

**UNIVERSITY INSTITUTE OF TECHNOLOGY
PIRAPPANCODE, THIRUVANANTHAPURAM**

PROJECT REPORT

CONVO WORLD



MOHAMMED ASHIK M

(32021956024)

Submitted in partial fulfillment of the requirements for the award of
Bsc(Computer Science) degree of University of Kerala 2023

CERTIFICATE

Certified that this report titled "CONVO WORLD" is a bonafide record of the project work done by **MOHAMMED ASHIK M** under our supervision and guidance, towards partial fulfillment of the requirements for the award of the Degree of BSc (Computer Science) of the University of Kerala.

Internal Guide

Mrs. Fathima M S

(Lecturer in Computer Science)

Dr.Sreedevi S

Principal

External Examiner:

Date:

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this along with the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank **Dr. Sreedevi S** Principal, University Institute of Technology for providing me an opportunity to do the project work in Computer Science and giving us all support and guidance, which made us complete the project duly. We are extremely thankful to her for providing such nice support and guidance.

We owe our deep gratitude to our project guide **Mrs. Fathima M S**, who took keen interest in our project work and guided us all along till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all teaching staff of the Computer Science Department. Our sincere thanks and apologies to anyone who deserves the credit but whose names fail to appear in the list above.

Sincerely,

MOHAMMED ASHIK M

DECLARATION

We hereby declare that the work presented in the project **CONVO WORLD** on the original work done by us under the guidance of **Mrs. Fathima M S** (Lecturer of Department, Computer Science). The same work has not been submitted elsewhere for another degree for the best of our knowledge.

MOHAMMED ASHIK M

Place: Pirappancode

Date:

CONTENT

1.INTRODUCTION

2.SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

2.2 PROPOSED SYSTEM

2.3 FEASIBILITY STUDY

3.SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENT

3.2 SOFTWARE REQUIREMENT

3.3 USED LANGUAGES

4.SYSTEM DESIGN

4.1 DATA FLOW DIAGRAM

4.2 USE CASE DIAGRAM

5.SOFTWARE DESIGN

6.CODE

7.SYSTEM TESTING

7.1 LEVELS OF TESTING

8.SYSTEM IMPLEMENTATION

9.MAINTENANCE

10.FUTURE ENHANCEMENT

11.CONCLUSION

12.APPENDIX

12.1 SCREENSHOTS

12.2 GANTT CHART

1. INTRODUCTION

The workbench is the website that provide a free chatting box for users. This chatting page can handle text message ,images,video files,audio files,etc. This website mainly focused on study purpose. By using this website users in any where in the world they can communicate easily. This service is totally online.

This is a website not a application so it dont need any storage spaces. This is only works on computers or desctops. Now it is not accessible for mobile phones. For using this website user want to create an account. After creating account a user can open a channel . This channel is work as a chating space .

Other users in any where of the world can join this channel using channel name and password. The users can follow this channel for future use. The special feature of this website is the channel owner can conduct a live exam or test paper on his channel. this feature is helpful for teachers.

With its user-friendly interface and extensive range of controls a user can easily manage this website. They can share study materials and other documents. this web site is very secure. This website is easily handled by anyone who can have basic computer knowledge.

We made this website for mainly students and teachers . This website is very helpful for students.

1. Home page: It is the main page of the website in which it shows the website details and Form of entering a channel.

2. Login Page: Login page will direct the user to enter this user email and password to enter to his home page. If the credentials belong to the admin, he would be redirected to admin page .

3. Create Channel page: There we can create new channels.

4. Account page: It will show user account details, user channel and channel followed by user. Also users can edit his account.

5. Main channel page: This is the main user working page of the website. This web page is divided into two parts. In the first part, we can share our information like text, document, video and image. Second part, admin can add any futures like exam, stream, live, poll etc.

- 6. Examination page:** Here users take the test by answering each question.
- 7. View answer page:** Admin will check users answer and here user mark list table will be shown.
- 8. Image view page:** Used for preview of images uploaded to chat page
- 9. Video view page:** Used for preview of videos uploaded to chat page.
- 10. PDF view page:** Used for preview of pdf uploaded to chat page.

2. SYSTEM ANALYSIS

A system is simply a set of components to accomplish an objective. Developing a new system, investigating into the operation and making possible changes in the existing system are called System Analysis. Analysis comprises a detailed study of the various operations performed by a system and their relationships within and outside the system. It is the process of gathering and interpreting facts, diagnosing problems and improving the system using the information obtained.

The objectives of System Analysis include the following are;

- Identifying the user's needs.
- Performing economic and technical analysis.
- Establishing cost and scheduled constraint.

Here the system analyst should study a system with an eye on solving its problem using computers. It is the essential part for the development of a project by a system analyst.

System analysis is for finding out what happens in the existing systems, deciding on what changes and new features are required and defining exactly what the proposed system must be. This process of system analysis is largely concerned with determining, developing and agreeing to the user's requirements. It provides prime opportunities to communicate well with the user and conceived a joint understanding of what a system should be doing, together with a view of the relative importance of the system facilities using interactive techniques. To analyze a system one has to study the system's work in detail. The system analyst must understand the functioning and concept of the system in detail, before designing to the appropriate computer-based system that will meet all the requirements of the existing system.

2.1 EXISTING SYSTEM

Currently there is no website that provides any document sharing and exam facilities. Current portals belong to companies that show their ads and collect user information to sell data and their services without any privacy. This website doesn't need any personal information. This is a totally free website.

2.2 PROPOSED SYSTEM

In the proposed system we prepare a website that provide a free chatting box for users. This chatting page can handle text message ,images,video files,audio files,etc. This website mainly focused on study purpose. For using this website user want to create an account. After creating account a user can open a channel . This channel is work as a chating space .

Other users in any where of the world can join this channel using channel name and password. The users can follow this channel for future use. The special feature of this website is the channel owner can conduct a live exam or test paper on his channel.

2.3 FEASIBILITY STUDY

In Software Engineering is a study to evaluate feasibility of a proposed project or system. Feasibility study is one of the important four stages of the Software Project Management Process. As the name suggests, a feasibility study is the feasibility analysis or it is a measure of the software product in terms of how beneficial product development will be for the organization from a practical point of view. Feasibility study is carried out based on many purposes to analyze whether a software product will be right in terms of development, implantation, contribution of project to the organization etc.

The feasibility study report must address five levels of feasibility. They are:

- Technical feasibility.
- Operational feasibility.
- Economical Feasibility.
- Legal Feasibility.
- Schedule Feasibility.

a. Technical Feasibility

In Technical Feasibility current resources both hardware software along with required technology are analyzed/assessed to develop projects. This technical feasibility study reports whether there exists correct required resources and technologies which will be used for project development. Along with this, the feasibility study also analyzes technical skills and capabilities of the

technical team, whether existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc.

b. Operational Feasibility

In Operational Feasibility the degree of providing service to requirements is analyzed along with how easy the product will be to operate and maintain after deployment. Along with this other operational scopes are determining usability of product, Determining suggested solution by software development team is acceptable or not etc.

c. Economic Feasibility

In the Economic Feasibility study, the cost and benefit of the project is analyzed. Means under this feasibility study a detailed analysis is carried out of what will be the cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on. After that it is analyzed whether the project will be beneficial in terms of finance for the organization or not.

d. Legal Feasibility

The Legal Feasibility study project is analyzed from a legality point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. Overall it can be said that a Legal Feasibility Study is a study to know if proposed projects conform to legal and ethical requirements.

e. Schedule Feasibility

In Schedule Feasibility Study mainly timelines/deadlines are analyzed for proposed projects which includes how many times teams will take to complete the final project which has a great impact on the organization as the purpose of the project may fail if it can't be completed on time.

3. SYSTEM REQUIREMENTS

Software requirements specification is one of the most important phases in developing the software. We need to know all the requirements and other valuable information before the process of designing a system. If the requirements are specified with the best of programmers knowledge, it is always good for the systems because the further modifications need it. The system specification describes the hardware and software specification used to develop the software.

3.1 SOFTWARE REQUIREMENTS

1. Operating System: Windows
2. Language: Node js
3. Web Technologies: Html 5, JavaScript, CSS
4. Server Environment:
5. Web Server:
6. Database: Mongodb
7. IDE: visual studio cod

3.2 HARDWARE REQUIREMENTS

1. Processor : Intel Pentium III or above
2. Processor Speed : 1GHz & above
3. RAM : 1024 MB & above
4. Hard Disk Capacity : 20GB & above
5. Monitor : SVGA Color Monitor
6. Keyboard : Multimedia Keyboard

3.3 USED LANGUAGES

Overview of JAVA SCRIPT

JavaScript is the first internet scripting language . Netscape, the creator of the most popular Internet Browser, developed it. As the name suggests , it is a scripting language only and not a programming language. It interprets the Java Script commands embedded in the HTML document. It has an excellent collection of event handling routines to make the web pages interactive and responsive to the user actions. It is a case-sensitive language. It is very easy to learn and use. Variable data type needs not be declared in JavaScript. It is not fully extensible.

