</td><td>"
                .$fin['state_name']."</td><td>";
                    if($fin['status'] == 0 || $fin['status'] ==")
                    {
                            echo "<a href='viewcontra.php?x=" .$fin['login_id']."
'>deactivated</a>";
                            }
                            else
                                    {
                                        echo "<a href='viewcontra.php?y="
.$fin['login_id']." '>activated</a>
                                        </td>";
                                    }
                  echo "</tr>";
                    }
                     ?>
                  </table>
                  </form>
                    </div>
                </div>
                <div style="height: 100vh;"></div>
                <div class="card mb-4"><div class="card-body"></div></div>
              </div>
            </main>
            <footer class="py-4 bg-light mt-auto">
              <div class="container-fluid">
                <div class="d-flex align-items-center justify-content-between small">
                  <div class="text-muted"></div>
                  <div>
                    <a href="#"></a>
                    &middot;
                    <a href="#"></a>
```

```
            </div>
                    </div>
                  </div>
                </footer>
            </div>
        </div>
        <script src="https://code.jquery.com/jquery-3.5.1.min.js" crossorigin="anonymous"></script>
        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
        <script src="js/scripts.js"></script>
    </body>
</html>
<?php
}
else
{
     header("location: ../../login.php");
}
?>
```

Project_Payment Code

```
<?php
session_start();
include("DbConne.php");
if(isset($_SESSION['uname']))

{
$temp=$_SESSION['uname'];
if(isset($_GET['proj_id']))
     {
               $u=$_GET['proj_id'];
                        $_SESSION['proj_id']=$u;
      $y=$_GET['con_name'];
               $_SESSION['con_name']=$y;
     }
     ?>
```

```
<!DOCTYPE html>
<html lang="en">
<?php include("header.php");
?>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
    <meta name="description" content="" />
    <meta name="author" content="" />
                    <title>Customer</title>
                    <link href="css/styles.css" rel="stylesheet" />

                    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
                    <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/js/all.min.js" crossorigin="anonymous"></script>
            </head>
<body>
    <div id="layoutSidenav">
            <div id="layoutSidenav_nav"<nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">

    <div class="sb-sidenav-menu">
                    <div class="nav">
                            <div class="sb-sidenav-menu-heading">Core</div>
    <a class="nav-link" href="index.php">

                                    <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>Dashboard
                                    </a>

                            <div class="sb-sidenav-menu-heading">Activities</div>
                    <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">

                                    <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>Project
```

```
<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>

                                    </a>

                                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne" data-parent="#sidenavAccordion">
<nav class="sb-sidenav-menu-nested nav">
                <a class="nav-link" href="viewproj.php">View Project</a>
                    <a class="nav-link" href="viewest.php">View Estimation</a>
                    </nav> </div>
                <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">

                                                <div class="sb-nav-link-icon"><i class="fas fa-chart-bar"></i></div>

                                                Weekly Progress Report

                                                <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>

                                    </a>

                                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne" data-parent="#sidenavAccordion">

                                                <nav class="sb-sidenav-menu-nested nav">

                                                                <a class="nav-link" href="viewreport.php">View Report</a>


                                                </nav>

                                    </div>


                                    <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
```

```
<div class="sb-nav-link-icon"><i class="fa fa-credit-card"></i></div>
                                                    Payment
                                                    <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
                                    </a>
                                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne" data-parent="#sidenavAccordion">
                                                    <nav class="sb-sidenav-menu-nested nav">
                                                                    <a class="nav-link" href="addpay.php">View Payment</a>
                                                                    <a class="nav-link" href="viewtran.php">View Transaction Log</a>
                                                    </nav>
                                    </div>

                                    <a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
                                                    <div class="sb-nav-link-icon"><i class="fas fa-comments"></i></div>
                                                    Complaints
                                                    <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
                                    </a>
                                    <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne" data-parent="#sidenavAccordion">
```

```
<nav class="sb-sidenav-menu-nested nav">
                                                    <a class="nav-link" href="viewcomp.php">View Complaints</a>
                                                </nav>
                                            </div>
                                        </nav>
                                    </div>
        <div id="layoutSidenav_content">
            <main>
                <div class="container-fluid">
                    <h2 class="mt-4">Payment Details</h2>
                    <ol class="breadcrumb mb-4">
                        <li class="breadcrumb-item"><a href="addpay.php">Dashboard</a></li>
                        <li class="breadcrumb-item active">Customer</li>
                    </ol>
                            <div class="card mb-4">
                                    <div class="card-body">
                                <form method="POST" action="#">
                                <div class="form-group">
                                <div class="form-label-group">
                                <label for="exampleInputEmail1">Contractor Name:</label>
                                <input type="text" class="form-control" id="name1" name="name" disabled value="<?php echo $_SESSION['con_name']; ?>"
                                placeholder="Contractor Name" autofocus="autofocus" required>
```

```
<br><label for="exampleInputEmail1">Amount:</label>

                                        <input type="number" class="form-control" id="amt1" name="amt"

                                        placeholder="Enter the amount" autofocus="autofocus" onblur="validate2()" required>



                                                                    </div>


                                                                    </div>



      <div class="modal-footer">
       <input type="submit" name="submit" value="Pay with Razorpay" class="btn btn-success">                                                                    </div>
                                        </form>
                                                                                </div></div>
  <div style="height: 100vh;"></div>
  <div class="card mb-4"><div class="card-body"></div></div>
</div>
</main>
<footer class="py-4 bg-light mt-auto">
<div class="container-fluid">
  <div class="d-flex align-items-center justify-content-between small">
     <div class="text-muted"></div>
     <div>
        <a href="#"></a>
        &middot;
        <a href="#"></a>
     </div>
  </div>
</div>
</footer>
</div>
</div>
```

