

TEAMS COLLABORATOR

A PROJECT REPORT

Submitted in partial fulfillment for the award of the degree

of

BACHELOR OF COMPUTER APPLICATION



MAHARAJA SURAJMAL INSTITUTE

**C-4, JANAK PURI,
NEW DELHI – 110058**

MAY 2022

SUBMITTED BY:

HARSHIT SHARMA (03621202019)

SURBHI PANDEY (00121202019)

BCA – 6TH SEM (2ND SHIFT)

SUBMITTED TO:

MS. TARUNIM SHARMA

SUPERVISOR

(ASSISTANT PROFESSOR)

MAHARAJA SURAJMAL INSTITUTE

BONAFIDE CERTIFICATE

Certified that this project report “ **TEAMS COLLABORATOR** ” is the bonafide work of “ **HARSHIT SHARMA and SURBHI PANDEY** ” who carried out the project work under my supervision.

SIGNATURE

Ms. Tarunim Sharma

SUPERVISOR

Assistant Professor

Dept. Computer Applications

Maharaja Surajmal Institute

C-4, Janakpuri,

New Delhi - 110058

ACKNOWLEDGEMENT

This is to acknowledge that the project entitled “ Teams Collaborator ” is the result of the cooperation and guidance of prominent persons in the subject line.

We would like to express our gratitude towards our faculty Ms. Tarunim Sharma for her significant support. We are sincerely thankful to the honorable Ma'am for providing us with the necessary facilities to carry out the project successfully. This is only because of her valuable guidance that we are capable of submitting this project with all its requirements duly fulfilled.

ABSTRACT

Before pandemic collaboration tools and applications had a limited reach. There were only few people, or say organizations and giant tech companies who knew their importance in workplace. But after, pandemic these tools became a common application for every person whose work was linked to IT field. Collaboration application are changing the way people work. A collaboration app is any piece of software that helps people get work done together.

Hence, keeping this feature in mind we created a user-friendly web-based collaboration application 'Teams Collaborator' which will provide features more than a basic collaborator application. Since, there were applications which only provided one specific type of functionality and user had to navigate among these for getting his work done. Therefore, we brought down all the features, for which a person has to use different tools, under one roof, i.e., included all the features in one application.

The application will also let people know about team activity on work that pertains to them. Through the application, user will be having access to the information and assets they need, so they can pick up their piece of a project at the right time without someone else having to email them to do so. The aim of this application is to reduce the complexity of user's workplace and give them an easy to interact web application.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF FIGURES	vii
	LIST OF TABLES	ix
1.	INTRODUCTION	10
	1.1 Overview on Collaboration Tools	11
	1.1.1. Classification based on dimensions	11
	1.1.2. Online Collaboration Tools	11
	1.2 What is Teams Collaborator?	12
	1.3 Why Teams Collaborator?	12
	1.4 Issues with other collaboration web applications	13
	1.4.1 How is Teams Collaborator Different?	14
2.	LITERATURE REVIEW	15
3.	WEB APPLICATION SPECIFICATION	18
	3.1 Overview	19
	3.2 Feasibility study of the project	21
	3.3 Software Requirements Specification (SRS) of the project	22
	3.3.1 Hardware Requirements	22
	3.3.2 Software Requirements	22
	3.3.3 Technologies Required	24
	3.4 SDLC Model used for the project	25
4.	SOFTWARE DESIGN AND ANALYSIS	27
	4.1 Overview	28
	4.2 Data Flow Diagrams (DFD)	28
	4.2.1 DFD Components	29
	4.2.2 Levels of DFD	29
	4.3 Input and Output Design	31
	4.3.1 Input Design	31
	4.3.2 Output Design	32

	4.4 Database Design	32
	4.5 Entity Relationship Diagram	35
	4.5.1 Components of ER diagrams	36
5.	PROJECT FEATURES AND FUNCTIONALITY	39
	5.1 High-level Languages used in project	40
	5.2 Various Features of Teams Collaborator	44
	5.2.1 ‘Home Page’ of Teams Collaborator	44
	5.2.2 ‘Notifications’ Feature	47
	5.2.3 ‘Teams’ Feature	47
	5.2.4 ‘Project Management’ Feature	49
	5.2.5 ‘Assignment Allocation’ Feature	51
	5.2.6 ‘Audio and Video Call’ Feature	53
	5.2.7 ‘Messaging’ Feature	54
	5.2.8 ‘File Sharing’ Feature	56
	5.2.9 ‘User Authentication and Services’ Feature	57
6.	SECURITY AND TESTING OF PROJECT	61
	6.1 Security of project	62
	6.2 Testing of project	66
	6.2.1 Manual Testing	66
	6.2.1.1 Stages of Manual Testing	66
	6.2.2 Other Types of Testing	68
	CONCLUSIONS	69
	REFERENCES	70

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	Steps involved in SDLC	20
3.4.1	Waterfall Model	26
4.2.2.1	Level 0 DFD	29
4.2.2.2	Level 1 DFD	30
4.2.2.3	Level 2 DFD	31
4.5.1.1	ER Diagram of project	38
5.2.1.1	Image showing Home page of teams collaboartor	45
5.2.1.2	Image showing payment being done through payment gateway	45
5.2.2.1	Image representing Notifications feature	47
5.2.3.1	Image representing Teams page	48
5.2.4.1	Image showing projects page with a list of various projects	50
5.2.4.2	Image showing task board of a particular project with various tasks in task detail section	50
5.2.5.1	Image showing assigned category in Assignments page	52
5.2.5.2	Image showing assignments details page	52
5.2.5.3	Image showing completed category in Assignments page	52
5.2.6.1	Image showing contact list in call page	53
5.2.6.2	Image showing ongoing video call	54
5.2.7.1	Image showing chat page having load message page	55
5.2.7.2	Image showing chat messages with a contact on chat page	55
5.2.8.1	Image showing shared document page containing list of documents	57
5.2.9.1	Image showing sign in page	58
5.2.9.2	Image showing password assistance page	59
5.2.9.3	Image showing sign up page	60
6.1.1	Image showing HTTPS security	78

6.1.2	Image showing SSL certificate logo	79
-------	------------------------------------	----

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
4.4.1	Assignment table description	33
4.4.2	File table description	33
4.4.3	Project details table description	34
4.4.4	Project table description	34
4.4.5	Team table description	34
4.4.6	User table description	35
4.4.7	User extra details table description	35

CHAPTER – 1

INTRODUCTION

1.1 OVERVIEW ON COLLABORATION TOOLS

A collaboration tool helps people to collaborate. The purpose of a collaboration tool is to support a group of two or more individuals to accomplish a common goal or objective. Collaboration tools can be either of a non-technological nature such as paper, flipcharts, post-it notes or whiteboards. They can also include software tools and applications such as collaborative software.

Three aspects of collaboration: communication, coordination and cooperation can be used to categorize collaboration tools.

- Communication tools provide exchange of information between individuals: email, voicemail, instant messaging(IM), video call or VoIP(voice over IP)
- Coordination is defined as "the deliberate and orderly alignment or adjustment of partners' actions to achieve jointly determined goals". Collaboration tools supporting this are the ones who allow a person to set up group activities, schedules and deliverables: online calendars, time trackers, spreadsheet
- Cooperation tools allow groups to have real-time discussions and to shape an idea or thought together. Trends in terms of collaboration target on helping to maintain the "main idea" within big organizations and make connections visible: video conferencing, IM teleconferencing

1.1.1 Classification based on dimensions

- i. Asynchronous collaboration tools – A collaboration tool is asynchronous when its users are collaborating at a different time: e-Mail, mailinglists and newsgroups, group calendar, workflow systems, hypertext.
- ii. Synchronous collaboration tools – A collaboration tool is synchronous, when its users are collaborating at the same time: shared whiteboard, video communication systems, chat systems, decision support systems, multiplayer video games.

1.1.2 Online Collaboration Tools

Online collaboration tools are web-based applications that offer basic services such as instant messaging for groups, mechanisms for file sharing and collaborative search engines (CSE) to find information distributed within the system of the organization, community or team. Additionally, the functionality is sometimes further expanded by providing for example integrated online

calendars, shared online-whiteboards to organize tasks and ideas or internet teleconferencing integrations.

The focus of online collaboration tools ranges from simple to complex, inexpensive to expensive, locally installed to remotely hosted and from commercial to open source.

Collaboration software or tool isn't exactly a clear-cut category because it overlaps with several other software categories such as: project management software, to-do list apps (as long as they have ample sharing and collaborating features), and kanban apps, team messaging app.

1.2 WHAT IS TEAMS COLLABORATOR?

Teams Collaborator is an online collaboration tool providing facilities ranging from varying categories of collaboration tools. It is a mix type of asynchronous and synchronous collaboration tools. Scheduling video calls will help keep synchronous web collaboration from disrupting workflow and aggravating teammates with too much digital chatter. Remote workers in other time zones can participate freely in ongoing, asynchronous dialogues.

So, our web-based collaboration tool will be providing the following features to all the organizations which will subscribe for its package:

- i. making audio and video calls
- ii. messaging facility to individual or posting it to a team
- iii. project management facility
- iv. creating different teams as per requirement
- v. getting regular notifications on any latest event
- vi. assignment allocation
- vii. file sharing facility
- viii. user-friendly interface

Hence, we can say that Teams Collaborator is a full package for an organization to level up its productivity.

1.3 WHY TEAMS COLLABORATOR?

It has become quite common to see employees working from all over the world on a single project. This increasingly dispersed workforce is becoming the trend and thus making online collaboration the need. Collaborating online is used to remove the gap between employees due to far-away locations, run effective meetings online and ensure that they work together at the same time on the

same page. Team Collaborator will help to run projects faster and will improve communication between employees.

Web collaboration tools for business can be simple or complex, depending on company's needs. For organizations seeking a platform to host contextual discussions and company-wide communication. With Teams Collaborator, we can create custom channels in a few clicks, and we are ready to start running with it. However, some teams may also need a web tool to coordinate projects, listing a project's bugs, its pending task, and etc. Since the collaboration module is built-in in Teams Collaborator, users can enjoy a one-stop-shop for both collaboration and project management.

1.4 ISSUES WITH OTHER COLLABORATION WEB APPLICATIONS

In offering all the functionalities and facilities in collaborative development there are a lot of problems and issues which have to be dealt with by the developers. These are the problems that arises in collaborative development and in social network-based development.

Following are the issues that most of the collaboration applications meets with and are the reason for project's failure:

- i. Updated Notification – Each collaborative network provides a real time notification usually via email if there are any changes in the user's profile. However, when these notifications are used in a negative way then they become problematic for the social network user.
- ii. Lack of Innovation – Another terrible reality regarding open source and online collaboratively developed is that these systems are reverse-engineered and poorly written copies of existing commercial software. Novelty is hardly a mere feature in collaborative developed projects.
- iii. Lack of Formal Mechanism – In social and collaborative development environment the overall development framework always lacks some formal system development and software engineering processes.
- iv. Overloading a team with too many options can actually make the working environment less organized. Scattering assets and insights in different screens, browser tabs, and apps can make it harder to get prompt answers.
- v. Simply having an unintuitive, clunky interface and being too expensive or requiring too much hardware.

1.4.1 How is Teams Collaborator Different?

Teams Collaborator addresses all the above described issues in an efficient and distinctive manner as described below:

- i. The Teams Collaborator do provide notifications about any modifications in user's profile but is also meticulous about the privacy of the information and whether the mail is being sent to the righteous place.
- ii. Teams Collaborator is a web-based collaborator application providing some basic features along with novel features. It provides general video and audio call, chat messaging, file sharing facility along with new project management, assignment allocation, teams creation features.
- iii. The development process of collaborator application followed the Waterfall Model of SDLC. It is the most organized way of developing software and applications.
- iv. The features which are embedded were selected precisely so that the application doesn't get complex and overloaded but still provide a full package tool experience to user.
- v. Teams Collaborator provides a simple, interactive, goal-specific and user-friendly interface. It is a cross platform web application and is affordable to all as well.

CHAPTER – 2

LITERATURE REVIEW

While Internet tools are ushering in new capabilities for collaboration, knowledge creation, and collective intelligence, traditional collaborative activities are also being enhanced by these tools. Participants can connect and communicate through more and richer channels to augment conversation. In addition to communication by talking, collaborators can add video, shared desktops, shared PowerPoint slides and flowcharts, and group Internet browsing.

Important improvements are those that permit or promote new and better ways of doing things. The tool that allows cheaper voice communication may gain a foothold against a more expensive tool. The tool that permits sharing of files during a voice communication may win fans for its convenience. The tool that seamlessly integrates with other tools is more likely to be adopted. 10 Collaboration Tools 11 Ultimately a tool should respect user time and reflect the values of the user. While a tool may appear to be a neutral communication medium, the uses that fans and users come up with is likely greatly influenced by the features and capabilities of the tool itself.[1]

Pappas, Menelaos[2] along with their other members researched, a web-based virtual collaborative platform called DiCoDEv - Distributed Collaborative Design Evaluation platform that can be used during manufacturing product and process design evaluation. Providing a multi-user real-time collaboration as well as a VR-based product and process verification, this platform is an integrated tool for designers, engineers and managers. The platform's integration into the VR environment enables the immersion and interaction of users with the virtual prototypes that lead to the efficient evaluation of product and process designs where the human intervention is crucial.