JavaScript, in its purest form is an object based, cross platform, loosely typed, multi use language that allows a programmer to deploy many types of solutions to many clients. It also allows server side processing for Netscape & Microsoft Web servers. JavaScript has also recently been included in Microsoft's Windows scripting host to allow programmers to write scripts to be executed on the operating system itself. This functionality is similar to be executed on the operating system functionality and versatility in what they can accomplish. In addition to the benefits of these environments where Java Scripts can be executed, though it is still young in terms of age, Java script in its current version is very mature and powerful. Its functionality, ability and versatility that positions Java Script as the solution for programmers. Client-side Java Script is several core objects that are created when a page is loaded in the browser. There are also derived objects that are created when certain tags are included on a page. These derived objects have some of the various characteristics of their parent object and also allow scripting access to the HTML tags properties.

Overview of CSS

Cascading Style Sheets (CSS) are the modern standard for website presentation. When combined with structural markup language like HTML, XHTML, or XML (though not limited to these), CSS provides Internet browsers with the information that enables them to present all the visual aspects and elements of a web document. CSS apply things like borders, spacing between paragraphs, margins, headings on images, control of font faces or colors, background colors and images, textual effects like underlined or strike-through text, layering, positioning, and a number of other presentational effects. CSS controls the presentational aspects of a web page's design, whereas HTML, XHTML, or XML control the structure of a webpage, which means more than determining that certain text is a heading, other text is a paragraph, other text as a list of hyperlinks, and so on.

By using modern standards like CSS and XHTML, you can dramatically reduce the cost of building and maintaining a website when compared to legacy HTML-only pages. You can also greatly reduce the amount of physical bandwidth and hard disk space required, resulting in immediate long-term benefits for any webmasters and web visitor.

Overview of HTML

HTML, which stands for **HyperText Markup Language**, is the predominant markup language for web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items. It allows images and objects to be embedded and can be used to create interactive forms. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content. It can include or can load scripts in languages such as JavaScript which affect the behavior of HTML processors like Web browsers; and Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. HTML documents are composed entirely of **HTML elements** that, in their most general form have three components: a pair of element **tags** with a "start tag" and "end tag"; some element attributes given to the element within the tags; and finally, all the actual, textual and graphical, information *content* that will be rendered on the display.

HTML defines several data types for element content, such as script data and style sheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

Overview of Node JS

Node.js is an open-source, server-side JavaScript runtime environment. It's designed to build scalable network applications and is commonly used for web development. Node.js uses an event-driven, non-blocking I/O model, making it efficient and suitable for real-time applications. You can use npm (Node Package Manager) to easily manage and install libraries and packages. Popular frameworks like Express.js are often used with Node.js for building web applications. Node.js is known for its speed and flexibility, making it a popular choice among developers.

Overview of MongoDB

MongoDB is a popular NoSQL database known for its flexibility, scalability, and ease of use. It stores data in a document-oriented format, using BSON (Binary JSON), which allows it to handle data with varying structures. MongoDB stores data in collections, similar to tables in relational databases, but without a fixed schema. This schemaless nature is advantageous for applications where data structures evolve.

MongoDB offers a powerful query language for data retrieval and filtering, supporting various querying options. It's designed for horizontal scalability, making it suitable for applications with growing data needs. MongoDB ensures data availability through replication, maintaining multiple copies of data.

The database includes an aggregation framework for complex data transformations and calculations, supports indexes for query optimization, and handles geospatial data for location-based applications. Its active developer community and ecosystem provide extensive support.

MongoDB finds applications in content management systems, e-commerce platforms, mobile app backends, real-time analytics, and more. Security features, such as authentication, authorization, and encryption, help protect data. In summary, MongoDB is a versatile, schema-flexible, and scalable database choice for modern data-intensive applications.

4. SYSTEM DESIGN

System Design develops the architectural details required to build a system or product. The system design process encompasses the following activities:

1. Partition the analysis model into the subsystems.
2. Identity concurrency that is dedicated to the problem.
3. Develop design for the user interface.
4. Choose a basic strategy or implement data management.
5. Design an appropriate control mechanism for the system, including task management

System design provides understanding of the procedural details, necessary for implementing the system recommended in the feasibility study. Basically it is all about the creation of a new system. This critical phase since it decides the quality of the system has a major impact on the testing and implementation phases.

1. The allocation of the equipment and the software to be used.
2. The identification of the user requirement for the system.
3. Drawing of the expanded system flowcharts to identify all the processing functions required

System design is the most creative and challenging phase of the system life cycle. The term design describes the final system and the process by which it is to be developed. During the system design phase the designers must design how to produce an efficient and effective system. There are two levels of system design, logical design and physical design.

In the logical design the designers produce a specification of the major features of the system which meets the objectives. The delivered product of logical design includes current requirements of the following system components:

1. Input design
2. Output design

Design objectives and principles

Input design is a part of the overall system design, which requires very careful attention. If data going into the system is incorrect, then processing and output will magnify these errors. Thus the designer has a number of clear objectives in the different stages of input design.

1. To produce a cost effective method of input.
2. To achieve the highest possible level of accuracy.

Module description

1. Admin
2. Public/User

➤ Admin

- Manage all contents
- View all messages

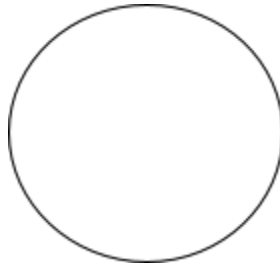
➤ Public/User

- View all pages in site
- Sent messages and files

4.1 DATA FLOW DIAGRAM

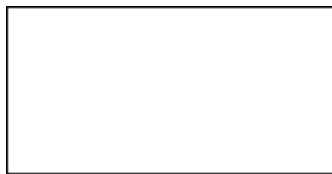
A DFD is a network that describes the flow of data throughout a system, data stores, and the process that changes or transforms data flows. Data Flow Diagrams are also known as Data Flow Graphs. DFDs are commonly used during the problem analysis stage. They are useful in understanding a system and can be effectively used for partitioning during analysis. The DFD network is a formal, logical abstract of a system that may have many possible physical configurations. This reason a set of symbols that do not imply a physical form are used to represent data source, data flows, data transformations and data storage. The basic element of DFD are:

- Process: A process that represents some amount of work being performed on data.



Circle or Bubble

- External Entity: This represents any outside agency, which interacts with the system. It represents the source or destination of data for the system under consideration.



Rectangle

- Data Flow: The data flow portrays an interface among different components in a DFD. It represents flow of data between a process and an external entity or between a process and data store.



Arrow

- Data Stores: A data store is a place for holding information within the system.