```
<?php
session_start();
include("DbConne.php");

if(isset($_POST['submit'])){
$temp=$_SESSION['uname'];
$siji=$_SESSION['proj_id'];
$amount=$_POST['amt'];
$_SESSION['amt']=$amount;
$amt=$_SESSION['amt'];
$query = "select * from tbl_login l,tbl_customer_reg h  where l.username='$temp' and l.login_id=h.login_id";
$results = mysqli_query($con,$query);
$x=mysqli_fetch_array($results);
$e=$x['cust_name'];
$f=$x['phno'];


$g=$x['email_id'];
}
    ?>
<button id="rzp-button1">Pay</button>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<style>
#rzp-button1{
  display: none;
}
</style>
<script>
$(document).ready(function(){
  $("#rzp-button1").click()
  });
</script>
<script src="https://checkout.razorpay.com/v1/checkout.js"></script>
<script>
```

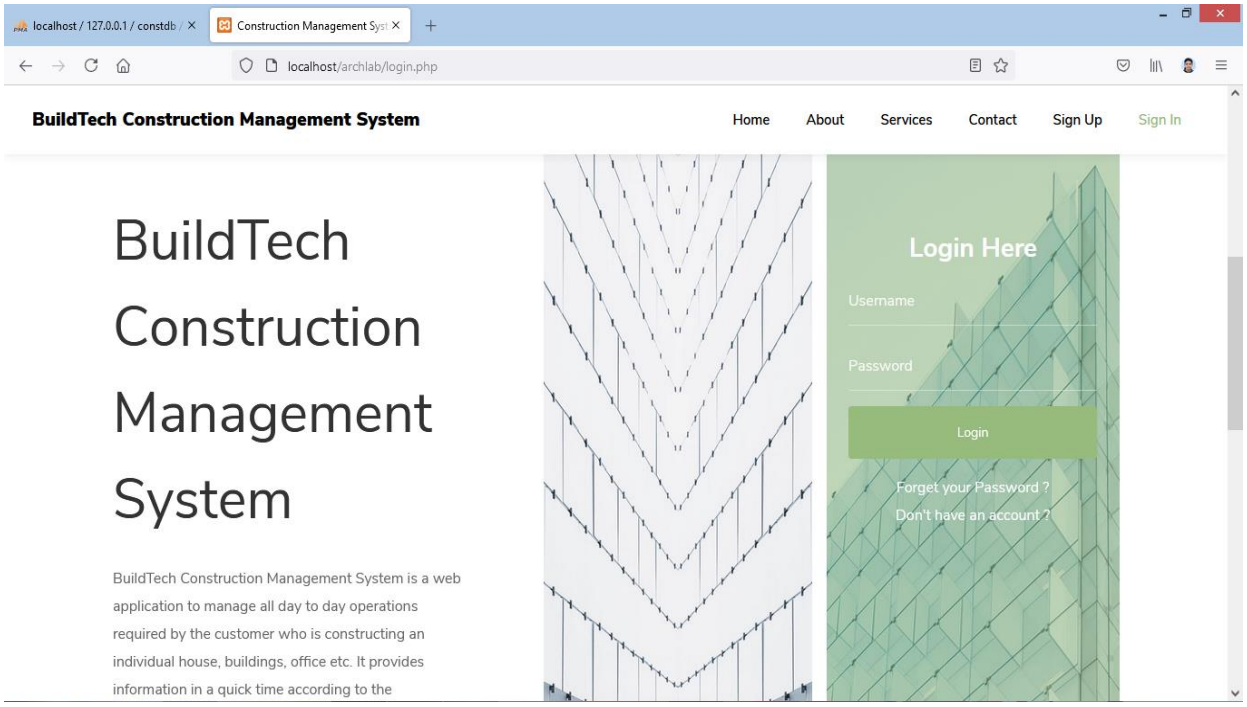
```
var options = {
    "key": "rzp_test_aG8mdESPQXjZBJ", // Enter the Key ID generated from the Dashboard
    "amount": "<?php  echo $amt*100;?> ", // Amount is in currency subunits. Default currency is INR. Hence, 50000 refers to 50000 paise
    "currency": "INR",



  "name": "<?php  echo $e;?>",
    "description": "Project Transaction",
    "image": "https://img-premium.flaticon.com/png/512/21/21104.png?token=exp=1622901614~hmac=4d8e22f1c6d8531f478dfaa653360e59",


    //"order_id": "order_9A33XWu170gUtm", //This is a sample Order ID. Pass the `id` obtained in the response of Step 1
    "handler": function (response){
        //alert(response.razorpay_payment_id);
        if (typeof response.razorpay_payment_id == 'undefined' ||  response.razorpay_payment_id < 1) {


} else {
  redirect_url = 'success.php';
}
location.href = redirect_url;
        //alert(response.razorpay_order_id);
        //alert(response.razorpay_signature)
    },
    "prefill": {
        "name": "<?php  echo $e;?>",
        "email": "<?php  echo $g;?>",
        "contact": "<?php  echo $f;?>"
    },
    "notes": {
        "address": "Razorpay Corporate Office"
    },
    "theme": {
        "color": "#3399cc"
    }
};
```