Cemil UZUN[3] reviewed that collaboration technologies like wikis, portals encourage the worker to share of knowledge, experience, and skills and work together throughout the organization. Collaboration tools and platforms help organization to convert from organizational tacit knowledge to organizational explicit knowledge with collaborative sharing environment and communication among workers in organization. Well planned strategic collaborative platform enhances knowledge workers to find easily and rapidly where related content, they become much more productive and collaborative with each other to improve organizational performance.

Considering communication and process management problems a framework is created for the students to conduct management, communication and collaboration between the institute, the

consultant and the department during the thesis studies or the lesson period studies. Problems in the management process have been solved with various collaboration tools and corporate social networks created. Furthermore, through the created information system, the institutes and departments are more easily monitored, reported and directed to the students' interest processes.[4]

Saad Razzaq[5] and his partners found that lack of coordination and communication facilities, poor usability, and low quality, less market adoption, lack of support and usage of formal methods are the major problem areas in collaborative software development. They recommended the use of web 2.0 tools and features for better communication and coordination. Distributed versioning system and bug tracking system should be incorporated to enhance the quality of collaboratively developed applications.

CHAPTER – 3

WEB APPLICATION SPECIFICATION

3.1 OVERVIEW

Every software or application that is created follows a Software Development Life Cycle (SDLC). It refers to the period of time that starts when a software product is conceived and ends when the product is no longer available for use. It typically includes a requirement phase, design phase, implementation phase, test phase, installation and check-out phase, operation and maintenance phase, and sometimes retirement phase.

There are various steps involved in the SDLC:

- i. Planning and Requirement Analysis - Requirement analysis is the most fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer and other departments. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas. Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.
- ii. Defining Requirements - Next step is to clearly define and document the product requirements and get them approved by the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.
- iii. Designing the Product Architecture - SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually, more than one design approach for the product architecture is proposed and documented in an SDD – Software Designing Document.

This SDD is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget, and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third-party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in SDD.

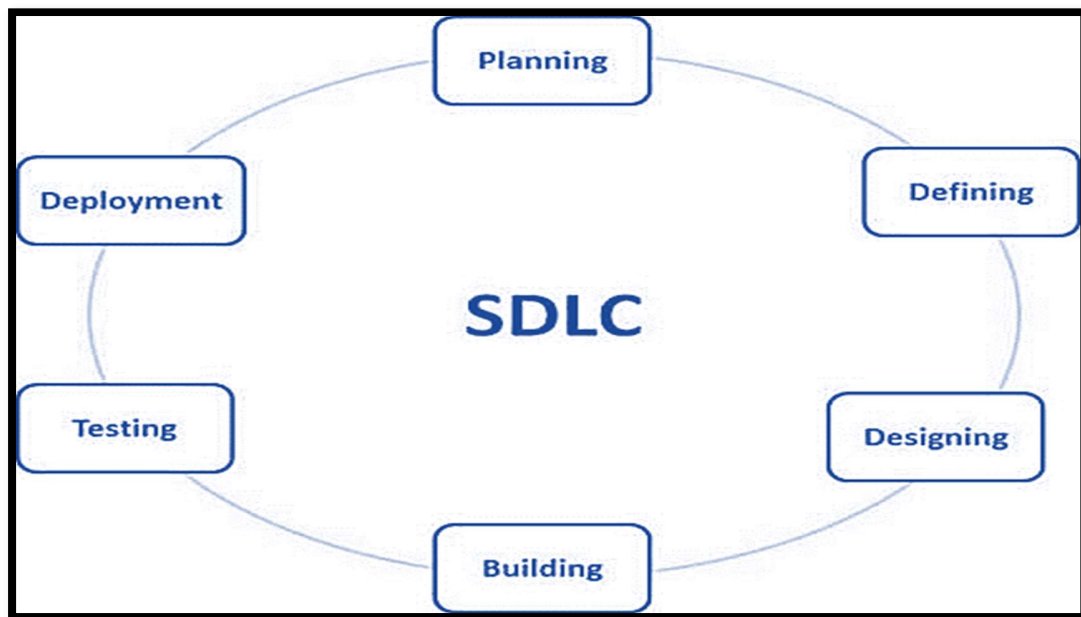


Fig. 3.1 Steps involved in SDLC

- iv. **Building or Developing the Product** - In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per SDD during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.
Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high-level programming languages such as C, C++, Java, and Python are used for coding. The programming language is chosen concerning the type of software being developed.
- v. **Testing the Product** - This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.
- vi. **Deployment in the Market and Maintenance** - Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first

be released in a limited segment and tested in the real business environment (UAT- User acceptance testing). Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

After following all these steps, we get in our hands for what we worked on.

3.2 FEASIBILITY STUDY OF THE PROJECT

The feasibility of the application is checked on various parameters like:

- i. Economical: The development of the collaborator application did not require any amount to paid as all the technical support material is free of cost and no extra hardware is required. Therefore, it is economically feasible.
- ii. Technology: The technologies to be used are available online and are open source, not complex to implement, and are not outdated. The application has real-world requirements that are feasible. Therefore, the application is technically feasible and after its release can be maintained easily for a long time.
- iii. Hardware and Software: The application does not pose any specific requirement for hardware as it will possess cross-platform feature. A computer system hardware with general system specification, an internet connection, speakers, microphone and camera are required to work on application.
The software required during the development period are free of cost, open source, and easily available on the internet. Therefore, it's all hardware and software requirements are feasible.
- iv. Social: The product is a web-based application that will have worldwide access. The application as a whole with its features is going to be unique in itself and will not be in any contradiction with any other available applications. It will be focusing on every type of organization such as institutes, schools, colleges, tech companies, firms. Therefore, it is a socially feasible and acceptable product.
- v. Legal: Everything that is required for the project is under legal terms and conditions. No pirated software will be used in the creation of application. All work done is authentic and is not a duplicate of any other collaborator application. Therefore, legally feasible .

Hence, the project is all way feasible to design, implement and maintain.

3.3 SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) OF THE PROJECT

3.3.1 Hardware Requirements

Selecting the suitable and best hardware is very important for the existence and proper working of any software because only after that we can specify the hardware for the customers on which the application can be used. It's a good practice to create the application or software to be device-friendly.

Hence the web application created is cross-platform, i.e., it can be executed on any device or on any hardware type with an internet connection, microphone, camera and speakers to use the video conferencing and audio call feature to its fullest.

The general device hardware requirements for running the web application are:

Processor	AMD A6-9225 RADEON R4, 5 COMPUTE CORES 2C+3G 2.60 GHz
RAM	4.00 GB
Input device	Camera, microphone
Output device	Speakers

To host the application server requirements will be:

Processor	Intel Pentium
RAM	1 GB
ROM	25 GB

The application will be functional on every type of operating system such as Windows, Mac, Android, iOS, etc.

3.3.2 Software Requirements

The softwares that were used in the creation of web-based collaborator application are:

- i. VS Code – Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
 - Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

- Visual Studio Code is a source code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, and C++. It is based on the Electron framework, which is used to develop Node.js web applications that run on the Blink layout engine.
 - Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language.
 - It supports several programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings.
 - Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.
 - To save switching between windows, VS Code offers an integrated terminal or CLI. This makes it easy to install NPM or Yarn dependencies, commit files to Git, and push files to Github — as well as anything else you might want to do via the command line.
- ii. Google Chrome – Google Chrome is a cross-platform web browser developed by Google. The browser is also the main component of Chrome OS, where it serves as the platform for web applications.
- Google Chrome is a web browser for mobile as well as desktop users. It is mostly used for accessing internet.
 - It is very fast and secured application that comes with preinstalled Adobe Flash. It has automatic language translation which helps in accessing the webpage in different language.
 - HTML is the primary file extension of Google Chrome web browser.
 - When you are working on Google Chrome, the key combination ‘shift + esc’ will find Google task manager with the list of currently opened websites.
 - The combination ‘ctrl + shift + N’ opens Incognito mode. This mode does not save history of your browsed sites and hence offers you a way of accessing website privately.

3.3.3 Technologies Required

- i. SQLite – SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client-server database engine. Rather, it is embedded into the end program.
 - SQLite generally follows PostgreSQL syntax. SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee domain integrity. This means that one can, for example, insert a string into a column defined as an integer.
 - SQLite will attempt to convert data between formats where appropriate, the string "123" into an integer in this case, but does not guarantee such conversions and will store the data as-is if such a conversion is not possible.
 - SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others.
 - SQLite has bindings to many programming languages.
 - SQLite uses an unusual type system for a SQL-compatible DBMS: instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms, it is dynamically typed.
- ii. Django – Django is a Python-based free and open-source web framework that follows the model–template–views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established in the US.
 - Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself.
 - Python is used throughout, even for settings, files, and data models.
 - Despite having its own nomenclature, such as naming the callable objects generating the HTTP responses "views", the core Django framework can be seen as an MVC architecture.
 - It consists of an object-relational mapper (ORM) that mediates between data models (defined as Python classes) and a relational database ("Model"), a system

for processing HTTP requests with a web templating system ("View"), and a regular-expression-based URL dispatcher ("Controller").

- Django maintains the website through URLs rather than the IP addresses on the server, which makes it easy for SEO engineers to add the website to the server while the web-developer don't have to convert the URL into some numeric code.

3.4 SDLC MODEL USED FOR THE PROJECT

There are several types of Software Development Life Cycle models. Each one has its own set of advantages and disadvantages. Though the approach might differ between models, each methodology has the goal of enabling the development team to deliver high-quality software in a cost-effective and quick manner.

The SDLC model which we followed for the development of project is Waterfall Model. The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use.

This model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

The sequential phases in the Waterfall model are as follows:

- i. Requirement Gathering and analysis – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- ii. System Design – The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- iii. Implementation – With inputs from the system design, the system is first developed in small programs called units, which are integrated with the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- iv. Integration and Testing – All the units developed in the implementation phase are integrated into a system after testing each unit. Post integration the entire system is tested for any faults and failures.

- v. Deployment of system – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- vi. Maintenance – Some issues come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

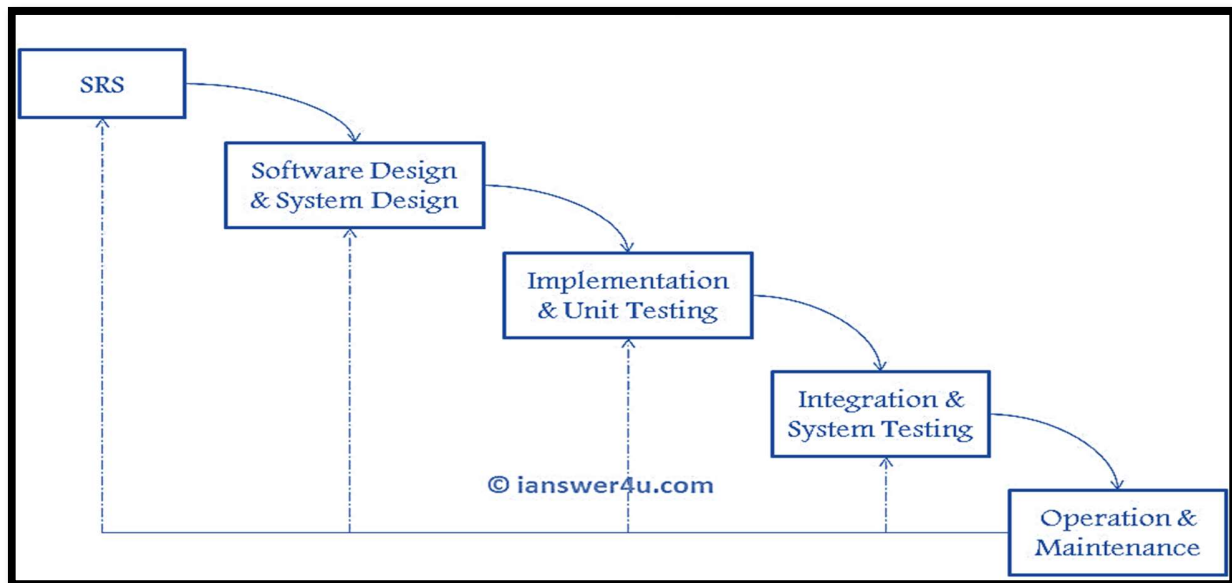


Fig. 3.4.1 Waterfall Model

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. In this model, all the requirements need to be specified in the beginning only. Except for phase one, there is no interaction or feedback from customers.

Waterfall model is most appropriate when:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.

CHAPTER – 4

SOFTWARE DESIGN AND ANALYSIS

4.1 OVERVIEW

Software analysis and design includes all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human readable and understandable documents, to which a computer has nothing to do. Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.

Software design is the solution to software requirement specification document. This phase focuses on the detailed implementation of the feasible system. Its emphasis on translating design specification to performance specification. Software system design has two phases of development

- Logical design
- Physical design

During logical design phase the analyst describes inputs (sources), outputs (destinations), databases (data stores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams, entity relationship diagrams and database design.

The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

4.2 DATA FLOW DIAGRAMS (DFD)

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.

There is a prominent difference between DFD and Flowchart. The flowchart depicts flow of control in program modules. DFDs depict flow of data in the system at various levels. DFD does not contain any control or branch elements.