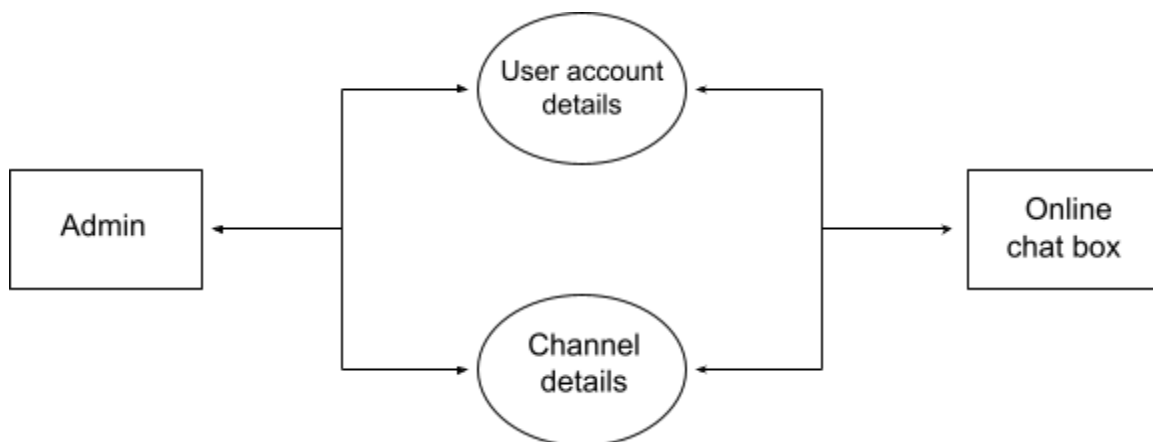


One-end opened rectangle

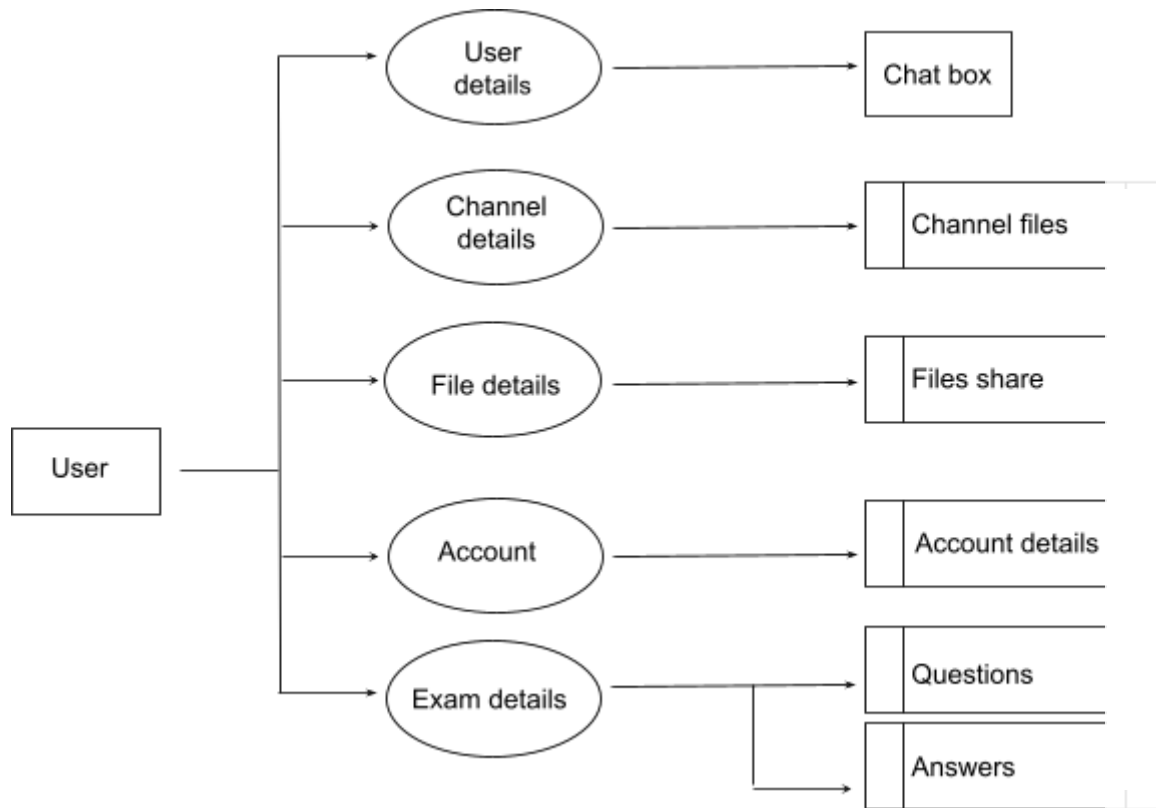
Level 0



Level 1



Level 2



4.2 USE CASE DIAGRAM

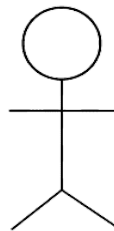
A Use Case Diagram displays the relationship among actors and Use Cases. Use Case Diagrams are drawn to capture the functional requirements of a system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements.

- To identify functions and how roles interact with them.
- For a high-level view of the system.
- To identify internal and external actors

The two main components of Use Case Diagrams are actors and cases.

Actor

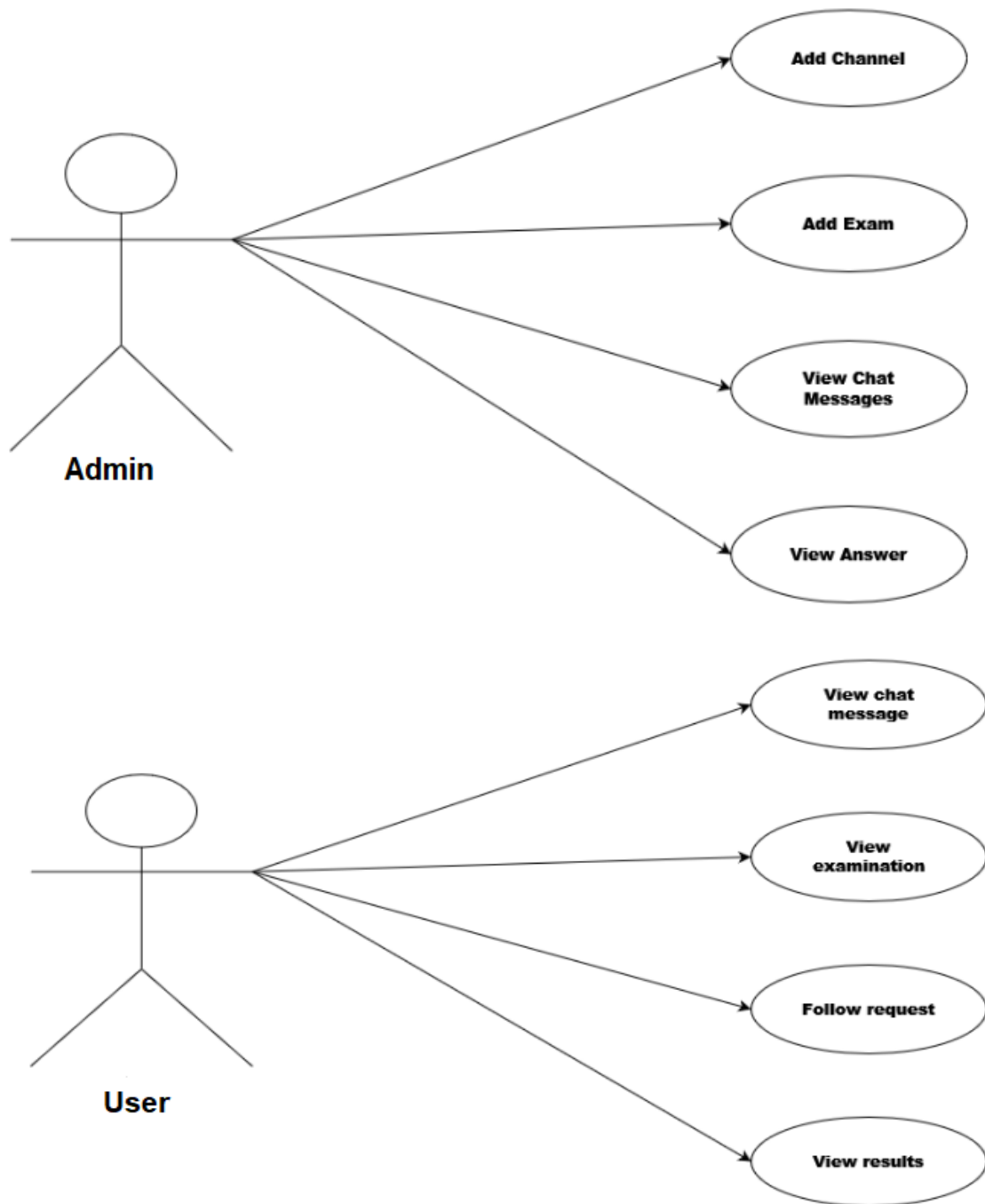
Actor in a Use Case Diagram is any entity that performs a role in one given system. This could be a person, organization or an external system and usually drawn like skeleton shown below:



Use case

Use case A Use case represents a function or an action within the system. It is drawn as an oval and named with the function. Symbol of Use case is shown below:





5. SOFTWARE DESIGN

System Design develops the architectural details required to build a system or product. The system design process encompasses the following activities:

- Partition the analysis model into subsystems.
- Identify concurrency that is dictated by the problem.
- Develop design for the user interface.
- Choose a basic strategy or implement data management.
- Design an appropriate control mechanism for the system, including task management

System design provides an understanding of the procedural details, necessary for implementing the system recommended in the feasibility study. Basically it is all about the creation of a new system. This is a critical phase since it decides the quality of the system has a major impact on the testing and implementation phases.

- The allocation of the equipment and the software to be used.
- The identification of the user requirement for the system.
- Drawing of the expanded system flow charts to identify all the processing functions required.

System design provides an understanding of the procedural details, necessary for implementing the system recommended in the feasibility study. Basically it is all about the creation of a new system. This is a critical phase since it decides the quality of the system and has a major impact on the testing and implementation phases. System design is the most creative and challenging phase of the system life cycle. The term design describes the final system and the process by which it is to be developed. During the system design phase the designers must design how to produce an efficient and effective system. There are two levels of system design, Logical design and Physical design. In the logical design, the designer produces a specification of the major features of the system which meets the objectives. The delivered product of logical design includes current requirements of the following system components:

- Input design.
- Output design
- Database design.