```
var rzp1 = new Razorpay(options);
rzp1.on('payment.failed', function (response){
      (response.error.code);
      (response.error.description);
        (response.error.source);
        (response.error.step);
         (response.error.reason);
        (response.error.metadata.order_id);
        (response.error.metadata.payment_id);
        location.href='failed.php';
});
document.getElementById('rzp-button1').onclick = function(e){
    rzp1.open();
    e.preventDefault();
}
</script>
<script>
function validate2()
{
var phone = document.getElementById("amt1").value;
var ph=/^\d+/;
 if(!phone.match(ph))
{
alert("Please Only Enter Numeric Characters As Amount! (Allowed  input:0-9)");
document.getElementById("amt1").value="";
}
}
</script>
<script src="https://code.jquery.com/jquery-3.5.1.min.js" crossorigin="anonymous"></script>
<script        src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
<script src="js/scripts.js"></script>
</body>
<script src="app.js"></script>
<script>
       history.pushState(null, null, location.href);
      window.onpopstate = function () {
                                history.go(1);
      };
```

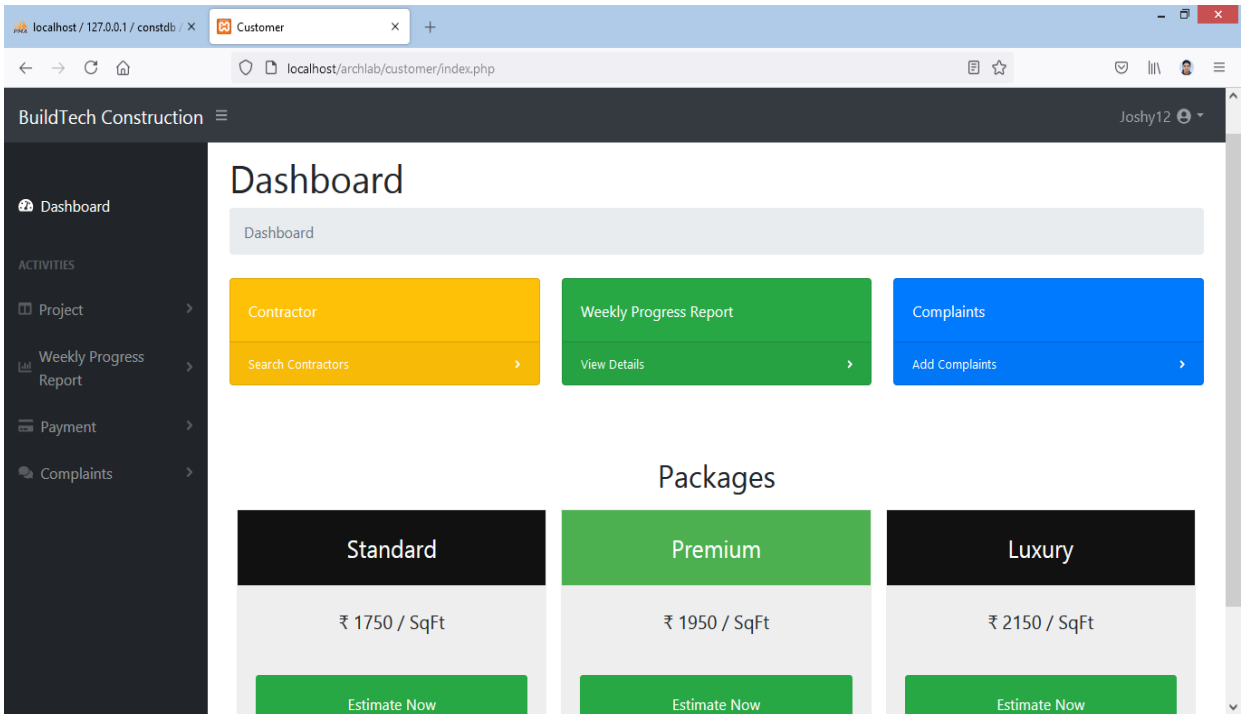
```
    </script>
</html>
<?php
}
else
{
    header("location: ../login.php");
}
?>
```

2.10.2 SCREENSHOTS

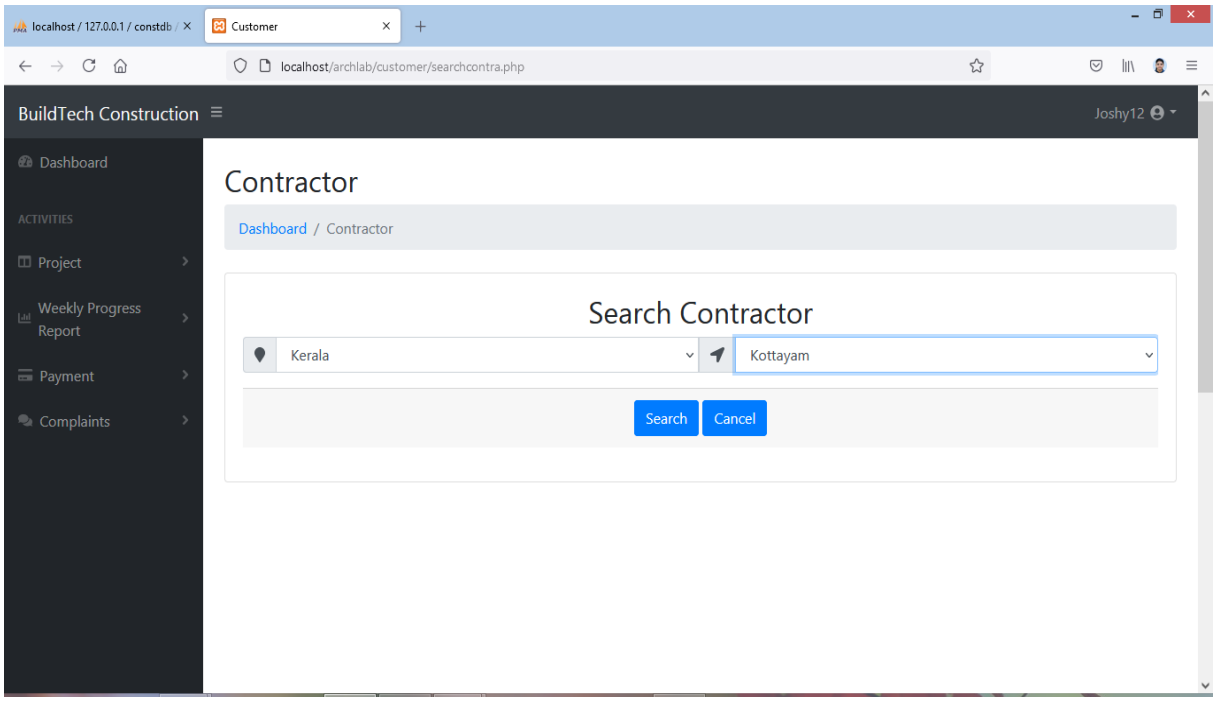
❖ Login Page



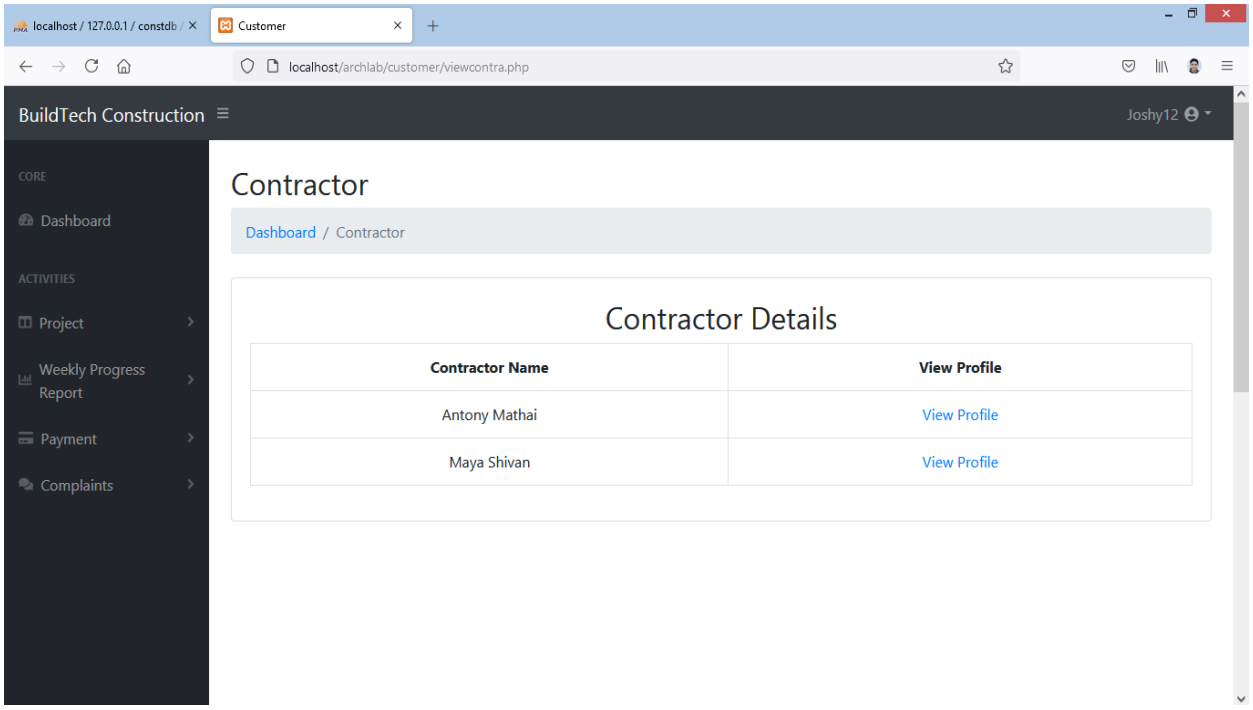
❖ Customer Home Page



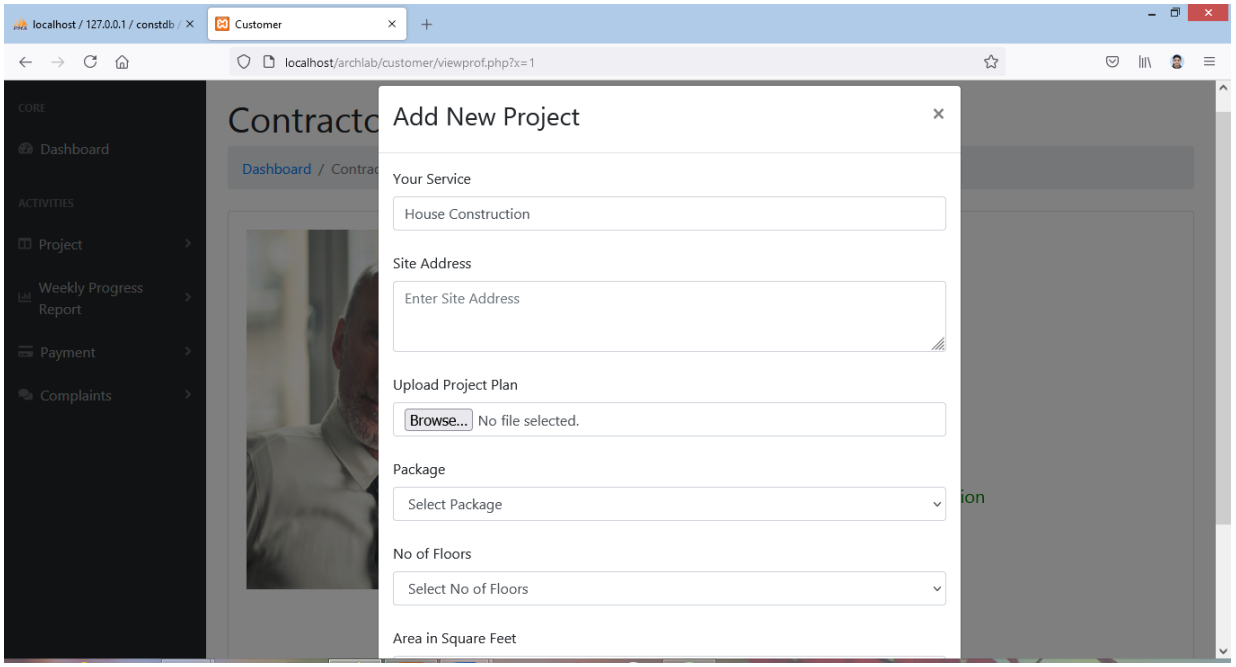
❖ **Search contractor page**