4.2.1 DFD Components

DFD can represent Source, destination, storage and flow of data using the following set of components:

- i. Entities - Entities are source and destination of information data. Entities are represented by a rectangle with their respective names.
- ii. Process - Activities and action taken on the data are represented by Circle or Round-edged rectangles.
- iii. Data Storage - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.
- iv. Data Flow - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

4.2.2 Levels of DFD

- i. Level 0 - It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

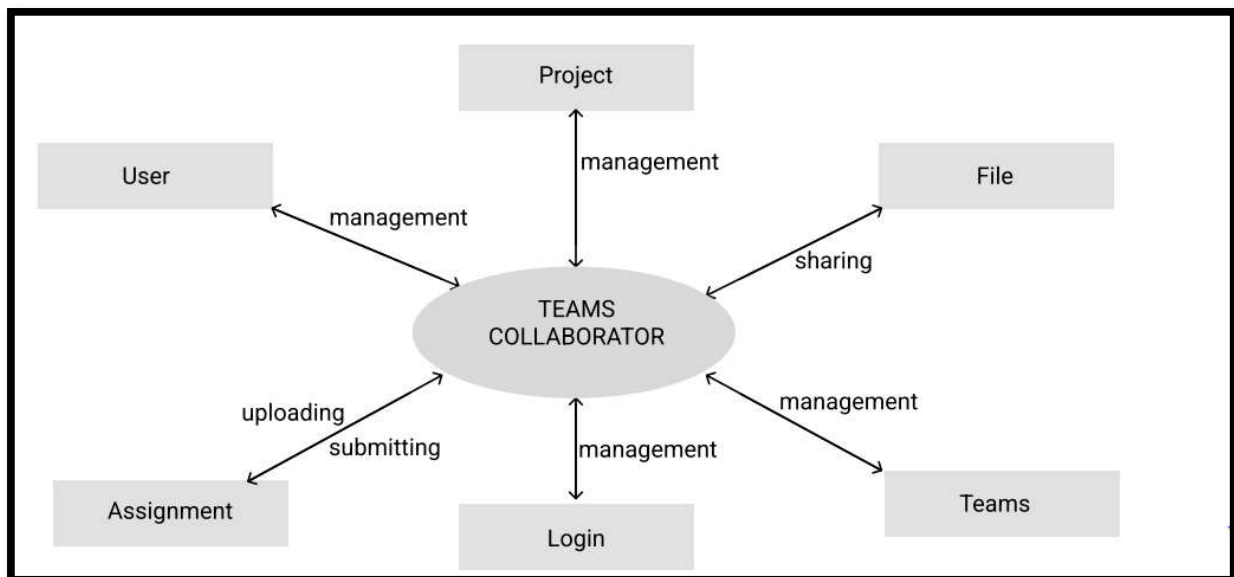


Fig. 4.2.2.1 Level 0 DFD

- ii. Level 1 - In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses. Level 1 DFD depicts basic modules in the system and the flow of data among various modules.

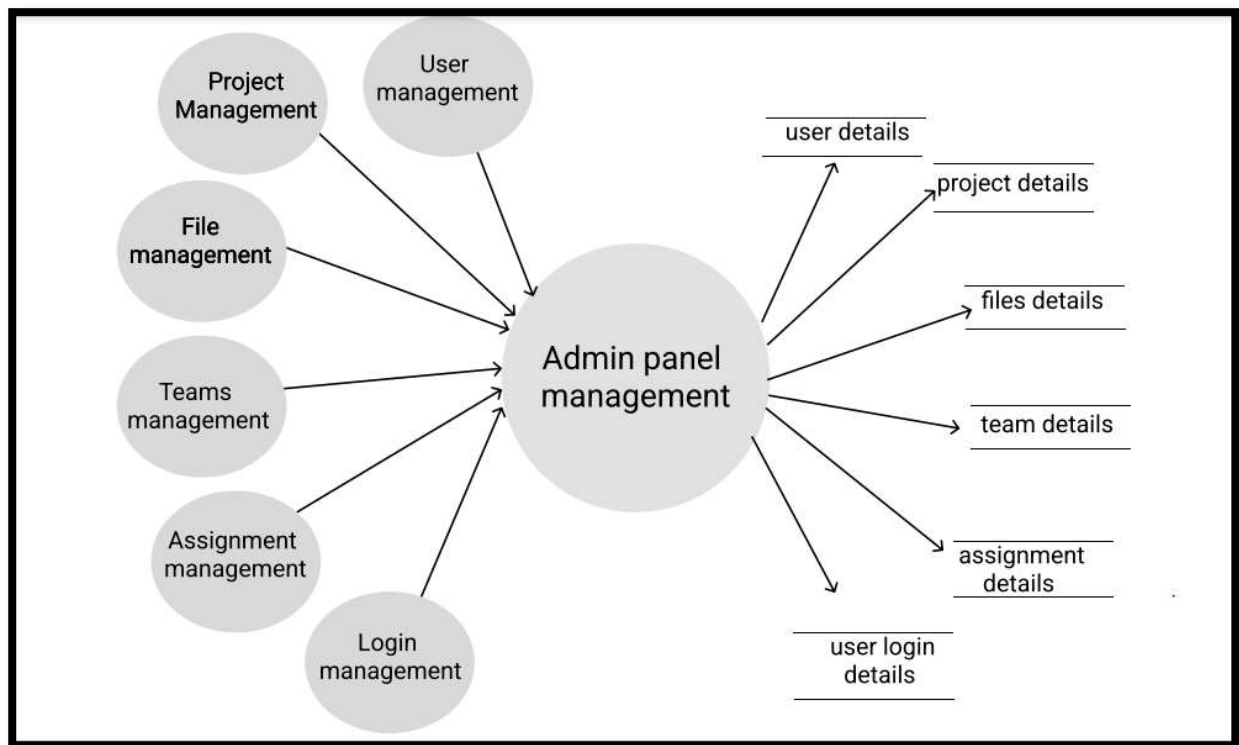


Fig.4.2.2.2 Level 1 DFD

- iii. Level 2 - At this level, DFD shows how data flows inside the modules mentioned in Level 1. Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved. It can be used to plan or record the specific/necessary detail about the system's functioning. We have shown the admin panel management process in level 2 DFD.

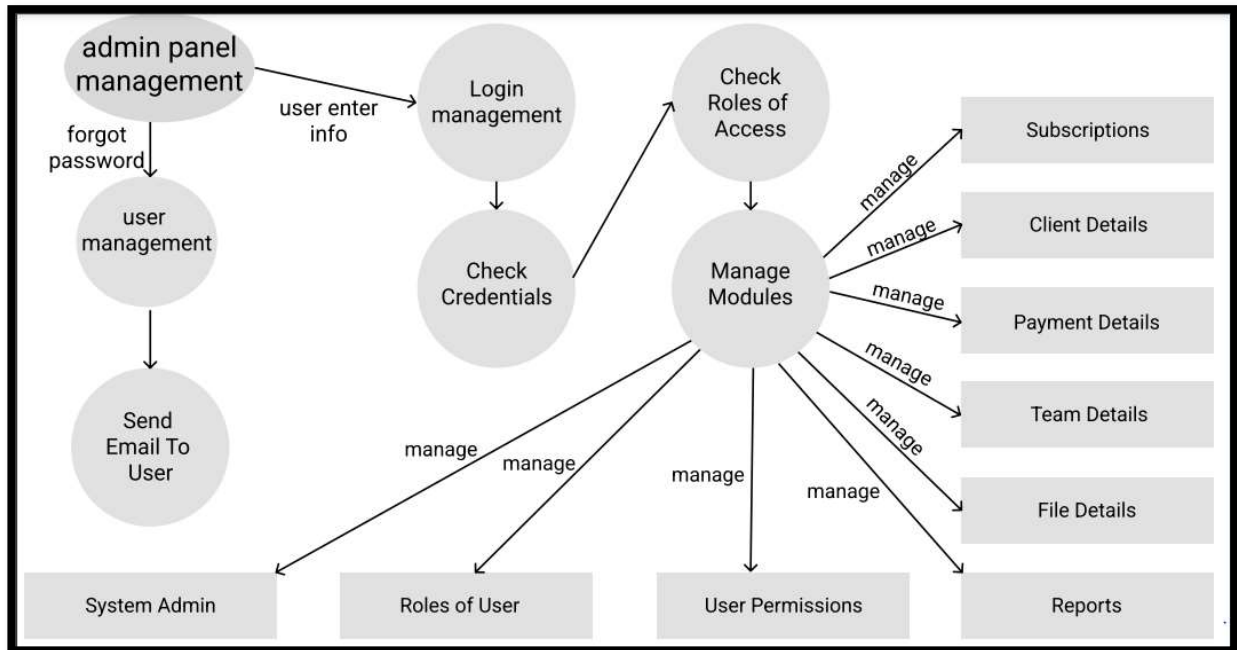


Fig.4.2.2.3 Level 2 DFD

4.3 INPUT AND OUTPUT DESIGN

4.3.1 Input Design

The input design is the link between the information system and the user. It is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulation can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database.

Thus, the objective of input design is to create an input layout that is easy to follow. It involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility.

4.3.2 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system, results of processing are communicated to the users and to other system through outputs.

In output design it is determined how the information is to be displayed for immediate need. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system easy and efficient to use. The output form of an information system should accomplish one or more of the following objectives:

- i. Convey information about past activities, current status or projections of the future.
- ii. Signal important events, opportunities, problems, or warnings.
- iii. Trigger an action.
- iv. Confirm an action.

4.4 DATABASE DESIGN

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are: the field that is unique for all the record occurrences; and the field used to set relation between tables. Normalization is a technique to avoid redundancy in the tables.

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed databases are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database design are to produce logical and physical designs models of the proposed database system:

- The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.
- The physical data design model involves translating the logical DB design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).

In the database structure of our web collaborator application we have the following tables:

- Assignment Table – to store the data about the assignments created by teams for the users. The data of the assignments page will be stored here. Primary key is id.

main_assignment			CREATE TABLE "main_assignment" ("id" integer NOT NULL PRIMARY KEY AUTOINCR
id	integer		"id" integer NOT NULL
assignmentname	varchar(500)		"assignmentname" varchar(500) NOT NULL
assignmentdescription	varchar(5000)		"assignmentdescription" varchar(5000) NOT NULL
assignedby	varchar(50)		"assignedby" varchar(50) NOT NULL
assignedto	varchar(50)		"assignedto" varchar(50) NOT NULL
complete	bool		"complete" bool NOT NULL
datecreated	datetime		"datecreated" datetime NOT NULL
datesubmission	datetime		"datesubmission" datetime NOT NULL

Table 4.4.1 Assignment table description

- File Table – to store data about all shred filed documents. Data of file page will be stored here. Primary key is id.

main_file			CREATE TABLE "main_file" ("id" integer NOT NULL PRIMARY KEY AUTOINCR
id	integer		"id" integer NOT NULL
filename	varchar(500)		"filename" varchar(500) NOT NULL
filetype	varchar(50)		"filetype" varchar(50) NOT NULL
filelink	varchar(100)		"filelink" varchar(100) NOT NULL
uploadedby	varchar(50)		"uploadedby" varchar(50) NOT NULL
datecreated	datetime		"datecreated" datetime NOT NULL

Table 4.4.2 File table description

- iii. Project details – to store the information about all the project task details. Data of tasks of a particular project will be stored here. Primary key is id.








main_projectassignment			CREATE TABLE "main_projectassignment" ("id" integer NOT NULL PRIM
 id	integer	"id" integer NOT NULL	
 projectname	varchar(50)	"projectname" varchar(50) NOT NULL	
 assignmenttype	varchar(250)	"assignmenttype" varchar(250) NOT NULL	
 assignmentdescription	varchar(1000)	"assignmentdescription" varchar(1000) NOT NULL	
 assignmentseverity	varchar(50)	"assignmentseverity" varchar(50) NOT NULL	
 comment	varchar(5000)	"comment" varchar(5000) NOT NULL	
 datecreated	datetime	"datecreated" datetime NOT NULL	

Table 4.4.3 Project details table description

- iv. Project Table – to store data about the projects. All the data of listed project in project page will be stored here. Primary key is id.







main_projectdetails			CREATE TABLE "main_projectdetails" ("id" integer NOT NULL, "projec
 id	integer	"id" integer NOT NULL	
 projectname	varchar(50)	"projectname" varchar(50) NOT NULL	
 projectdescription	varchar(5000)	"projectdescription" varchar(5000) NOT NULL	
 datecreated	datetime	"datecreated" datetime NOT NULL	
 projectmanager	varchar(50)	"projectmanager" varchar(50) NOT NULL	
 deadline	varchar(50)	"deadline" varchar(50) NOT NULL	

Table 4.4.4 Project table description

- v. Team Table – to store data about the teams created. Data of all the teams listed on team page will be stored here. Primary key is id.





main_team			CREATE TABLE "main_team" ("id" integer NOT NULL PRIMARY KEY AUT
 id	integer	"id" integer NOT NULL	
 teamname	varchar(50)	"teamname" varchar(50) NOT NULL	
 members	varchar(5000)	"members" varchar(5000) NOT NULL	
 teamadmin	varchar(50)	"teamadmin" varchar(50) NOT NULL	

Table 4.4.5 Team table description

- vi. User Table – to store user details when they create account on web application. Data coming from sign up page is stored in here. Primary key is id.

auth_user			CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTO
id	integer		"id" integer NOT NULL
password	varchar(128)		"password" varchar(128) NOT NULL
last_login	datetime		"last_login" datetime
is_superuser	bool		"is_superuser" bool NOT NULL
username	varchar(150)		"username" varchar(150) NOT NULL UNIQUE
last_name	varchar(150)		"last_name" varchar(150) NOT NULL
email	varchar(254)		"email" varchar(254) NOT NULL
is_staff	bool		"is_staff" bool NOT NULL
is_active	bool		"is_active" bool NOT NULL
date_joined	datetime		"date_joined" datetime NOT NULL
first_name	varchar(150)		"first_name" varchar(150) NOT NULL

Table 4.4.6 User table description

- vii. User extra detail Table – to store all users extra modified details. Primary key is id.

main_userdetails			CREATE TABLE "main_userdetails" ("id" integer NOT NULL PRIMARY KE
id	integer		"id" integer NOT NULL
username	varchar(50)		"username" varchar(50) NOT NULL
userinitial	varchar(5)		"userinitial" varchar(5) NOT NULL
firstname	varchar(50)		"firstname" varchar(50) NOT NULL
lastname	varchar(50)		"lastname" varchar(50) NOT NULL

Table 4.4.7 User extra details table description

4.5 ENTITY RELATIONSHIP DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining

requirements for an information systems project. It's also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.

ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong.

4.5.1 Components of ER Diagrams

ER Diagrams are composed of entities, relationships and attributes.

- i. Entity - An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. All these entities have some attributes or properties that give them their identity. An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values.

They are represented by means of rectangles. Rectangles are named with the entity set they represent.

- ii. Relationship - The association among entities is called a relationship. A set of relationships of similar type is called a relationship set. A relationship can also have attributes. These attributes are called descriptive attributes.

These are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

- iii. Attributes - Entities are represented by means of their properties, called attributes. All attributes have values.

These are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).

Given below is the ER diagram of e-commerce web application. The various relationship represented in the ER diagram are described as follows:

- a. Comprises of – it is many-to-many relationship between users and teams because users can have many teams and teams can have many users. Participation from both users and teams side is total.
- b. Have – it is a many-to-many relationship between users and assignment because users can have many assignments and assignments can have many users. There is partial participation

from user side because there can be users who do not have any assignment and total participation from assignment side.

- c. Uploaded - it is one-to-many relationship between users and files because users can upload many files and a file will be uploaded by one user only. There is partial participation from user side because there can be users who do not upload any file but every file uploaded will have a user so total participation from file.
- d. Have – it is a many-to-many relationship between users and projects because user can have many projects project can have many users. There is partial participation from user side because there can be user who do not have any project but all project will always have a user.
- e. Consists of – it is a one-to-many relationship between project and project details because there can be many project details but a project detail will only have one project under which it is specified. There is total participation from both sides.

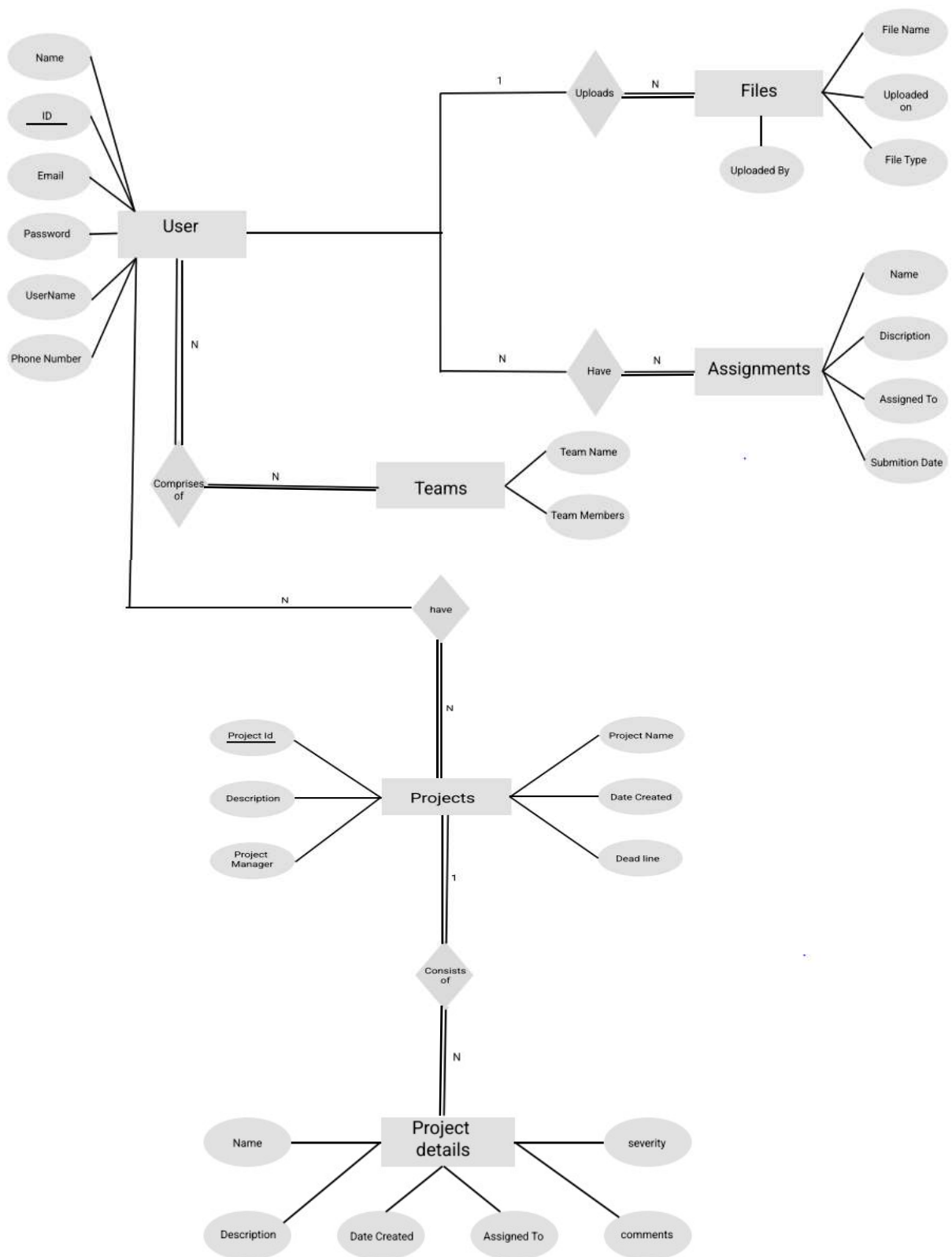


Fig. 4.5.1.1 ER Diagram of project

CHAPTER – 5

PROJECT FEATURES AND FUNCTIONALITY

5.1 HIGH-LEVEL LANGUAGES USED IN PROJECT

- i. HTML - The Hyper Text Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.
 - Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.
 - HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page.
 - HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements.
 - Browsers do not display the HTML tags, but use them to interpret the content of the page.
 - HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages.
 - Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.
 - HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references.
 - HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some represent empty elements and so are unpaired, for example ``. The first tag in

such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).

- Another important component is the HTML document type declaration, which triggers standards mode rendering.
- ii. CSS - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
- CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.
 - Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices.
 - CSS also has rules for alternate formatting if the content is accessed on a mobile device.
 - The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.
 - The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.
 - In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.
 - Each web browser uses a layout engine to render web pages, and support for CSS functionality is not consistent between them. Because browsers do not parse CSS

perfectly, multiple coding techniques have been developed to target specific browsers with workarounds (commonly known as CSS hacks or CSS filters). Adoption of new functionality in CSS can be hindered by lack of support in major browsers.

iii. JavaScript - JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled and multi-paradigm. It has dynamic typing, prototype-based object-orientation and first-class functions.

- Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often incorporating third-party libraries.
- All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.
- As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).
- The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.
- JavaScript engines were originally used only in web browsers, but they are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.
- Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.
- JavaScript supports much of the structured programming syntax from C (e.g., if statements, while loops, switch statements, do while loops, etc.). One partial exception is scoping: originally JavaScript only had function scoping with var; then block scoping was added in ECMAScript 2015 with the keywords let and const.

- Like C, JavaScript makes a distinction between expressions and statements. One syntactic difference from C is automatic semicolon insertion, which allows semicolons (which terminate statements) to be omitted.
 - Over 80% of websites use a third-party JavaScript library or web framework for their client-side scripting.
 - jQuery is by far the most popular library, used by over 75% of websites. Facebook created the React library for its website and later released it as open source; other sites, including Twitter, now use it. Likewise, the Angular framework created by Google for its websites, including YouTube and Gmail, is now an open source project used by others.
 - In contrast, the term "Vanilla JS" has been coined for websites not using any libraries or frameworks, instead relying entirely on standard JavaScript functionality
- iv. Python - Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
- Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming.
 - It is often described as a "batteries included" language due to its comprehensive standard library.
 - Python consistently ranks as one of the most popular programming languages.
 - Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation.
 - Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but are rarely, if ever, used. It has fewer syntactic exceptions and special cases than C or Pascal.
 - Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may

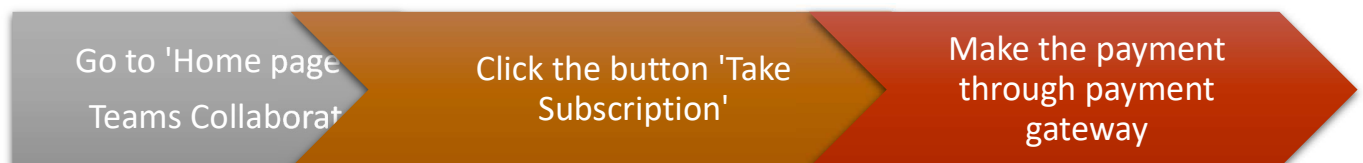
fail, signifying that the given object is not of a suitable type. Despite being dynamically-typed, Python is strongly-typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.

- Python allows programmers to define their own types using classes, which are most often used for object-oriented programming. New instances of classes are constructed by calling the class, and the classes are instances of the metaclass type, allowing metaprogramming and reflection.

5.2 VARIOUS FEATURES OF TEAMS COLLABORATOR

5.2.1 'Home Page' of Teams Collaborator

- Whenever a user will open the web application the home page of Teams Collaborator will be displayed as shown in below figure. The page will list the features provided by application.
- It will also show two 'subscription offer' cards. One subscription offer will be for user having team of ten members. Other one will be for users having more than ten members in a team.
- When a user clicks to subscribe for small team members then a payment gateway will be displayed to pay the amount of the subscription offer. Once payment is been done the user will be redirected to a 'sign up' page to register himself in the application. Once registered, he will be directed to 'home page' of core application part where all the features and work is being done.
- Else, when a user selects the subscription offer for large team he will be directed to mail service in order to contact the service provider about the subscription deal.
- Home page of Teams Collaborator will also contain a login button. So, all the registered users will be able to login directly to core application.



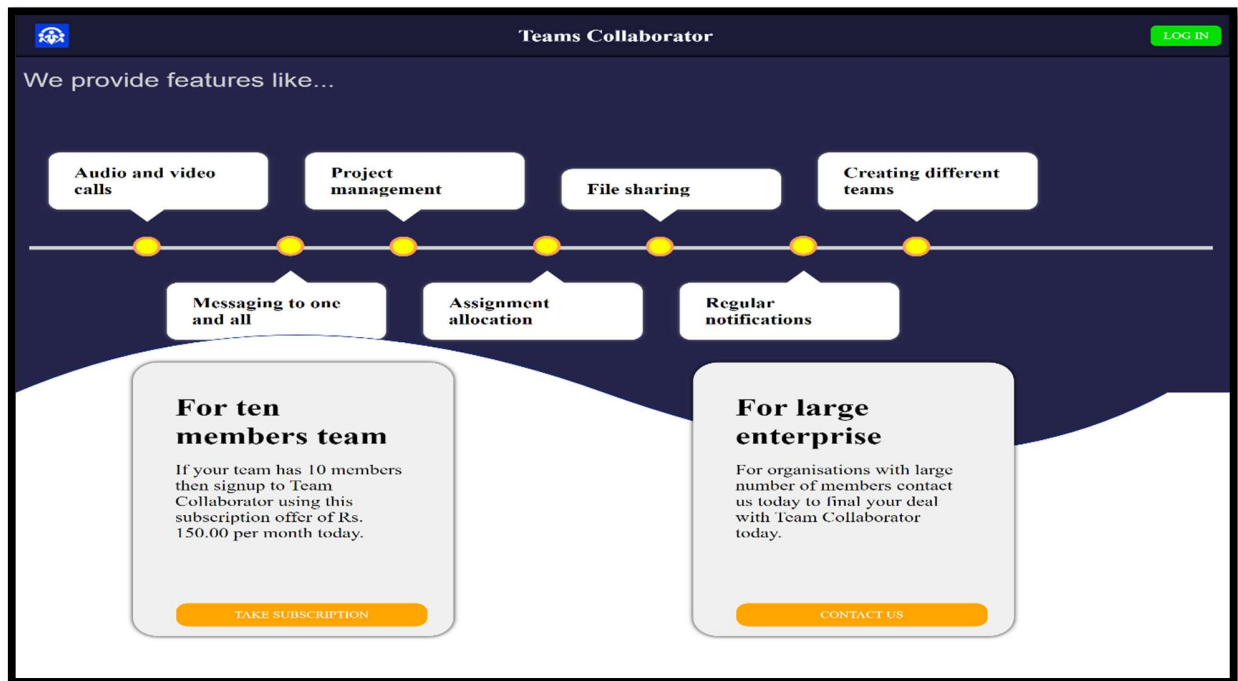


Fig.5.2.1.1 Image showing Home page of Teams Collaborator



Fig.5.2.1.2. Image showing payment being done through payment gateway

Function code to redirect to payment gateway:

```
param_dict = {
    'MID': 'My-Merchant-Id',
    'ORDER_ID': str(temp.orderid),
    'TXN_AMOUNT': "150",
    'CUST_ID': str(request.user.username),
    'INDUSTRY_TYPE_ID': 'Retail',
    'WEBSITE': 'WEBSTAGING',
    'CHANNEL_ID': 'WEB',
    'CALLBACK_URL': 'http://127.0.0.1:8000/payment/',
}

checksumhash = PaytmChecksum.generateSignature(param_dict, "KeyForGateway")
param_dict['CHECKSUMHASH'] = checksumhash
return render(request, "paytm.html", {'param_dict': param_dict})
```

Function code to checking whether payment is done or not:

```
@csrf_exempt
def handelrqst(request):
    form = request.POST
    response_dict = {}
    for i in form.keys():
        response_dict[i] = form[i]
        if i == 'CHECKSUMHASH':
            checksum = form[i]

    verify = PaytmChecksum.verifySignature(response_dict, "KeyForGateway", checksum)
    if verify:
        if response_dict['RESPCODE'] == '01':
            print('Subscription made successfully ' + response_dict['ORDERID'])
    return redirect("/")
```

5.2.2 'Notifications' Feature

- When the user has logged in he will be directed to home page of core application where actual work is being done.
- The home page will consist of several items such as navigation bar, header with logo, search bar and user name initials.
- The green signal along with username initials will signify that the user is active and is currently logged in to the application.
- The main content of home page will show the latest notifications telling about the events happening in application.
- Along with notifications a digital clock is also been displayed to check out the current date so as user doesn't get late for his scheduled work or meeting.