- Process design

Physical design takes this logical design blueprint and produces the program specification. Physical design and user interfaces for a selected hardware and software. Structured design is data flow based methodology that partitions a program into a hierarchy of modules organized top-down manner with details at the bottom. The value of using a top-down approach, starting at the general levels to gain an understanding of the system and gradually moving down to levels of greater detail, one data flow diagram becomes several at the next lower level. The top top-down method is also widely used in systems engineering and software design. Each function the system will perform is first and then developed in greater detail.

Design Objectives and Principles

The two operational design objectives continually sought by the developers are system reliability and maintainability. A system is said to be reliable if it does not produce dangerous or costly failures when it is used in a reasonable manner, in a manner that a typical user expects is normal. There are two levels of reliability. The first is that the system is meeting the right requirements. For instance, a system might be expected to have specific security features or controls built into it by users. The second level of system reliability involves the actual working of the system, reliability involves workings of the system delivered to the user. At this level, systems reliability is interwoven with software engineering and development. An error occurs whenever the system does not produce the expected speed. No program is ever neither fully debugged nor fully tested, nor proven correct. The correctness of the design of a system depends upon the level of the precision of the system being built which satisfies the requirement of the system. The aim of the design phase is to produce the best possible design within the limitations proposed by the requirements and the facilities. Some of the properties of the design are as follows. It points out how easily the correctness of the design can be argued. All design elements must be traceable to the requirements. All the required components of the design must be specified. There should not be inherent inconsistencies. An efficient system is one that consumes less processing time and requires less memory and so it must be satisfied. Simplicity and understandability are the most important quality criteria for the software system. It would be a great help for the analyst in future while repairing the system.

a. Input Design

Inaccurate input data is the most common cause of error in processing data. Errors entered by the data entry operators can be controlled by the input design. Determine what data to input & what medium to use, data arrangement and coding, guidelines provided for the users regarding input, data & transaction needing error validation and methods for performing input validation when an error occurs. The arrangement of messages and comments in online conversation as well as placement of data, headings and titles on display screens or source documents is also a part of input design. The design of input also includes specifying the means by which end user and system operators direct the system in which action to take. The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. So in input design the following things are considered.

- What data should be given as input?
- How should the data be arranged or coded?
- Methods for preparing input validations steps to follow when errors occur.
- Identify the inputs into the system.
- Determine the contents of their inputs.
- Choose an appropriate device to change the user's data into a form processed by the computer system.
- Complete the detailed design work involved in specifying forms, inputs, screen and any other data collection document.

b. Output Design

Output refers to the results and information that are generated by the system. Here determine information to be present, decide layout and select output medium, arrange presentation of information in accepted format and decide how to distribute output to intended recipients. Location characteristics and format of column headings and pagination are specified. Output design plays a major role in providing the user with required format. The major function of the output is to convey information and so its layout and design are careful considerations. Information must be carefully considered to the needs of the user. Standards for printed output

suggest giving each output a name or title, providing a sample of the output layout, and specifying the procedure for providing the accuracy of the output data. The output devices to consider depends on the compatibility of the devices with the system, response time requirement and printed quality required. The design output form, attention is given to proper identification and wording, readability and use, composition and layout, order of data items and clarity of instructions. A well designed form with clarity stated captions should be self-instructing. An organization's form must be centrally controlled for efficient handling. Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the systems relationship with the user and help in decision making. Since the reports are directly referred to by the management for taking decisions and to draw conclusions. They must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user. So while designing output the following things are to be considered.

- Determine what information to present.
- Arrange the presentation of information in an acceptable format.
- Decide how to distribute the output to intend receipts.
- Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy.

Efficient and intelligent output design should improve system relation with the user and help in decision making. That is, this makes the system user friendly to be displayed or printed as per the user's choice. A quality output is one which meets the requirements of the end user and which presents the information in a way which is clear, easy to read and visually attractive. In order to decide on an appropriate method of presentation, and a suitable format, a number of issues to be considered like who receives the output, under what circumstances the output is received etc. In this project the output will display the result based on the choice of selection made by the user.

c. Form Design

Form Design is an important feature of the input design phase, because using the forms as a means of entry enters almost all the information needed during the implementation of the project. This system provides many user-friendly forms that help the user to interact with the system easily. The input forms are designed in such a manner so as to avoid any sort of confusion and to guide the user in the correct track. A thorough study has been made on the type and how the input form is to be designed. Some inputs from the user may cause several errors and hence there are strict validation measures provided.

d. User Interface Design

Purpose of a USER INTERFACE DESIGN is to communicate effectively through form designs; there are several major requirements,

- The form title must clearly identify its purpose. Columns and rows should be labeled to avoid confusion. The form should be identified by firm name or code number to make it easy to reorder
- The form must be easy to use and fill out. It should be legible, intelligible and uncomplicated.
- The data requested should reflect a logical sequence.
- The form composition, color, layout and paper stock levels them to easy reading.
- The form should have field positions indicated under each column of data and should have some indication of where decimal points are.
- The form must be easily stored and filed. Important items must be in a prominent location on the form
- The form must be cost effective (i.e.) eliminating unnecessary data

e. Procedural design

This step involves specifications of how processing will be performed. In this, there are two aspects:

- Computer Procedure-The computer procedure will specify what functions will be carried out on the computer, what will be different programs and in what sequence the programs will be run.
- Non-computer procedure-The non-computer procedure will specify the manual procedures for feeding input data, receiving outputs etc.
- Control Design-The control design indicates necessary procedures which will ensure correctness of processing, accuracy of data, timely output etc. this will ensure that the system is functioning as per plan.

f. Development and Deployment Design

Major responsibilities include system Requirement Study, Preparing System Design Document, Preparing the Program (Form) Specs, Peer Review of the Specs, Implement Quality procedures

as per the QMS Document, Documentation of SRS Revalidation & Design Documents, thorough grasping of the user functionality and applying, approving the same in the project, Developing, Testing & Debugging of Forms, User Training and Implementation of the module.

g. Database Design

A database is a collection of interrelated data stored with minimum redundancy to serve users more quickly and efficiently. The general objective of a database is to make information access easy, quick, inexpensive, integrated and shared by different applications and users. Database design is an important yet sometimes overlooked part of the application development lifecycle. An accurate and up-to-date data model can serve as an important reference tool for Database Administrators, developers, and other members of joint application development teams. The process of creating a data model helps the team uncover additional questions to ask of end users. Effective database design also allows the team to develop applications that perform well from the beginning. By building quality into the project, the team reduces the overall time it takes to complete the project, which in turn reduces project development costs. The central theme behind database design is to "measure twice, cut once".

Effective database designers will keep in mind the principles of normalization while they design a database. Normalization is a database design approach that seeks the following four objectives:

- Minimization of data redundancy,
- Minimization of data restructuring,
- Minimization of I/O by reduction of transaction sizes, and
- Enforcement of referential integrity.

The concepts and techniques used when designing an effective database includes: An entity is a logical collection of things that are relevant to your database. The physical counterpart of an entity is a database table. An attribute is a descriptive or quantitative characteristic of an entity. The physical counterpart of an attribute is a database column or field.

A primary key is an attribute (or combination of attributes) that uniquely identifies each instance of an entity. A primary key cannot be null and the value assigned to a primary key should not change over time. A primary key also needs to be efficient. For example, a primary key that is associated with an INTEGER data type will be more efficient than one that is associated with a CHAR data type. Primary keys should also be non-intelligent; that is, their values should be assigned arbitrarily without any hidden meaning. Sometimes none of the attributes of an entity

are sufficient to meet the criteria of an effective primary key. In this case the database designer is best served by creating an "artificial" primary key.

A relationship is a logical link between two entities. A relationship represents a business rule and can be expressed as a verb phrase. Most relationships between entities are of the "onetomany" type in which one instance of the parent entity relates to many instances of the child entity. The second type of relationship is the "many-to-many" relationship. In a "manytomany" relationship, many instances of one entity relate to many instances of the other entity.

"Many-to-many" relationships need to be resolved in order to avoid data redundancy. "Many-to-many" relationships may be resolved by creating an intermediate entity known as a cross reference (or XREF) entity. The XREF entity is made up of the primary keys from both of the two original entities. Both of the two original entities become parent entities of the XREF entity. Thus, the "many-to-many" relationship becomes resolved as two "one-to-many" relationships. A "foreign key" exists when the primary key of a parent entity exists in a child entity. A foreign key requires that values must be present in the parent entity before values may be inserted in the child entity. The concept of maintaining foreign keys is known as "referential integrity".

Relationships between two entities may be classified as being either "identifying" or "non identifying". Identifying relationships exist when the primary key of the parent entity is included in the primary key of the child entity. On the other hand, a non-identifying relationship exists when the primary key of the parent entity is included in the child entity but not as part of the child entity's primary key. In addition, non-identifying relationships may be further classified as being either "mandatory" or "non-mandatory".