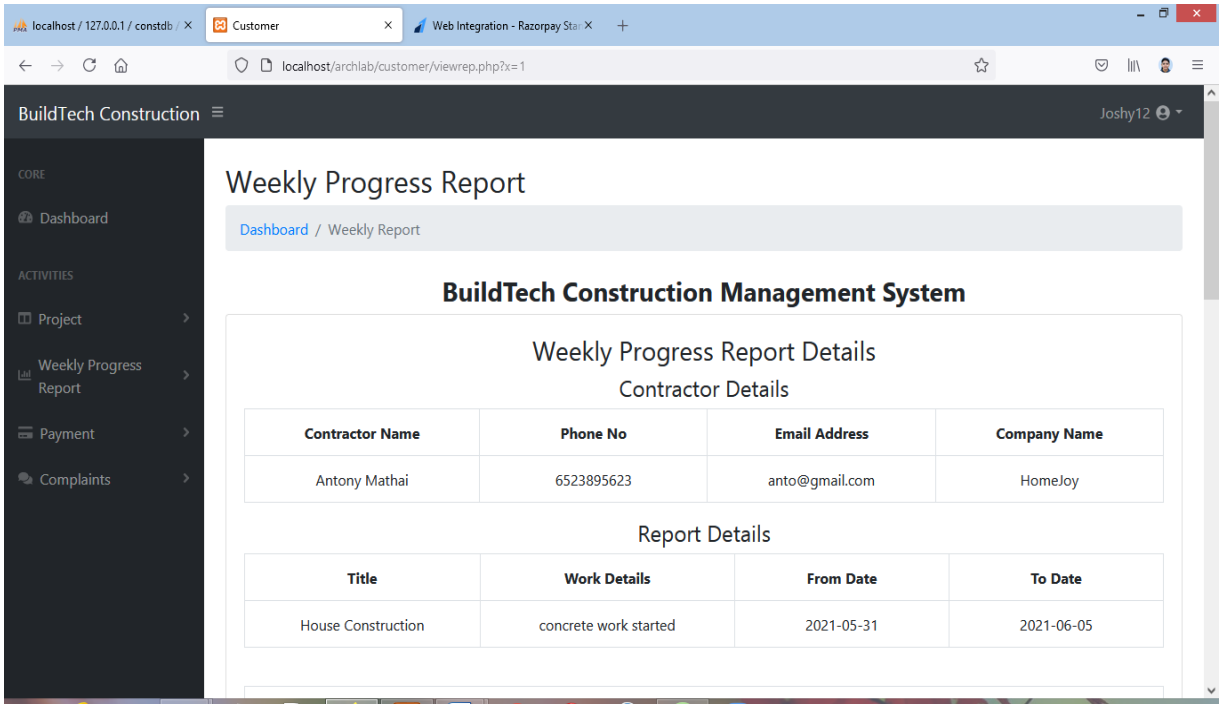
❖ **View contractor page**



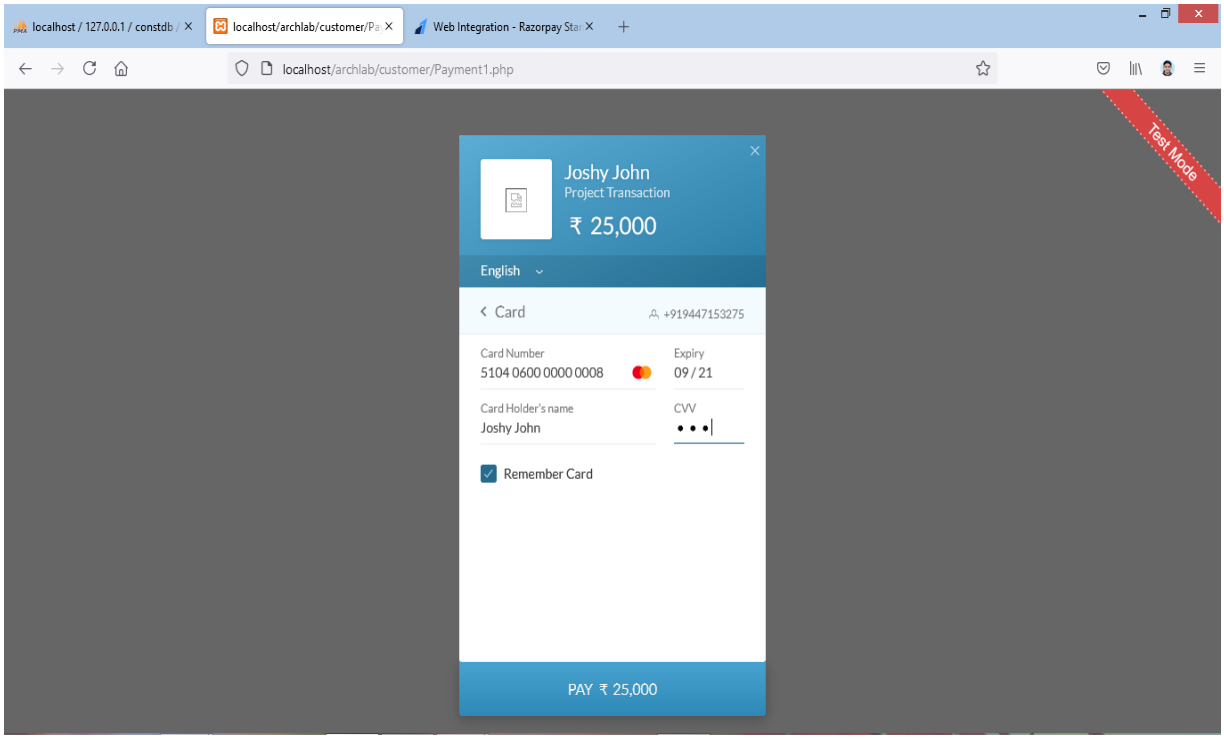
❖ **Add new project page**



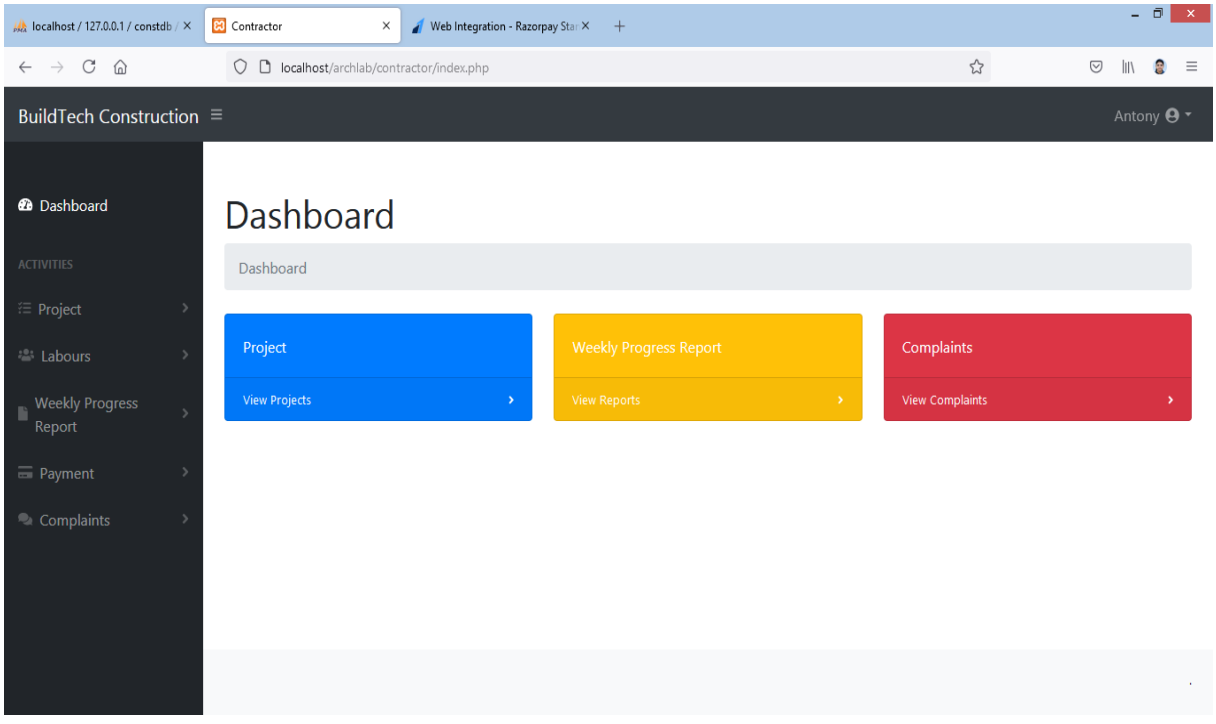
❖ **View weekly progress report page**



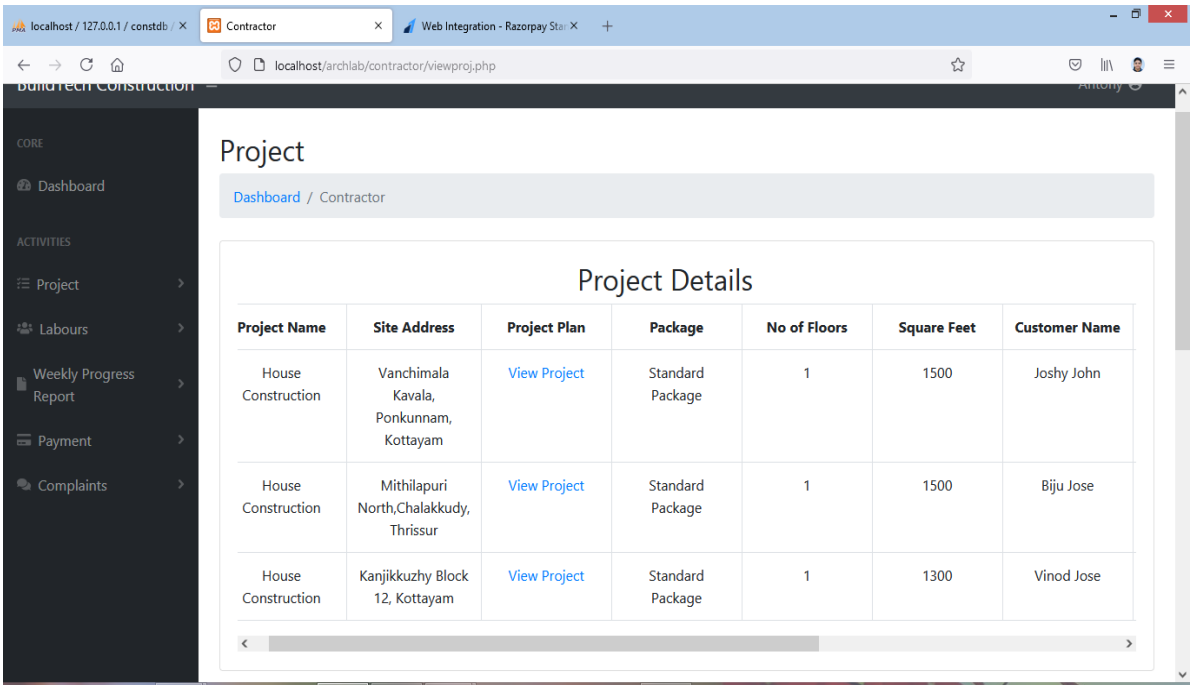
❖ **Payment via Razorpay gateway**



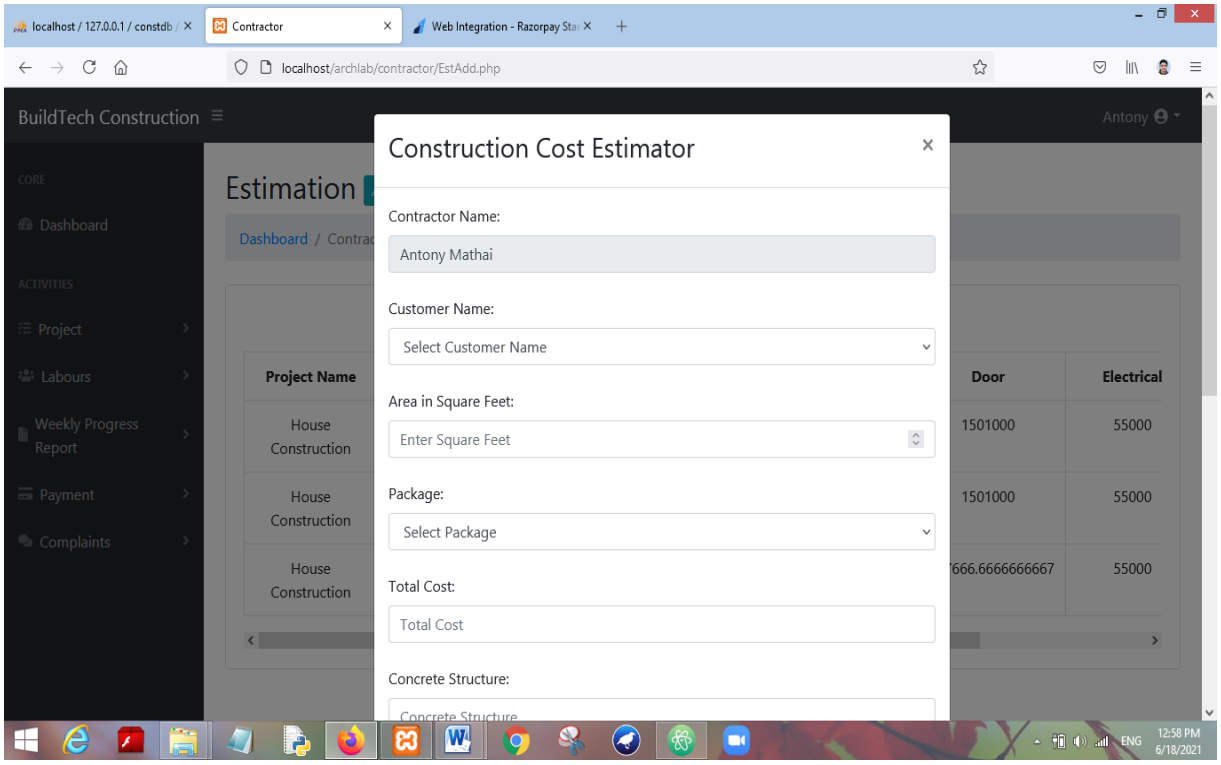
❖ **Contractor Dashboard Page**