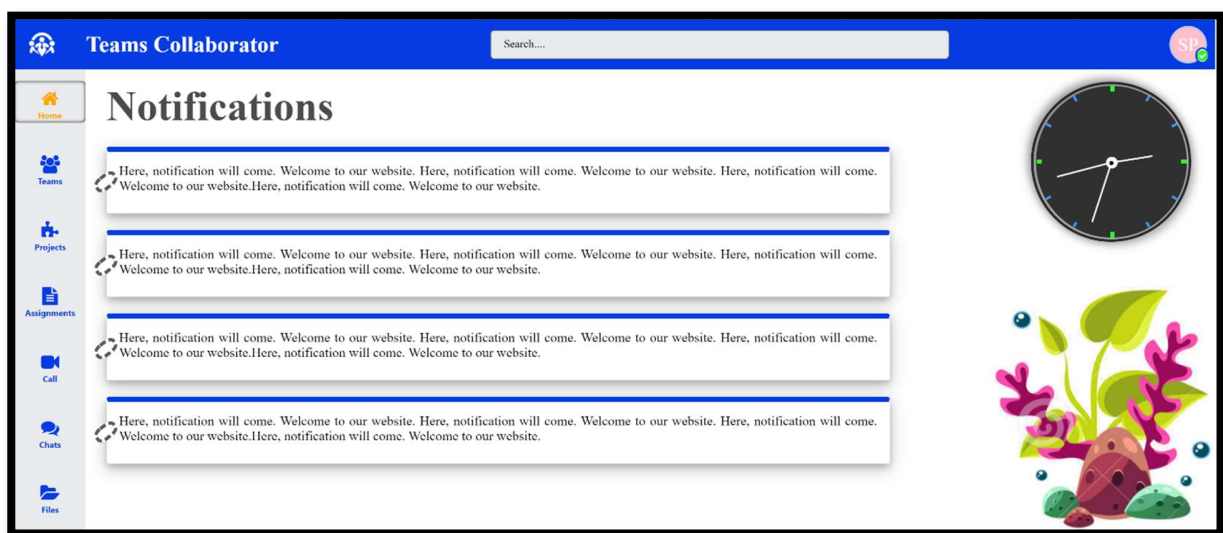


Fig. 5.2.2.1 Image showing Notifications feature

5.2.3 'Teams' Feature

- When a user selects the Teams option from navigation bar the teams page displays.
- A user can create many teams as per his needs.
- The teams page comprises of two parts one where all the team's name will be displayed like a contact list and above that is a search bar to search for a particular team; Other part contains the post made by team members.
- Each team will have a corresponding post section.

- When a user selects a team name from the given side list simultaneously the corresponding post section will be displayed.
- The post section contains a header displaying the name of the team and a video call option.
- It will also contain an input bar through which post will made.

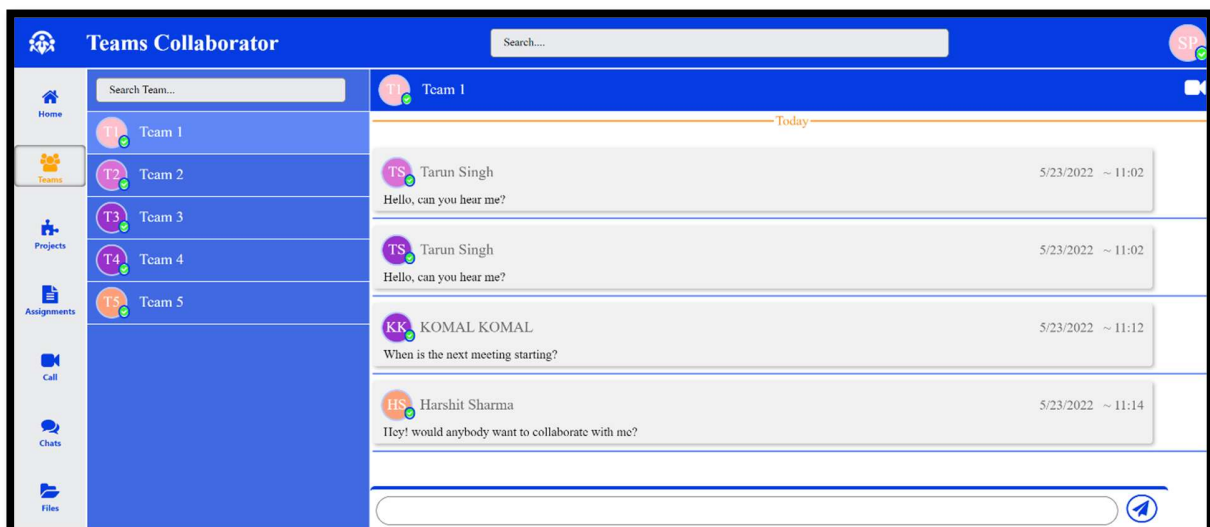
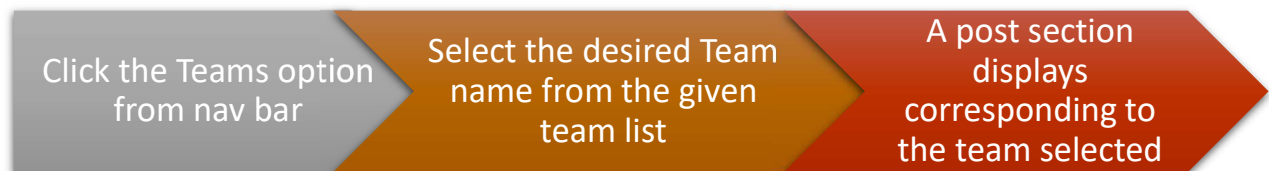


Fig. 5.2.3.1 Image representing Teams feature

Function code to display the teams page when Teams option selected from navigation bar:

@xframe_options_sameorigin

def teamspage(request):

teams = team.objects.all()

return render(request, 'teamspage.html', {'teams': teams})

On clicking the submit button user will be given a product catalog of best-suggested products as shown.

5.2.4 'Project Management' Feature

- This is a very useful and creative feature of our application.
- When a user selects the Projects option from navigation bar the project section is displayed.
- The project section contains the list of projects been created by user.
- The project list will be organized in rows having the name of project written on it.
- So, whenever the user selects a particular project the task board of particular project is displayed.
- The task board is a hierarchical abstracted view of project which is further divided into subcategories.
- It comprises of two parts: the first part contains four categories namely, backlog, to-do, doing and done and the other contains task details.
 - Backlog category – it will contain all the tasks that are pending in the project.
 - To-do category – the task present in this list are the ones which need to be completed.
 - Doing category – it will comprise of the tasks on which work is being done, i.e., the task which are under process.
 - Done category – the task present in this list are the one which have been worked upon, i.e., those tasks whose execution have been completed.
- The tasks are displayed in the form of container which contains the task number, its title and the name of the member to whom it is been assigned.
- Every task category contains the count of the number of tasks in its list.
- When a user clicks on a particular task then details of corresponding task is displayed on the task details section.
- Task details section contains the brief description of the task such as its number, the category to which it belongs, any screenshot to visually tell about the task where is to be done or how it is to done, a label of severity telling the how much important is the task to be done in priority basis, name of the member to which it is assigned, and a comment section where extra detail about task is given.
- This feature is helpful for managers or project leaders to keep an eye on the progress on the project, to know who are working on project.

- Hence, it helps in the management of projects and it is possible that a number of projects belongs to a user.

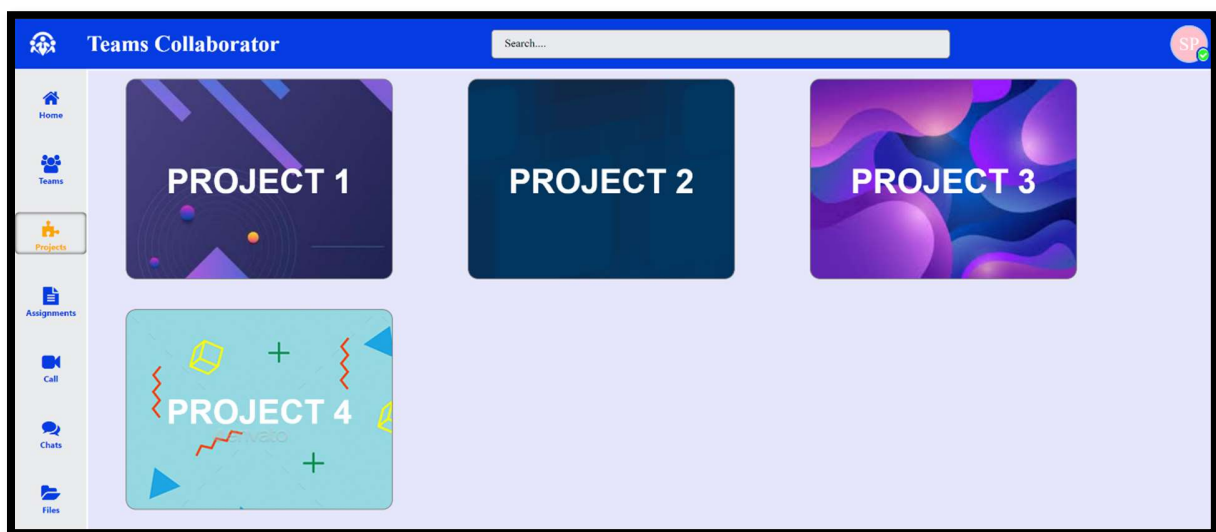
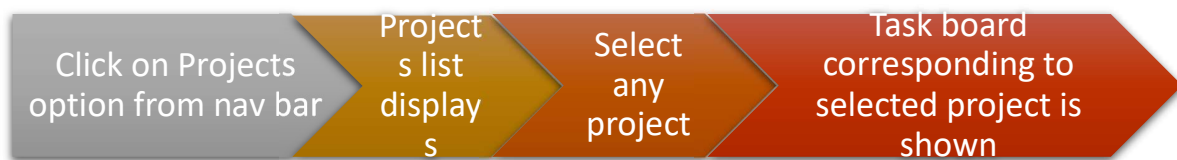


Fig.5.2.4.1 Image showing Projects page with a list of various projects

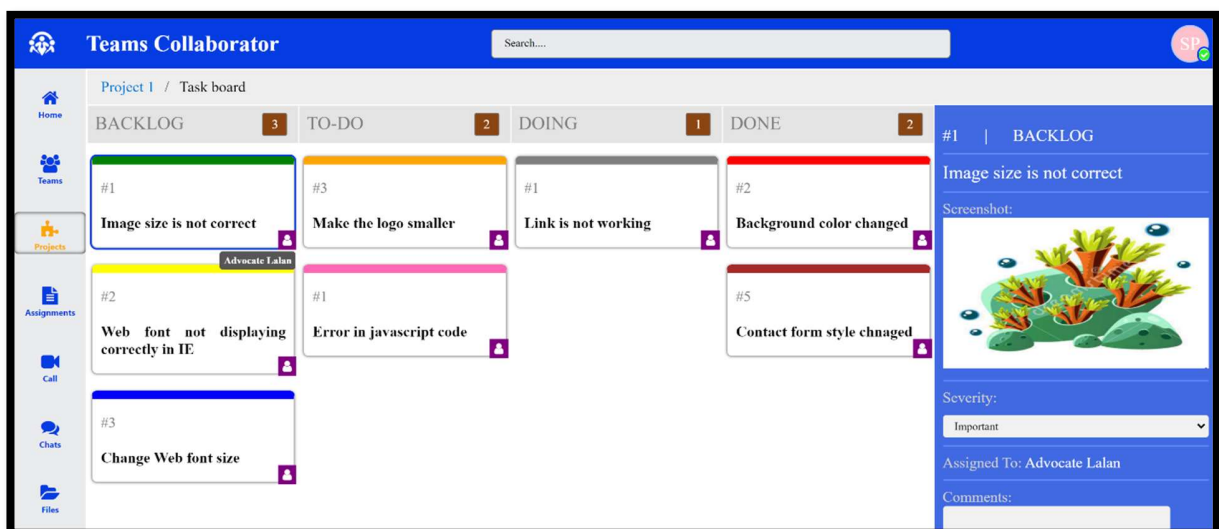


Fig.5.2.4.2 Image showing task board of a particular project with various tasks and task detail section