A mandatory non identifying relationship exists when the value in the child table cannot be null. On the other hand, a non-mandatory non-identifying relationship exists when the value in the child table can be null.

In conclusion, effective database design can help the development team reduce overall development time and costs. Undertaking the process of database design and creating a data

model helps the team better understand the user's requirements and thus enables them to build a system that is more reflective of the user's requirements and business rules. A primary key is an attribute (or combination of attributes) that uniquely identifies each instance of an entity. A primary key cannot be null and the value assigned to a primary key should not change over time. A primary key also needs to be efficient. For example, a primary key that is associated with an INTEGER data type will be more efficient than one that is associated with a CHAR data type. Primary keys should also be non-intelligent; that is, their values should be assigned arbitrarily without any hidden meaning. Sometimes none of the attributes of an entity are sufficient to meet the criteria of an effective primary key. In this case the database designer is best served by creating an "artificial" primary key.

A relationship is a logical link between two entities. A relationship represents a business rule and can be expressed as a verb phrase. Most relationships between entities are of the "onetomany" type in which one instance of the parent entity relates to many instances of the child entity.

In conclusion, effective database design can help the development team reduce overall development time and costs. Undertaking the process of database design and creating a data model helps the team better understand the user's requirements and thus enables them to build a system that is more reflective of the user's requirements and business rules.

6.CODING

Login.hbs

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="../stylesheets/login.css">
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body >
  <div class="main">
    <div class="container1">
      <div class="backDiv">
        {{#if data}}
        <div class="empty" id="container4">
          {{else}}

<div class="container4" id="container4">
  {{/if}}
  <h2>Login Account</h2>
  <form action="/login" method="post">
    {{#if data2}}
    <div class="container3" >
    {{else}}
    <div class="container2" >
    {{/if}}
    <input type="text" placeholder="Enter Email" name="email" required><br>
    <input type="password" placeholder="Enter Password" name="password" required><br>
  </div>
  <div class="singup-part">
    <div onclick="Singup()">Create new account ?</div>
  </div>
  <div class="loginpart">
```

```

<button type="submit">Login</button>
</div>
</form>
</div>
    {{#if data}}
    <div class="" id="signupdiv">
    {{else}}
    <div class="empty" id="signupdiv">
    {{/if}}
</div>
<h2>SignUp</h2>
<form action="/signup" method="post">
    <div class="input-group">
        <input type="text" id="username" name="username" placeholder="Username" required>
    </div>
    <div class="input-group">
        <input type="email" id="email" name="email" placeholder="Email" required>
    </div>
    <div class="input-group">
        {{#if data}}
        <p>! this already existed</p>
        {{else}}
        {{/if}}
        <input type="password" id="password" name="password" placeholder="Password"
required>
    </div>
    <div class="button">
        <div class="singup-part">
            <a class="signup" href="/login">Already have account ?</a>
        </div>
        <div class="loginpart">
            <button type="submit">Sign Up</button>
        </div>
    </div>
</form>
</div>
</div>
    {{!-- back div end --}}
<div class="frontDiv">
    <div class="sepfond">
        <P>We're thrilled to see you again.</P>

```



```

    </div>
    <h1>WELCOME BACK</h1>
    <p>We're thrilled to see you again.Please enter your credentials to access your
account</p>
    <p>Don't have an account yet?</p>
    <button>Signup</button>
  </div>

</div>
{ {!-- <div class="last-Name"><samp>designed by</samp></div> --} }
</div>
<script>
function Singup(){
console.log("click")
const Loginpart=document.getElementById("container4")
const SignUppart=document.getElementById("signupdiv")
Loginpart.classList.add("empty")
SignUppart.classList.remove("empty")
}
</script>
</body>
</html>

```

Newchannel.hbs

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="/stylesheets/newclass.css">
<title>Document</title>
</head>
<body>
<h1>Create New Class</h1>
<form action="/newclass" method="POST">

```

```

<label for="className">Class Name:</label>
<input type="text" id="className" name="className" required>
<div class="maindiv">
<label for="className">Class Id:</label>
{{#if classexist}}
<div >ClassId Exist</div>
{{else}}
{{/if}}
</div>
<input type="text" id="classId" name="classId" required><input type="button"
onclick="random()" class="random" value="random">
<label for="Email">Email:</label>
<input type="text" id="username" name="email" required>
<label for="password">Password:</label>
<input type="password" id="password" name="password" required>
{{#if userdata}}
<button type="submit" class="last-submit">Create Class</button>
{{else}}
<a href="/login">please login</a>
{{/if}}
</form>
<script>
function getRandomAlphabet() {
const alphabets = "abcdefghijklmnopqrstuvwxyz";
const randomIndex = Math.floor(Math.random() * alphabets.length);

return alphabets[randomIndex];
}
function
generateRandomAlphabets(count) {
let randomAlphabets = "";
for (let i = 0; i < count; i++) {
randomAlphabets += getRandomAlphabet();
}
return randomAlphabets;
}
function random(){
const clkbutton=document.getElementById("classId")
const fiveRandomAlphabets = generateRandomAlphabets(5);
clkbutton.value=fiveRandomAlphabets+Math.round(Math.random()*1000)

```

```
}  
</script>  
</body>  
</html>
```

Project.hbs

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Document</title>  
  <link rel="stylesheet" href="/stylesheets/project.css">  
</head>  
<body>  
  <header>  
    <div class="container">  
      <a href="/login">Logo</a>  
      <nav>  
        <a href="/">Home</a>  
        {{#if data}}  
        <a href="/accounts">Account</a>  
        {{/if}}  
        <a href="#">About</a>  
        <a href="#">Contact</a>  
      </nav>  
    </div>  
    </header> <!--header end -->  
    <main>  
  
    <section class="banner-section">  
      <div class="empty" id="follow">  
<a href="/login">Login</a>  
<a href="/accounts">Following</a>  
<a href="">My Group</a>
```

```

</div>
<div class="container">

<div class="section-left">
  <h1>Welcome to <br>
    Logic Bot</h1><br>
  <p>Welcome to our studying website!
    We're excited to have you here and to help you achieve your academic goals. Whether you're
a
  student looking to improve your grades, a professional seeking to enhance your skills, or
  simply
    someone who loves to learn, you've come to the right place</p>
    {{#if data}}
  <button type="button" ><a href="/logout">logout</a></button>
    {{else}}
  <button type="button" ><a href="/login">login</a></button>
    {{/if}}
  <button type="button" ><a href="/newclass">create class</a></button>

</div>
<div class="section-right">

</div>
</div>
</section>
</main>

<main class="main-body">
  <section>
    <div class="main-left">

</div>
    <div class="main-right">
      <h2>Choose Course</h2>
      <form action="/" method="post" >
        <label for="semester">classId:</label><br>

        <input name="classId" ><br>
        <label for="subject">password:</label><br>

```

```

    <input type="password" placeholder="Enter Password" name="password"><br>
</select><br>
<label>Enter Name</label><br>
<input value="" name="name" id="name"><br>
<button type="submit" value="submit">Enter</button>

</div>
</section>
</form>
</main>
<!-- note start -->
<div class="scroll">

    <div>
<button class="left" onclick="scrollwindow1()"></button>
    </div>
    <div class="cover">
<div class="window">
    <div class="prodect">

        <h1 align="center" >C++</h1>
<p>C++ is an object-oriented programming language which gives a clear structure to
    programs and allows code to be reused, lowering development costs.
    C++ is portable and can be used to develop applications that can be adapted
    to multiple platforms. C++ is fun and easy to learn!
    </p>
</div>

<div class="prodect">
    <h1 align="center" >Java</h1>

    <p>Java is a widely used object-oriented programming language and software
        platform that runs on billions of devices, including notebook computers,
        mobile devices, gaming consoles, medical devices and many others. The rules
        and syntax of Java are based on the C and C++ languages.
    </p>
</div>
<div class="prodect">
    <h1 align="center" >C</h1>

```

<p>C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972. It is a very popular language, despite being old. C is strongly associated with UNIX, as it was developed to write the UNIX operating system.

</p>

</div>

<div class="prodect">

<h1 align="center" >javascript</h1>

<p>JavaScript is a dynamic computer programming language. It is lightweight and most commonly

used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented

capabilities.

</p>

</div>

</div>

</div>

<div>

<button class="right" onclick="scrollwindow()">></button>

</div>

</div>

<!-- note end -->

<main class="last-section">

<section>

<div class="features">

<h2>Features</h2>

Courses

Knoeledge

Study packs

Programs

Exam Cram

</div>

<div class="products">

<h2>Products</h2>

Memberships

Our Events

Shop