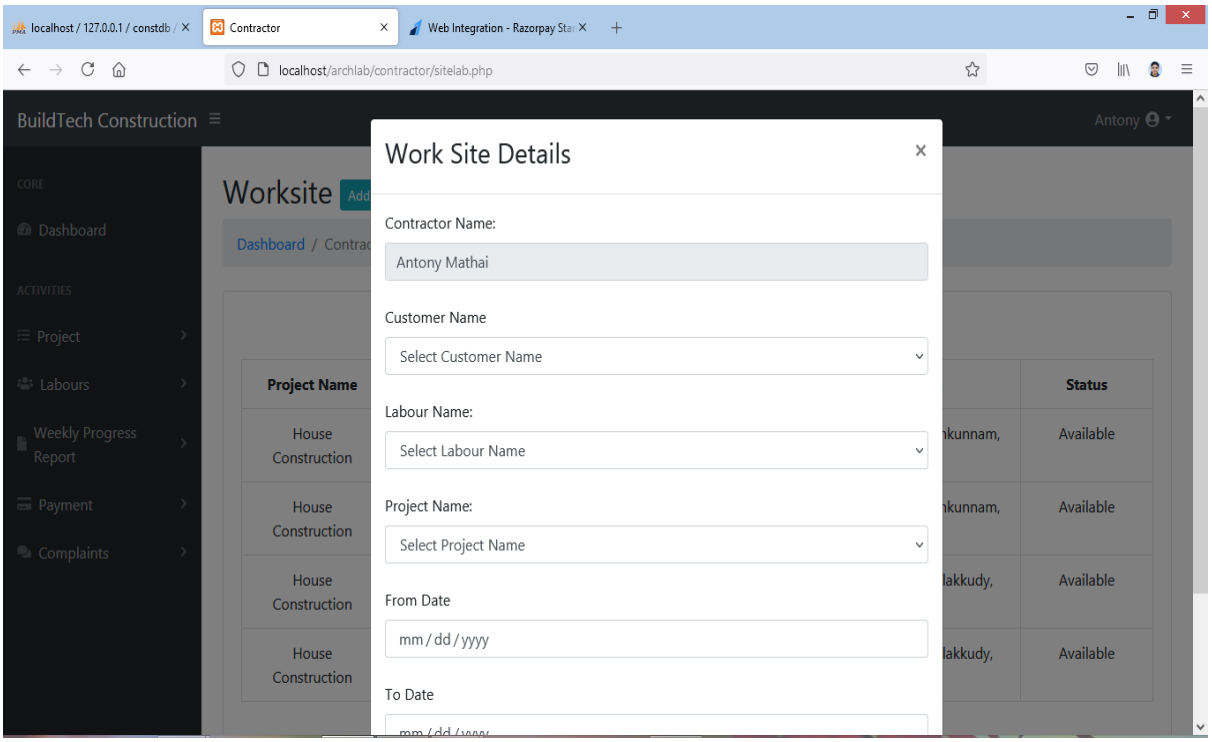
❖ **View uploaded project details page**



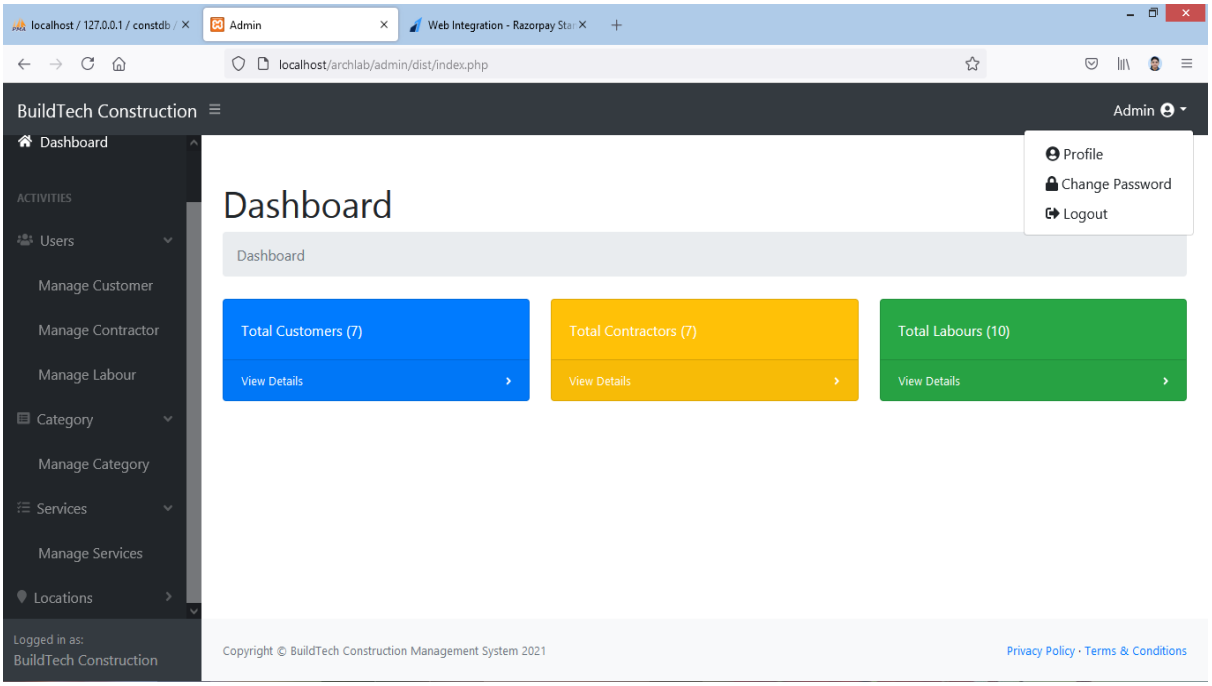
❖ **Construction cost estimator page**



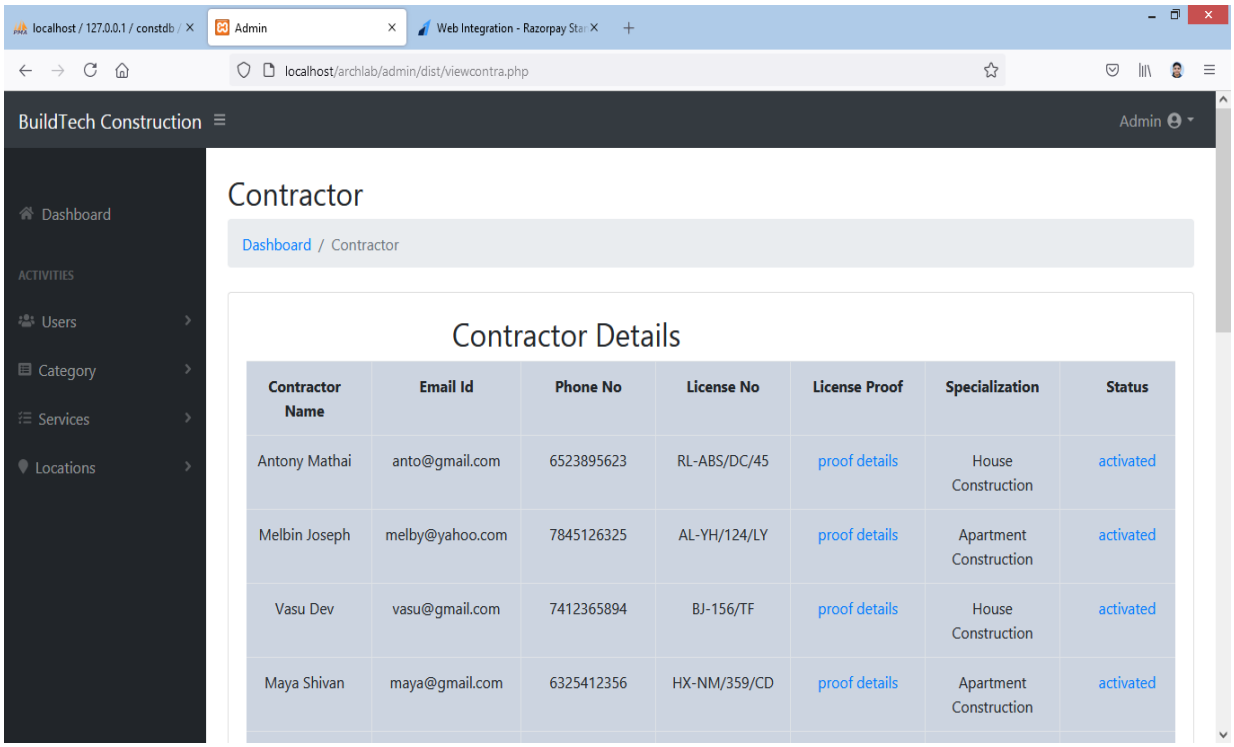
❖ **Assign labours to worksite page**



❖ **Admin Dashboard Page**



❖ **Registered contractors's page**



❖ **View/Add/Update labour category page**