Function code to display projects page on selection of Projects option from navigation bar:

```
@xframe_options_sameorigin
```

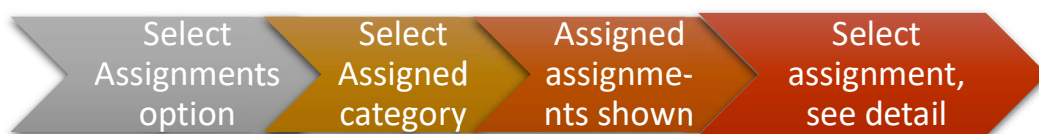
```
def projectmanagement(request):
```

```
    assigned = projectassignment.objects.all()
```

```
    return render(request, 'project management.html' , {'assigned': assigned})
```

5.2.5 ‘Assignment Allocation’ Feature

- Using this feature a user or organization will be able to assign assignments to their teams.
- When a user clicks on the Assignments option from the navigation bar, a assignment page displays showing two categories on the top: Assigned and Completed.
- The assigned category contains those assignments which have been recently assigned to the user. Assignments here will be having their title, name of the team from which it is assigned and its due date.
- The completed category contains those assignments which have been completed by user. Assignments in here will be having their title, name of the team from which it was assigned and completed tag.
- When a user selects a particular assignment under any category the assignment detail page opens up. It shows the full detail of the assignment.
- The assignment detail page will contain its title, the team name from which it has been assigned and due date or completed tag. Under this the description of the assignment will be given.
- The assignment detail page consists of two buttons:
 - one for adding a file ‘Add File’ button using which user can add his work in form of a file and
 - other to hand in the assignment ‘Hand In’ button using which the assignment with the attachments will be send back to the person whose assigned it and will move this assignment to the completed assignment category list. .



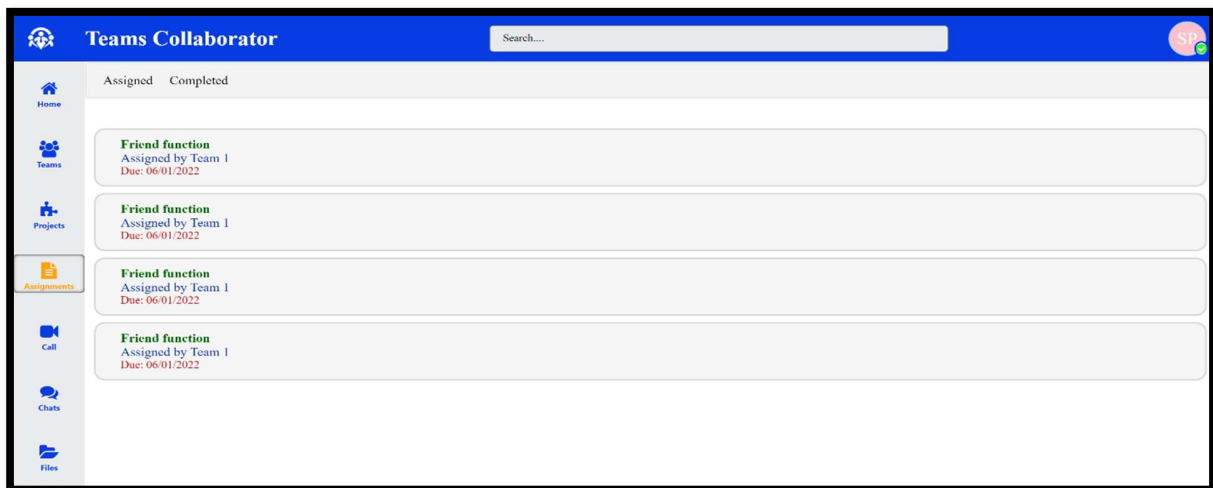


Fig.5.2.5.1 Image showing assigned category in Assignments page

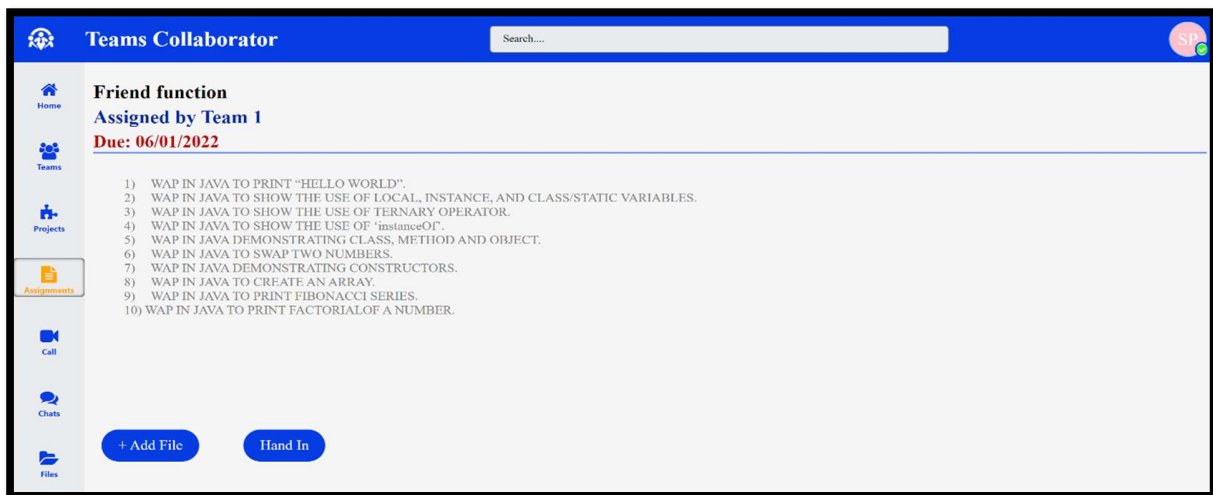


Fig. 5.2.5.2 Image assignment details page

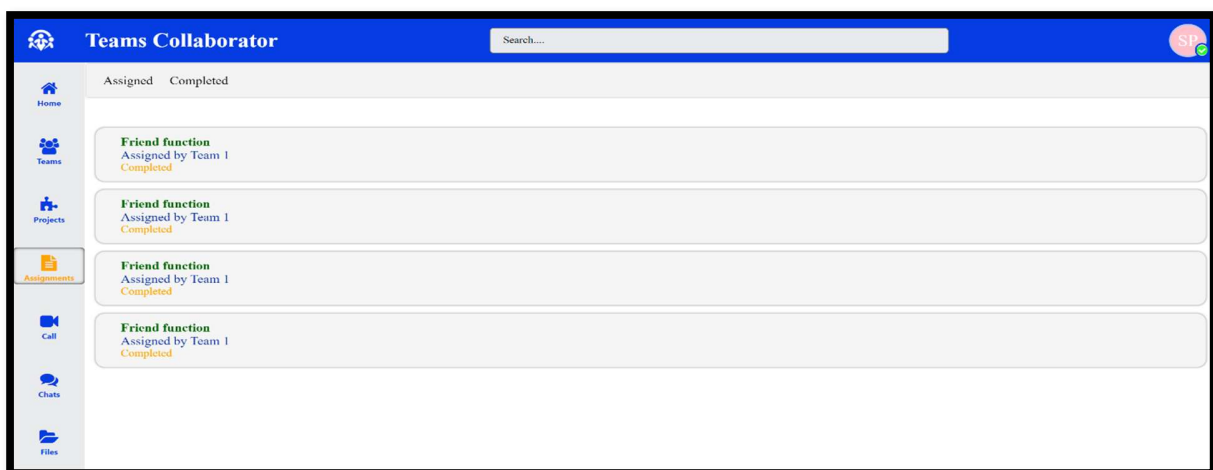


Fig. 5.2.5.3 Image showing completed category in Assignments page

Function code to display assignments page on selection of Assignments option from navigation bar:

```
@xframe_options_sameorigin
def assignmentpage(request):
    assignmentvar = assignment.objects.all().order_by('datecreated')
    return render(request, 'assignment page.html')
```

5.2.6 'Audio and Video Call' Feature

- Its an important feature of any collaboration application as without a communication system the collaboration will always have a void space.
- When a user selects the Call option from navigation bar a contact list is displayed.
- Each contact in contact list will have two calling options one for video call and other for audio call.
- The video call option will connect a video call to the selected contact with its video open by default.
- The audio call option will connect a video call to the selected contact with its video closed by default.

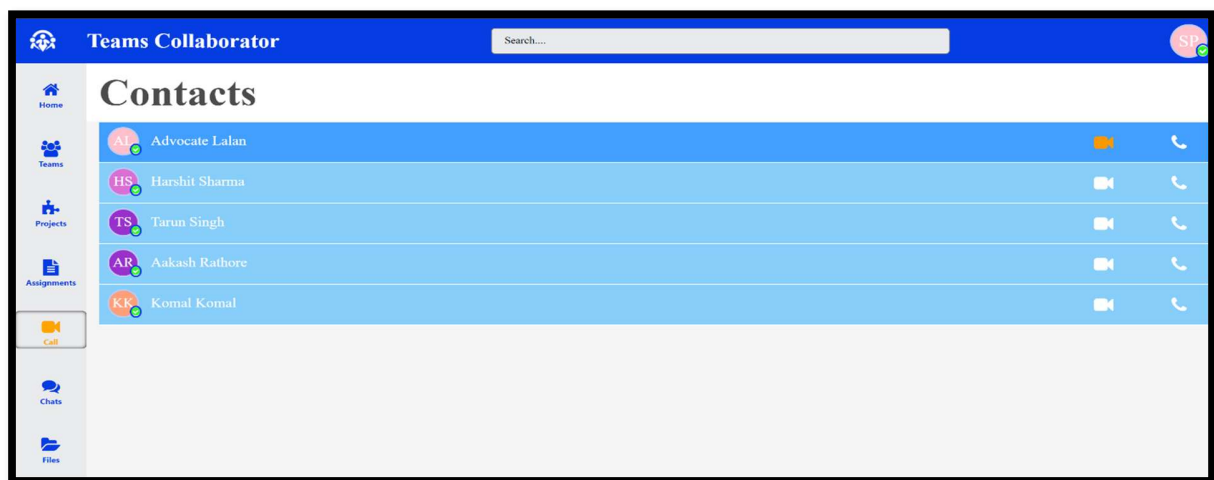


Fig.5.2.6.1 Image showing contact list in Call page

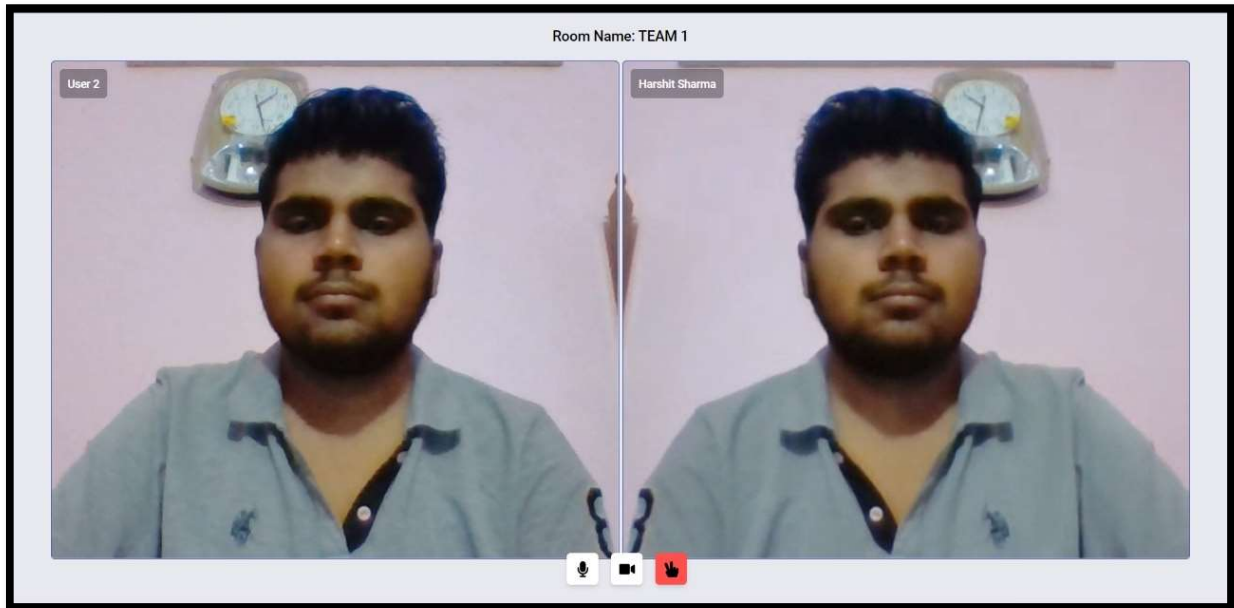


Fig.5.2.6.2 Image showing ongoing video call

Function code to display assignments page on selection of Assignments option from navigation bar:

@xframe_options_sameorigin

def callpage(request):

 users = userdetails.objects.exclude(username=request.user.username)

 return render(request, 'callpage.html' , {'users': users})

5.2.7 ‘Messaging’ Feature

- The messaging feature is available in Chat option of navigation bar.
- Once a user selects the Chat option from navigation bar the chat page having two sections is displayed.
- The first section of chat page is the contact list of the members and a search bar to search contact of a particular member available in the list.
- The second section of chat page is the section where chat messages corresponding to every contact member is shown.
- Before selecting any contact member, a load messages page is shown in second section chat page.