```

</div>
<div class="ourhub">
  <h2>Our Hub</h2>
  <a href="#">About</a><br>
  <a href="#">News</a><br>
</div>
<div class="support">
  <h2>Support</h2>
  <a href="#">Contact</a><br>
  <a href="#">Terms of Use</a><br>
  <a href="#">Privacy policy</a><br>
</div>
</section>
<samp>Design and Development</samp><span> LogicalBot</span>
</main>
<div>

</div>

<script src="/javascripts/project.js"></script>

</body>
</html>

```

Class.hbs

```

<!DOCTYPE html>
<html>
<head>
<title>Selection Result</title>
<link rel="stylesheet" href="/stylesheets/output.css">
</head>
<body>
<header>
<div class="container">
<a href="#">Logo</a>
<nav>
<a href="/">Home</a>

```

```

<a href="#">About</a>
<a href="#">Contact</a>
</nav>
</div>
</header> <!--header end -->
<div class="main">
<div class="left-main">
<button id="chat" onclick='displaychats()'>chat</button>
{{!-- {{#if userdata}} --}}
<button id="exam" onclick='displaychats2()'>Examination</button>
{{!-- {{/if}} --}}
<button id="pol" onclick='displaychats3()'>Live</button>
</div>
<div class="right-main" >
<div class="rightchat" id="right-chat">
<div class="topclass">
{{#each classdata}}
<h1 id="classname" >{{this.className}}</h1>
{{/each}}
{{!-- following session --}}
{{#if userdata}}
{{#if leader}}
{{else}}
{{#if followdata}}
{{else}}
<button id="follow" onclick="follow()" >Follow</button>
{{/if}}
{{/if}}
{{/if}}
{{!-- following session end --}}
</div>
{{#each username}}
<div id="name" class="name">{{this}}</div>
{{/each}}
{{#each classid}}
<div id="classid" class="empty">{{this}}</div>
{{/each}}
<div class="message" id="message">
{{#each classdata}}
<div id="username" class="username" >{{this.UserName}}</div>

```



```

<div class={{this.textclass}}>{{this.message}}</div>
<div class={{this.class}}>
<div></div>
<a href={{this.Image}} >{{this.originalname}}</a></div>
{{/each}}
</div>
<div class="socket-message" id="socket-message">
</div>
<div class="typing">
{{#if userdata}}
<input class="senddata" id="input-field" type="text" placeholder="Enter message">
<input type="file" id="image-input">
<button id="submit-button" type="submit">Send</button>
{{else}}
<p>you don't login so you can not chat</p>
{{/if}}
</div>
</div>
{{!-- exam section --}}
<div class="right-exam">
<div class="chatempty" id="right-exam">
<h1 bgcolor="white">Exam</h1>
{{#if leader}}
<div class="qsdiv">
<form action="/exam" method="POST">
<div class="createexam" id="createexam">
<input class="newexam" id="newexam" type="button" onclick="newinput()" value="+
required>
<div>
</div>
<div class="addinput" id="addinput">
<label for="qs" id="qslabel">qs1</label>
<input class="examtext" type="text" name="qs1" id="examtext" placeholder="qs"
required>
<label for="mark" class="marklabel">mark</label>
<input class="mark" type="number" name="mark1" id="marktext" placeholder="0"
required>
{{!-- <button id="submit-exam" class="submitexam" type="submit">✔</button> --}}
</div>
</div>

```

```

<div class="lastlook">
<label for="time">Time(m):</label>
<input type="number" name="Time" placeholder="0" required>
<label for="Examname">Examname:</label>
<input type="text" name="Examname" placeholder="examname" required>
<button type="submit" class="examsubmit">submit</button>
</div>
</form>
</div>
{{/if}}
{{#each examdata}}
<div class="examdata">
<h2>ExamName: {{this.Examname}} </h2>
<form action="/answer" method="post" >
<input type="text" class="empty" name="Examname" value={{this.Examname}}>
{{!-- {{#if leader}} --}}
<button type="submit">answer</button>
{{!-- {{/if}} --}}
</form>
<form action="/qustions" method="post">
<input type="text" class="empty" name="Examname" id="Examname"
value={{this.Examname}}>
<label for="ans">Id&RollNo</label>
<input type="text" name="Idnumber" id="rollno" placeholder="0" required><br>
<input type="text" class="empty" name="date" value={{this.adddate}}>
<input type="text" name="time" class="empty" value={{this.addtime}}>
<input type="text" name="am_pm" class="empty" value={{this.am_pm}}>
<button id="ans-submit">Enter</button>
</form>
<samp>{{this.createtime}}</samp>
<samp id="FinalDate">FinalDate: {{this.addtime}} {{this.am_pm}}</samp>
</div>
{{/each}}
</div>
</div>
{{!-- examination session end --}}
{{!-- pol session --}}
<div class="chatempty" id="right-pol">
<div class="chatempty"></div>
hello

```

```

</div>
{ {!-- pol session end --} }
</div>
<script>
const following=document.getElementById("follow")
function follow(){
console.log("click")
following.classList.add('chatempty')
}
</script>
<script> const submitButton = document.getElementById('submit-button');
const textInput = document.getElementById('input-field');
submitButton.addEventListener('click', () => {
const imageFile = imageInput.files[0];
if(imageFile){
sendData2( imageFile);
imageInput.value=""
}else{
console.log("no data")
}
const textData = textInput.value;
if(textData){
sendData(textData);
textInput.value = "";
}
});
const sendData2 = async (image) => {
try {
const formData = new FormData();
formData.append('image', image);
const response = await fetch('/results', {
method: 'POST',
body: formData
});
if (response.ok) {
console.log('Data saved successfully');
} else {
console.log('Error:', response.status);
}
} catch (error) {

```

```

console.log('Error:', error);
}
};
const sendData = async (text) => {
  try {
    const formData = new FormData();
    formData.append('text', text);
    const response = await fetch('/results', {
      method: 'POST',
      body: formData
    });
    if (response.ok) {
      console.log('Data saved successfully');
    } else {
      console.log('Error:', response.status);
    }
  } catch (error) {
    console.log('Error:', error);
  }
};
</script>
<script src="/socket.io/socket.io.js"></script>
<script src="/javascripts/sockethelp.js"></script>
<script src="/javascripts/chat.js"></script>
</html>

```

Login.js

```

var express = require('express');
var router = express.Router();
const path=require('path')
const bodyParser = require('body-parser');
const userhelper=require('./helper/user-helper');
/* GET home page. */
router.get('/', function(req, res, next) {
  req.session.rel=false
  data2=req.session.rel2
  res.render('login', {data2})

```

```

req.session.rel2=false
});
router.post('/', bodyParser.urlencoded({ extended: true }), async (req, res) => {
  userhelper.doLogin(req.body)
    .then((response)=>{
      if(response.status){
        req.session.loggedId=true
        req.session.data=response.data
        req.session.name=response.name
        res.redirect("/")
      }else{
        req.session.rel2=true
        res.redirect("/login")
      }
    })
  // res.redirect("/")
})
module.exports = router;

```

Package-lock.json

```

{
  "name": "main",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "nodemon ./bin/www"
  },
  "dependencies": {
    "bcrypt": "^5.0.1",
    "body-parser": "^1.20.2",
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "express": "^4.18.2",
    "express-handlebars": "^5.0.0",
    "express-session": "^1.17.3",
    "handlebars": "^4.7.7",
    "hbs": "^4.2.0",
    "http-errors": "~1.6.3",

```

```
"mongodb": "^5.5.0",  
"mongoose": "^5.13.17",  
"morgan": "~1.9.1",  
"multer": "^1.4.5-lts.1",  
"socket.io": "^4.6.2"  
}  
}
```

7.TESTING

Software testing is a critical element of software quality assurance and represent the ultimate review of specification, design and coding. System testing is actually a series of different test whose purpose is to fully exercise the computer-based system. Although each has a different purpose, all of them work to verify that all system elements have been properly integrated and all of them perform allocated functions. If the test is conducted successfully, it will uncover errors in the software. A second benefit is that the software is appearing to be working according to specification and that performance requirements appear to have been met.

System testing is an inexpensive but critical process that can take as much as 50% of budget for development, the view of testing holds by users that is performed to prove that there is no error in the program. However, this is virtually impossible since analysis cannot prove that software is free and clear to errors. Testing is the process of executing a program with explicit intentions of finding errors.

7.1 LEVELS OF TESTING

7.1.1 Unit Testing

Unit testing comprises of a set of tests performed by an individual programmer prior to the integration of unit into a large system. A large program unit is usually small enough that the programmer who developed it can be in great detail and certainly in greater detail than possible when the unit is integrated into an evolving software project. Module unit testing should be as exhaustive as possible to ensure that each representation is handled by each module that has been tested. All the units that make up the system were tested independently to ensure that they work as required. These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.

7.1.2 Integration testing

Integration testing is a system technique for constructing program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take the unit tested modules and build a program structure that has been dictated by design.

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together (“big bang”). Normally, the former is considered as a better practice since it allows interface issues to be located more quickly and fixed.

Integration testing works to detect defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

7.1.3 system testing

System testing is actually a series of different tests, whose purpose is to test the completed system in its entirety as a whole exercise. System testing involves security testing, performance testing and recovery testing. After output testing the whole system is tested in different platforms and browsers for testing the overall functionality and user interface of the system in various environments. This testing is done after completing all the testing, i.e. system testing is the final phase of the testing process.

7.1.4 validation testing

At the culmination of integration testing, the software was completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software validation testing began. In validation testing the entered data validated for correct format, and correct order.

7.1.5 output testing

After performing the validation test, the next phase is the output test of the system, since no system could be useful if it does not produce desired output in the desired format. The output format was considered in two ways: one is on the screen and the other as a printed form

8. SYSTEM IMPLEMENTATION

Implementation is the process of converting a new or revised system design into operation. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system. It must therefore be carefully planned and controlled. Apart from planning the two major tasks of preparing for implementation are education and training of users and testing of the system. Education of users should really take place much earlier in the project .i.e. when they are involved in the investigation and design work. Training has been given to the staff regarding the new system. Once staff has been trained, the system can be tested. Implementation is the stage of a project where the theoretical design is turned into a working system. Therefore it must be carefully planned and controlled. It can also be considered as the most crucial stage in achieving a successful new system and giving the user confidence that the new system will work and be effective. Implementation is the final and important phase. It is the phase where theoretical design is turned into a working system, which works for the user in the most effective manner. It involves careful planning investigation of the present system and the constraints involved, user training, system testing and successful running of the developed proposed system. The Implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out regarding the equipment and resources and the additional equipment have been acquired to implement the new system. The user tests the developed system and the changes are made according to the needs. The testing phase involves the testing of a system using various kinds of data. The method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain types of transaction while using the new system. An elaborate testing of data is prepared and the system is tested using that data, while testing errors are noted and corrections are made. The users are trained to operate the developed system. Both hardware and software are made to run the developed system successfully in nature. The method of implementation and the scale to be adopted are found out initially. Next the system is tested properly and the users are trained in the new procedures. 48 10.

9.MAINTENANCE