- When a user selects a contact from the contact list then the chat messages between that contact and the user is displayed.
- The chat messages page also has an input bar through which user can type their message and sent it to the other person.

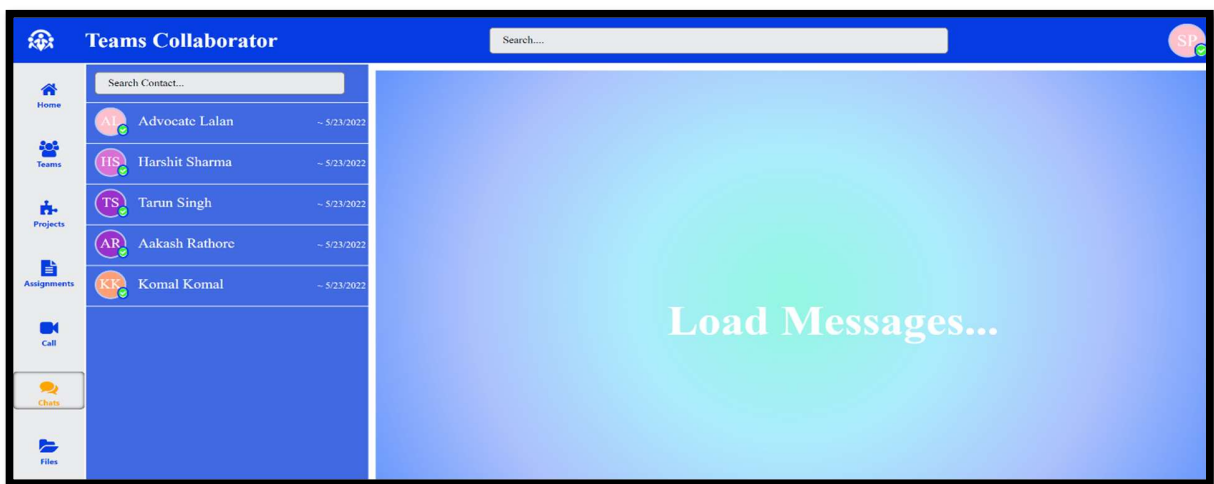
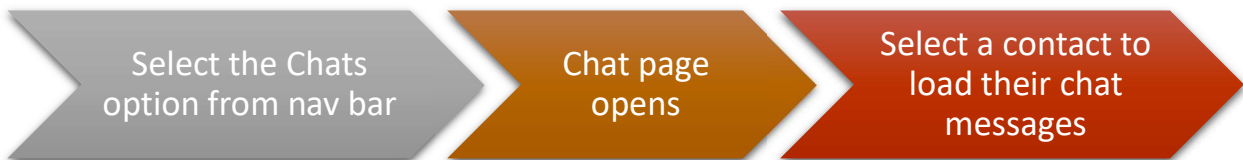


Fig.5.2.7.1 Image showing Chat page having load messages page.

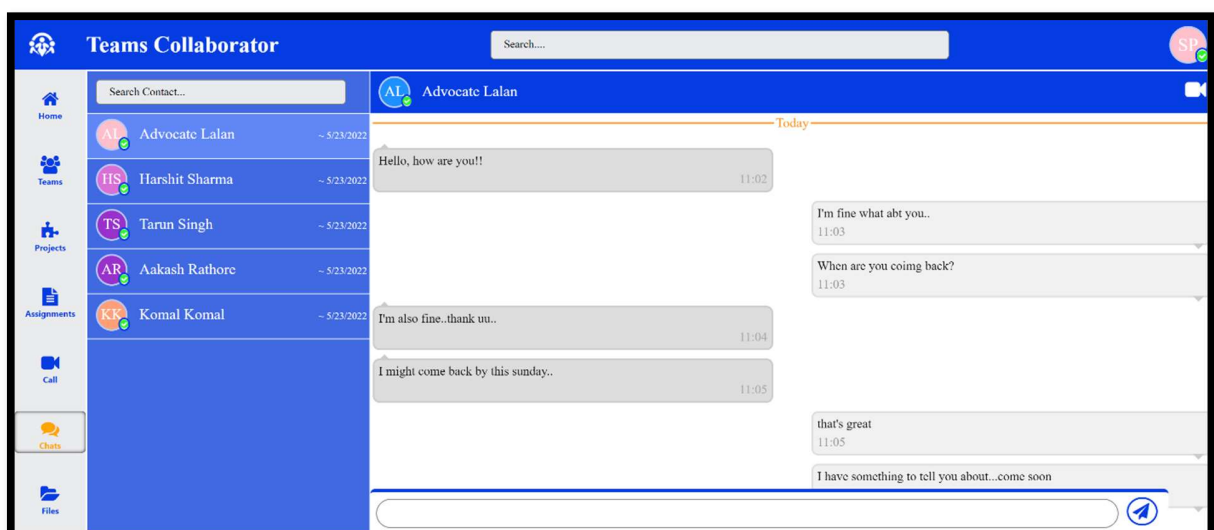


Fig.5.2.7.2 Image showing chat messages with a contact on chat page

Function code to display chat page on selection of Chats option from navigation bar:

```
@xframe_options_sameorigin
def chatpage(request):
    users = userdetails.objects.exclude(username=request.user.username)
    return render(request, 'chatpage.html', {"users": users})
```

Function code for load messages page:

```
@xframe_options_sameorigin
def loadchat(request):
    return render(request, 'load chat.html')
```

Function code to display chat messages on chat page:

```
@xframe_options_sameorigin
def userchat(request):
    userchatlist = chat.objects.all().order_by('time')
    return render(request, 'person1 chat.html', {"userchats": userchatlist })
```

5.2.8 'File Sharing' Feature

- Users can share documents at one place using collaborator application.
- Users can also upload their documents on application.
- Selecting the Files option from navigation bar displays the shared files page.
- Sharing of documents is necessary in a collaboration application so as all the documents will be stored at one place and will reduce the confusion among the versions of file.
- The files page contains a list of documents that are been shared by members.
- Any type of file such as zip file, image file, audio file, video file, .doc file, .pp. file, .xls file, adobe file, code file can be shared on the application.
- A user can also download the file by clicking the download option.
- To upload the files user has to click the upload file button given on the file page then user will be asked about the file to be shared and once he selects the file it will get uploaded on the application.
- The files shared by users are accessible by all users in a team.

- The details of shared file that are shown are their type, name, date of when uploaded and owner name who uploaded the document.



Function code to display shared documents page on selection of Files option from navigation

bar:

@xframe_options_sameorigin

def filepage(request):

files = file.objects.all()

return render(request, 'file sharing.html' , {"files": files})

Type	Name	Date	Uploaded By
	Folder 1	05/26/2022	Advocate Lalan
	Folder 2	05/12/2022	Tarun Singh
	Folder 3	05/32/2022	Komal Komal
	Folder 4	05/45/2022	Aakash Rathore
	Folder 4	05/45/2022	Aakash Rathore
	Folder 4	05/45/2022	Aakash Rathore

Fig.5.2.8.1 Image showing shared documents page containing list of documents

5.2.9 "User Authentication and Services" Feature

- Login Panel - The login panel or page will be used by user to login to Teams Collaborator application.

Input data validation will be done when user submits the details. If user enters wrong input details then error will be shown to him in red color text. Else he will be directed to the following page.

Validation of input data is necessary to prevent storage of incorrect details in database and to avoid unauthorized access to application.

There is also a forgot password option to be used by user when they forgot their password.

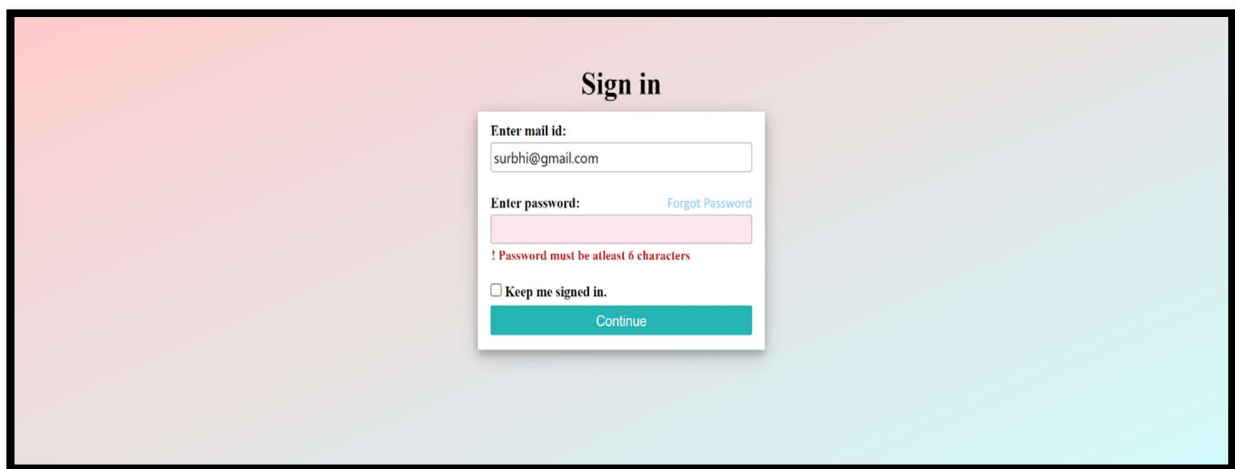


Fig.5.2.9.1 Image showing sign-in page

Function code for login:

```
def loginuser(request):  
    if request.method == 'POST':  
        email = request.POST.get('usermail')  
        password = request.POST.get('userpass')  
        user = authenticate(username=email, password=password)  
        if user is not None:  
            login(request, user)  
            return redirect('/')  
    return render(request, 'signin page.html')
```

Function Code for logout:

```
def logoutuser(request):  
    logout(request)  
    return redirect("/")
```

- **Forgot Password** - If user forgot their password, they can change it by clicking the forgot password option available in sign in page.

A password assistance page will be loaded. User has to enter the email address associated with their account. An OTP will be sent to user email address.

After entering the OTP and selecting continue a new page will open where they can create new password. This new password now will be stored and customer will be required to sign-in using the same password.

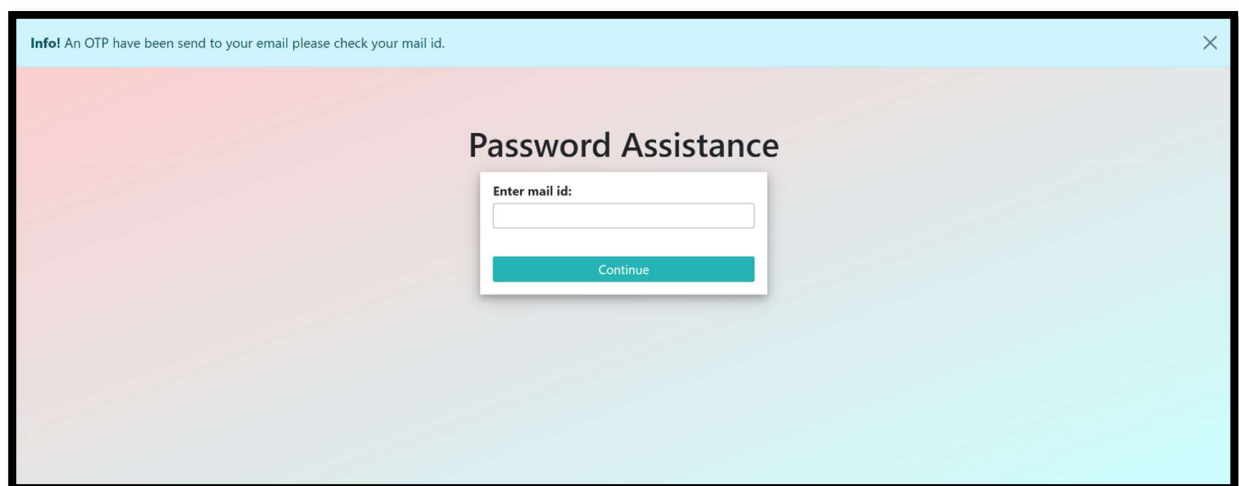


Fig.5.2.9.2 Image showing password assistance page

- **Creating New User** – User will register by signing up to the application. This page will only be available after successfully doing the payment for subscription.

Input data validation will be done when user submits the details. If user enters wrong input details then error will be shown to him in red color text. Else he will be directed to the following page.

Validation of input data is necessary to prevent storage of incorrect details in database and to avoid unauthorized access to application.



Sign up

Enter your username:

Enter mail id:

Enter password:

! Password must be atleast 6 characters

Confirm password:

! Enter the same password as above

Fig.5.2.9.3 Image showing Sign up page

CHAPTER – 6

SECURITY AND TESTING OF PROJECT

6.1 SECURITY OF PROJECT

Web application security is a central component of any web-based business. The global nature of the Internet exposes web properties to attack from different locations and various levels of scale and complexity. Web application security deals specifically with the security surrounding websites, web applications and web services such as APIs.