Maintenance is a characteristic of design and implementation, which is expressed as the probability that an item will be retained in or restored to a specific condition within a given period of time. When maintenance is performed in accordance with the prescribed procedures and resources. Maintenance is the enigma of system development. It holds the software industry captive, tying up programming resources. Analysts and programmers spend far more time maintaining programs than writing them. Maintenance can be classified as corrective, adaptive or perfective. Corrective maintenance means repairing processing or performance failure or making changes because of previously uncorrected problems or false assumptions. Adaptive means repairing process or performance or modifying the program to respond to the user's additional or changing needs of this type. More time and money are spent on prefectures than on corrective and adaptive maintenance. Technical and management approaches to the maintenance phase can be implemented with little upheaval. However , tasks performed during the software engineering process define maintainability and have an importance on the success of any maintenance approach.

10.FUTURE ENHANCEMENT

- **Screen sharing:** This would allow users to share their screen with each other, which could be useful for presentations, troubleshooting, or collaboration.
- **Games and activities:** This would make the chat experience more fun and engaging. For example, users could be able to play games, watch videos, or listen to music together.
- **Location sharing:** This would allow users to share their location with each other, which could be useful for meeting up or finding friends nearby.
- **customization options:** For example, users could be able to choose from different themes, fonts, and emojis.
- **Improved user interface and user experience:** This includes making the chat interface more intuitive and easy to use, as well as providing more.

11.SECURITY, BACKUP AND RECOVERY MECHANISMS

Security is an important consideration in application. By default PHP applications are available to any user who can connect to our server. Although this is ideal for many applications it is not always appropriate. The first step in securing our application is deciding where you need security and what it needs to protect. Security concept:

1. Authentication :

This is the process of determining a user's identity and forcing users to prove they are who they claim to be, usually this involves entering credentials (username and password) in some sort of login or windows.

2. Authorization :

Once the user is authenticated, authorization is the process of determining where that user has sufficient permission to perform a given action, such as viewing a page or retrieving information from a database.

Backup facility is used in this software for backing up data. If any error occurs in the database due to any database error or software error or if the database is detected in any fault operation, you can copy the backup file to solve this problem. For protecting the system from any kind of loss or damage, backup facility is offered. The entire program that is associated with the database can be saved into CD or DVD or floppy disc for the purpose of future use. If the program is lost due to some kind of system failure, the backup copy of the program on any of the above disks can be used. The data or those discs will not be lost due to any usual failure.

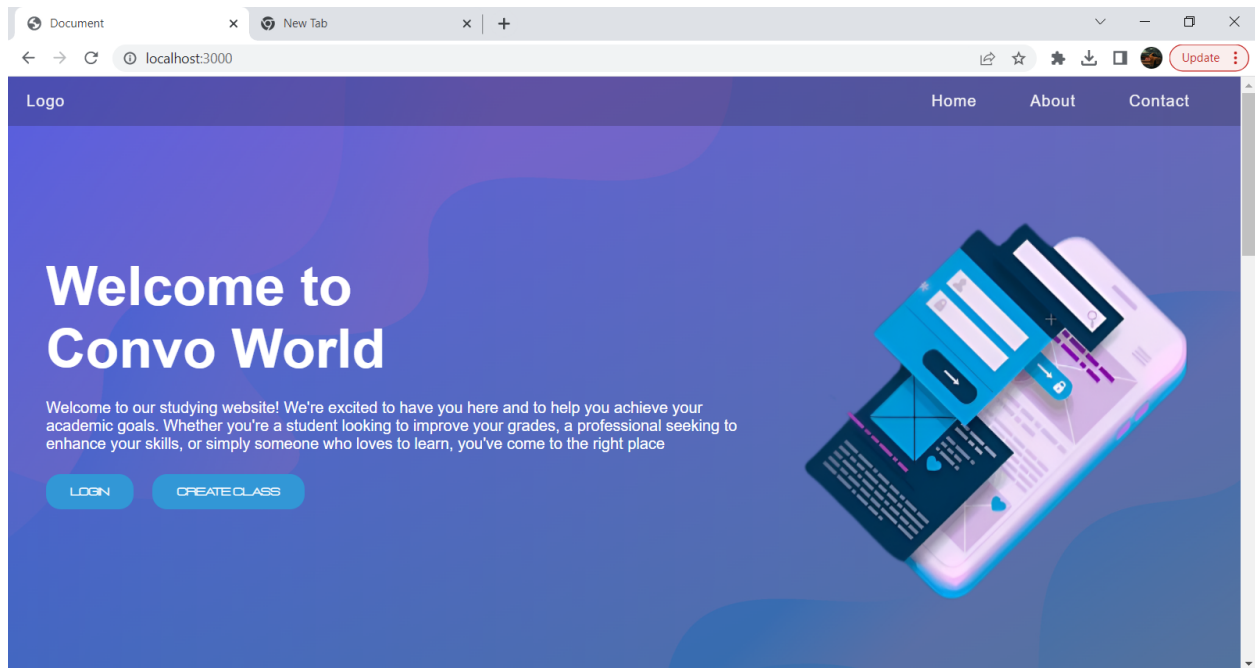
12.CONCLUSION

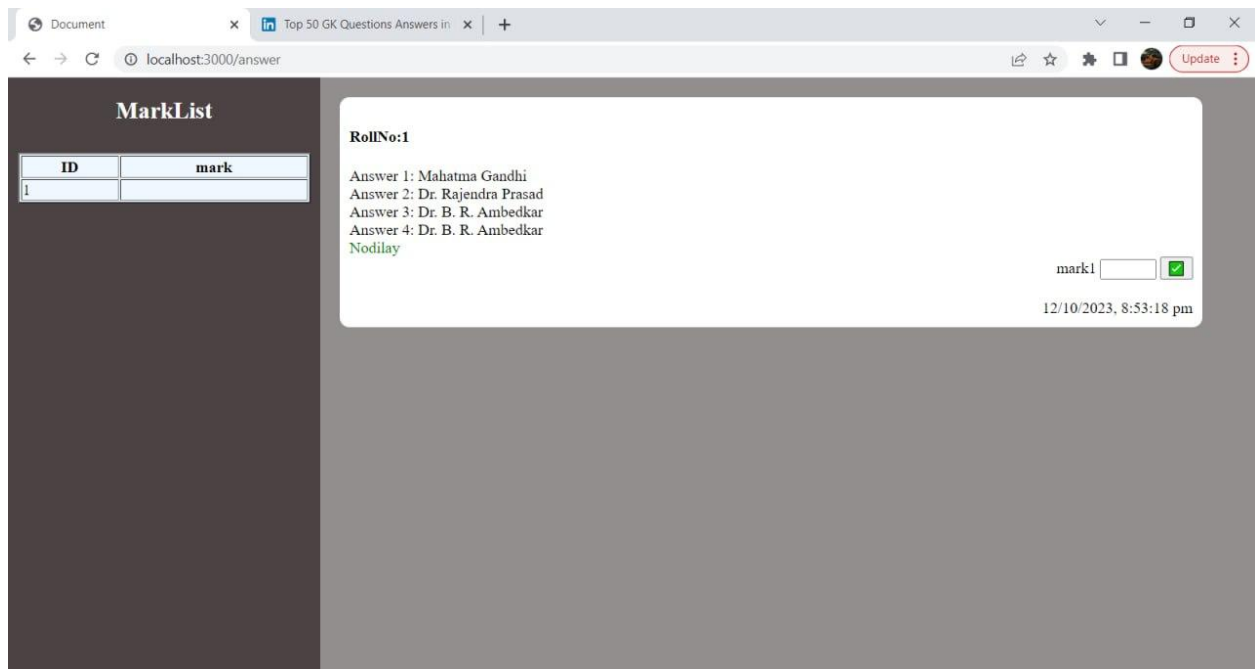
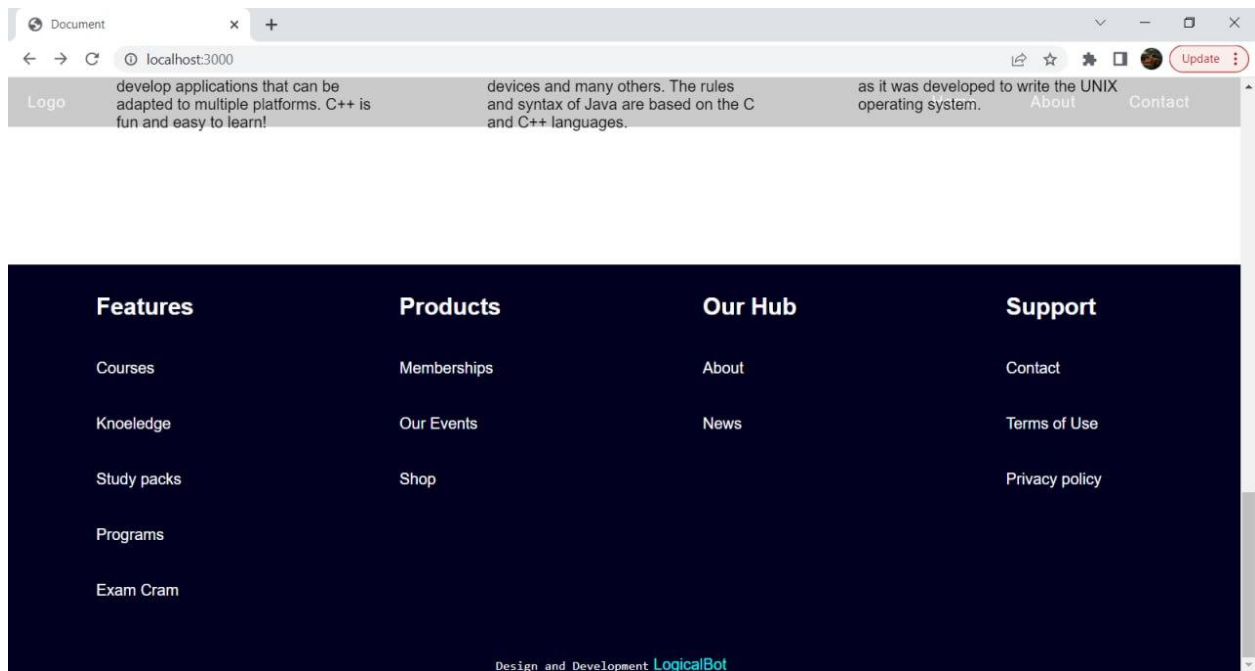
We prepare a website that provide a free chatting box for users. This chatting page can handle text message ,images,video files,audio files,etc. This website mainly focused on study purpose. For using this website user want to create an account. After creating account a user can open a channel . This channel is work as a chating space . It has an exam or text paper feature. It help full for teachers to conduct online exams.

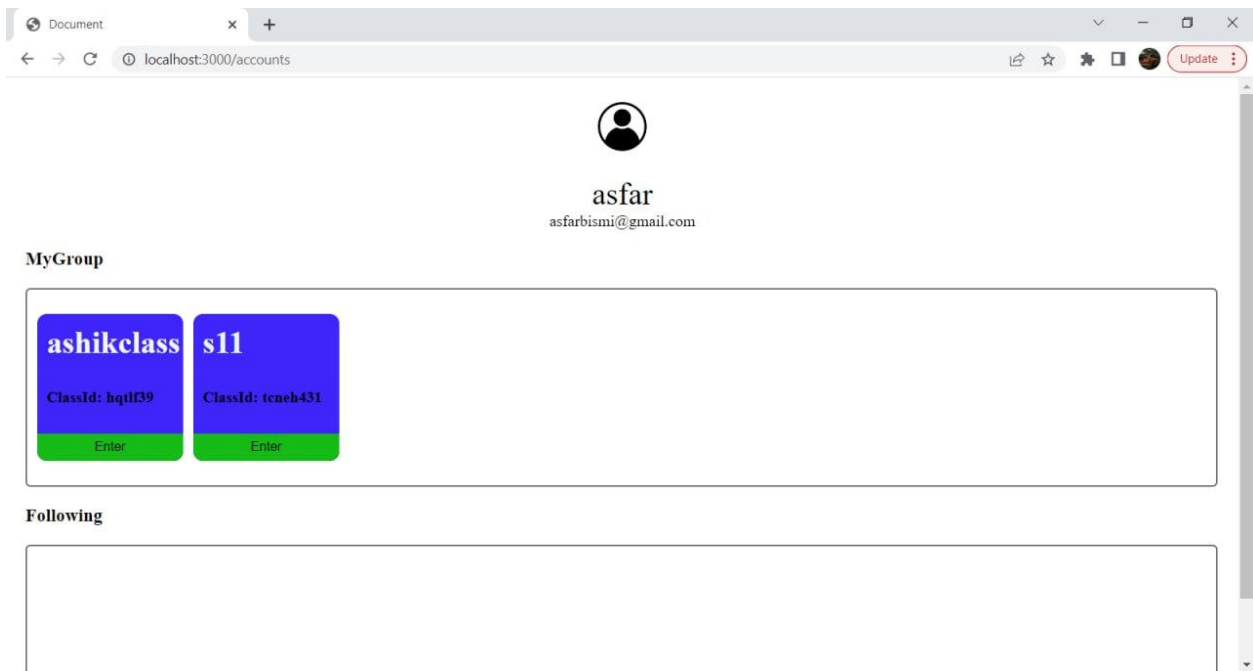
The user have an account then he/she can follow others account and channels for future reference. By the use of this website students and teachers can communicate teacher without the help of any applications. This is absolutely free website it doesn't need any personal information. This website can be a good choice for students. In future this website will get more features.

13.APPENDIX

13.1 SCREENSHOTS







Selection Result x Top 50 GK Questions Answers in x +

localhost:3000/results

chat

Examination

Exam

qs1	Who is the Father of our Nation?	mark 1
qs2	Who was the first President of India?	mark 1
qs3	Who was the first President of India?	mark 1
qs4	Which is the most sensitive organ in our body?	mark 1

Time(m): Examname:

Document x Top 50 GK Questions Answers in x +

localhost:3000/questions

Qs1: Who is the Father of our Nation? Mark1:1

Qs2:Who was the first President of India? Mark2:1

Qs3:Who was the first President of India? Mark3:1

Qs4:Which is the most sensitive organ in our body? Mark4:1

Document

Top 50 GK Questions Answers in

localhost:3000/answer

Update

MarkList

ID	mark
1	

RollNo:1
Answer 1: Mahatma Gandhi
Answer 2: Dr. Rajendra Prasad
Answer 3: Dr. B. R. Ambedkar
Answer 4: Dr. B. R. Ambedkar
Nodilay

mark1

☒

12/10/2023, 8:53:18 pm

13.2 GANTT CHART

Gantt chart shows the time relationship between 'events' of the production program has been regarded as revolutionary in management. Gantt charts recognize the total program goals and it should be regarded as a series of interrelated supporting plans (or events) that people can comprehend and follow.

The plan explains the task versus the time they will take to complete.

Sl. No.	Task Name	Duration	Start Design	End Design	February				March				April				May			
					W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4
1	Analysis	10	01.02.2023	10.02.2023																
2	System Design	18	11.02.2023	28.02.2023																
3	Code Design	40	05.03.2023	13.04.2023																
4	Testing	24	05.04.2023	28.04.2023																
5	Documentation	18	29.04.2023	17.05.2023																