The level of security built into your project management software dictates how safe your project will be. This encompasses the integrity and confidentiality of your data, as well as the security of the infrastructure and the stability of your network. Too little security can open you up to hackers and scammers. But too much security can restrict your team from accessing the information they need.

- i. HTTPS – Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet.
 - In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS) or, formerly, Secure Sockets Layer (SSL). The protocol is therefore also referred to as HTTP over TLS, or HTTP over SSL.
 - The principal motivations for HTTPS are authentication of the accessed website, and protection of the privacy and integrity of the exchanged data while in transit.
 - It protects against man-in-the-middle attacks, and the bidirectional encryption of communications between a client and server protects the communications against eavesdropping and tampering.

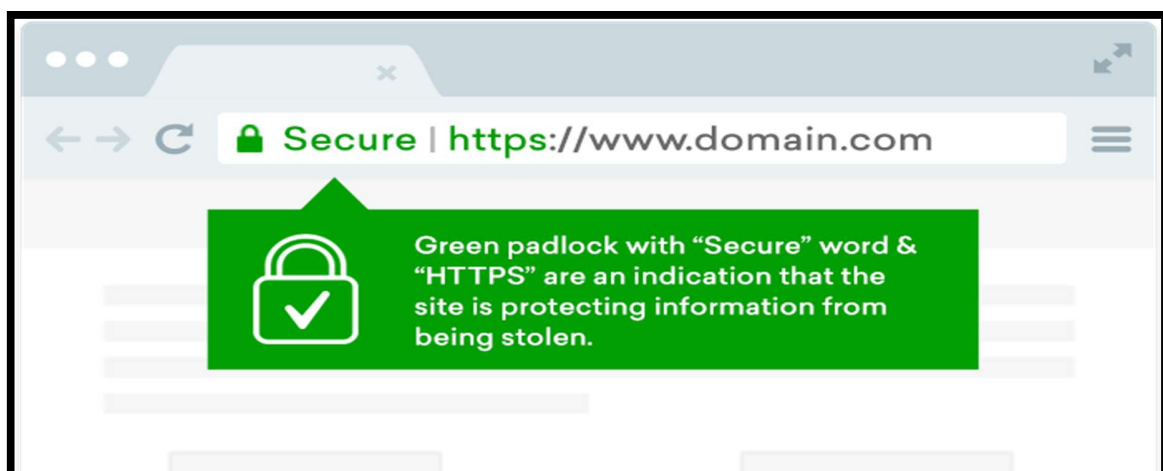


Fig.6.1.1 Image showing HTTPS security

- The authentication aspect of HTTPS requires a trusted third party to sign server-side digital certificates.
 - This was historically an expensive operation, which meant fully authenticated HTTPS connections were usually found only on secured payment transaction services and other secured corporate information systems on the World Wide Web. In 2016, a campaign by the Electronic Frontier Foundation with the support of web browser developers led to the protocol becoming more prevalent. HTTPS is now used more often by web users than the original non-secure HTTP, primarily to protect page authenticity on all types of websites; secure accounts; and to keep user communications, identity, and web browsing private.
 - The Uniform Resource Identifier (URI) scheme HTTPS has identical usage syntax to the HTTP scheme.
 - However, HTTPS signals the browser to use an added encryption layer of SSL/TLS to protect the traffic. SSL/TLS is especially suited for HTTP, since it can provide some protection even if only one side of the communication is authenticated. This is the case with HTTP transactions over the Internet, where typically only the server is authenticated (by the client examining the server's certificate).
 - HTTPS creates a secure channel over an insecure network. This ensures reasonable protection from eavesdroppers and man-in-the-middle attacks, provided that adequate cipher suites are used and that the server certificate is verified and trusted.
- ii. SSL Certificate - An SSL certificate is a digital certificate that authenticates a website's identity and enables an encrypted connection. SSL stands for Secure Sockets Layer, a security protocol that creates an encrypted link between a web server and a web browser.
- Companies and organizations need to add SSL certificates to their websites to secure online transactions and keep customer information private and secure.
 - SSL works by ensuring that any data transferred between users and websites, or between two systems, remains impossible to read.



Fig.6.1.2 Image showing SSL certificate logo

- It uses encryption algorithms to scramble data in transit, which prevents hackers from reading it as it is sent over the connection. This data includes potentially sensitive information such as names, addresses, credit card numbers, or other financial details.
- The process works like this:
 - A browser or server attempts to connect to a website (i.e., a web server) secured with SSL.
 - The browser or server requests that the web server identifies itself.
 - The web server sends the browser or server a copy of its SSL certificate in response.
 - The browser or server checks to see whether it trusts the SSL certificate.
 - If it does, it signals this to the webserver. The web server then returns a digitally signed acknowledgment to start an SSL encrypted session.

Encrypted data is shared between the browser or server and the webserver.

This process is sometimes referred to as an "SSL handshake." While it sounds like a lengthy process, it takes place in milliseconds.

- When a website is secured by an SSL certificate, the acronym HTTPS (which stands for HyperText Transfer Protocol Secure) appears in the URL. Without an SSL certificate, only the letters HTTP – i.e., without the S for Secure – will appear.
- A padlock icon will also display in the URL address bar. This signals trust and provides reassurance to those visiting the website.

- If a website is asking users to sign in, enter personal details such as their credit card numbers, or view confidential information such as health benefits or financial information, then it is essential to keep the data confidential.
 - SSL certificates help keep online interactions private and assure users that the website is authentic and safe to share private information with.
- iii. Symmetric encryption - It is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic data. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process.
- This encryption method differs from asymmetric encryption where a pair of keys - one public and one private - is used to encrypt and decrypt messages.
 - By using symmetric encryption algorithms, data is "scrambled" so that it can't be understood by anyone who does not possess the secret key to decrypt it. Once the intended recipient who possesses the key has the message, the algorithm reverses its action so that the message is returned to its original readable form.
 - The secret key that the sender and recipient both uses could be a specific password/code or it can be random string of letters or numbers that have been generated by a secure random number generator (RNG).
- iv. VP8 - VP8 is an open and royalty-free video compression format released by On2 Technologies in 2008. Initially released as a proprietary successor to On2's previous VP7 format, VP8 was released as an open and royalty-free format in May 2010 after Google acquired On2 Technologies.
- Google provided an irrevocable patent promise on its patents for implementing the VP8 format, and released a specification of the format under the Creative Commons Attribution 3.0 license. That same year, Google also released libvpx, the reference implementation of VP8, under the revised BSD license.
 - Opera, Firefox, Chrome, and Chromium support playing VP8 video in HTML5 video tag. Internet Explorer officially supports VP8 with a separate codec. According to Google, VP8 is mainly used in connection with WebRTC and as a format for short looped animations, as a replacement for the Graphics Interchange Format (GIF).

- VP8 can be multiplexed into the Matroska-based container format WebM along with Vorbis and Opus audio. The image format WebP is based on VP8's intra-frame coding. VP8's direct successor, VP9, and the emerging royalty-free internet video format AV1 from the Alliance for Open Media (AOMedia) are based on VP8.

6.2 TESTING OF PROJECT

Testing is a vast domain. When a software application is developed, it goes through various stages of testing. The tests may vary from being functional to non-functional, subject to test requirements. There are two ways for QAs(quality assurancer) to carry out tests:

- By manually executing test cases as mentioned in the written test plan
- By automating test scenarios using frameworks like Selenium WebDriver

In a tech-advanced world where automation tools and frameworks are making life easier for test engineers, it would be normal to assume that manual testing is obsolete.

6.2.1 Manual Testing

Manual testing, as the term suggests, refers to a test process in which a QA manually tests the software application in order to identify bugs. To do so, QAs follow a written test plan that describes a set of unique test scenarios. The QA is required to analyze the performance of the web or mobile application from an end user's perspective.

QAs verify the actual behavior of software against expected behavior, and any difference is reported as a bug.

Manual testing is an essential part of any test strategy as it helps QAs gain deeper insight from an end user's perspective. Since manual testing is carried out by a human without the intervention of test automation frameworks, it judges software from the most important metric: User Experience. Manual testing plays a vital role in exploratory testing or in test cases that are executed once or twice. This helps QAs to discover bugs in the early stages of the development cycle.

6.2.1.1 Stages of Manual Testing

- i. Unit Testing - Unit Testing involves verification of individual components or units of source code. A unit can be referred to as the smallest testable part of any software. It focuses on testing the functionality of individual components within the application.

- It is often used by developers to discover bugs in the early stages of the development cycle. A unit test case would be as fundamental as clicking a button on a web page and verifying whether it performs the desired operation.
- ii. Integration Testing - Integration testing is the next step after unit testing. Multiple units are integrated to be tested as a whole. For example, testing a series of webpages in a particular order to verify interoperability.
 - This approach helps QAs evaluate how several components of the application work together to provide the desired result. Performing integration testing in parallel with development allows developers to detect and locate bugs faster.
- iii. System Testing - As the name suggests, system testing involves testing all the integrated modules of the software as a whole. It helps QAs verify whether the system meets the desired requirements.
 - It includes multiple tests like validating output based on specific input, testing user experience and more. Teams perform several types of system testing like regression testing, stress testing, functional testing and more, depending on their access to time and resources.
- iv. UI Testing - UI Testing, also known as GUI Testing checks and verifies various aspects of any software that a user would interact with when using it. This usually means testing visual elements to ensure that they are functioning according to requirements regarding functionality and performance.
 - UI Testing covers the gamut of visual indicators and graphic-based icons – toolbars, fonts, menus, text boxes, radio buttons, checkboxes, colors, and more. It ensures that UI functions are error-free and working exactly as they are supposed to. Along with testing UI elements, UI testing must consider the various browsers, browser versions, and devices. In other words, cross browser testing must be an essential part of any UI testing strategy.
- v. Acceptance Testing - The main goal of acceptance testing is to verify whether the system as a whole is fit for use in the real world. Acceptance testing is performed both internally and externally. Internal acceptance testing (also known as alpha testing) is performed by the members of the organization.

- External testing (also known as beta testing) is performed by a limited number of actual end-users. This approach helps teams evaluate how well the product satisfies the user's standards. It also identifies bugs in the last stage before releasing a product.

6.2.2 Other Types of Testing

- Functional testing - Checking functions by emulating business scenarios, based on functional requirements. Black-box testing is a common way to verify functions.
- Performance testing - Testing how the software performs under different workloads. Load testing, for example, is used to evaluate performance under real-life load conditions.
- Regression testing - Checking whether new features break or degrade functionality. Sanity testing can be used to verify menus, functions and commands at the surface level, when there is no time for a full regression test.
- Stress testing - Testing how much strain the system can take before it fails. Considered to be a type of non-functional testing.
- Usability testing - Validating how well a customer can use a system or web application to complete a task.

CONCLUSION

The purpose of a collaboration tool is to support a group of two or more individuals to accomplish a common goal or objective. Collaborating online is used to remove the gap between employees due to far-away locations, run effective meetings online and ensure that they work together at the same time on the same page. And this helps to run projects faster and an improve communication between employees. Our web application Teams Collaborator does provide all the basic collaboration features one expects to be in. There were a few points which we will try to accomplish in its next version such as full-fledged security using SSL, toggling between light and dark mode, text to speech and a little more.

REFERENCES

- [1] Lomas, Cyprien & Burke, Michael & Page, Carie. (2008). Collaboration Tools.
- [2] Pappas, Menelaos & Karabatsou, Vassiliki & Mavrikios, Dimitris & Chryssolouris, George. (2006). Development of a web-based collaboration platform for manufacturing product and process design evaluation using virtual reality techniques. Int. J. Computer Integrated Manufacturing. 19. 805-814. 10.1080/09511920600690426.
- [3] Cemil UZUN, Contributions of Web Based Collaborative Tools and Platforms on Organizational Performance
- [4] Cihan Çilgin and Vahap Tecim, (November 2018), Designing Online Collaboration Tool for Managing Postgraduate Education with New Web Technologies
- [5] Saad, Razzaq. (2011). Challenges and Issues in Collaborative Software Developments.
- [6] Software Engineering, (2005), K.K. Aggarwal
- [7] <https://docs.djangoproject.com/en/3.2/>
- [8] <https://docs.python.org/3/>
- [9] What is an SSL Certificate? <https://www.kaspersky.com/resource-center/definitions/what-is-a-ssl-certificate>
- [10] ER Diagram Representation, https://www.tutorialspoint.com/dbms/er_diagram_representation.htm
- [11] Web Collaboration Guide, <https://kissflow.com/digital-workplace/collaboration/web-collaboration-guide/>
- [12] Software Analysis Design Tools, https://www.tutorialspoint.com/software_engineering/software_analysis_design_tools.htm
- [13] How to Write an Introduction for A Project, <https://www.javatpoint.com/how-to-write-an-introduction-for-a-project>
- [14] Final Project Report Format for Electronics Engineering Students, <https://www.elprocus.com/final-project-report-format-for-electronics-engineering-students/>
- [15] Online Collaboration Advantages and Challenges, <https://kissflow.com/digital-workplace/collaboration/online-collaboration-advantages-and-challenges/>

- [16] Best Online Collaboration Tools, <https://www.proofhub.com/articles/best-online-collaboration-tools>
- [17] The Best Online Collaboration Software, <https://www.pcmag.com/picks/the-best-online-collaboration-software>
- [18] Manual Testing for Beginners, <https://www.browserstack.com/guide/manual-testing-tutorial>