Annexure – 1

SUPPORT HUB

A Project Report submitted
in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

1. N.SHANTHI SRI
2. N.KAVYA SRI
3. M.DIVYA SARADA
4. N. ANUPAMA RANI

Under the esteemed guidance of
Mr. P. SUNIL
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)
(Approved by AICTE, Accredited by NBA & NAAC, Affiliated to JNTU Kakinada)
BHIMAVARAM – 534 202
2020 – 2021

Annexure – 2

SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)

(Approved by AICTE, Accredited by NBA & NAAC, Affiliated to JNTU Kakinada)
BHIMAVARAM – 534 202

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

*This is to certify that the project entitled "**Support Hub**", is being submitted by **N.SHANTHI SRI,N.KAVYA SRI, M.DIVYA SARADA N. ANUPAMA RANI** bearing the **Regd. No. 17B01A05C1, 17B01A05A9, 17B01A0596, 17B01A05A7** in partial fulfillment of the requirements for the award of the degree of "**Bachelor of Technology** in **Computer Science & Engineering**" is a record of bonafide work carried out by her under my guidance and supervision during the academic year 2020–2021 and it has been found worthy of acceptance according to the requirements of the university.*

Internal Guide **Head of the Department**

External Examiner

ACKNOWLEDGEMENT

It is nature and inevitable that the thoughts and ideas of other people tend to drift in to the conscious due to various human parameters, where one feels acknowledge the help and guidance derived from others. We acknowledge each of those who have contributed for the fulfillment of this project report.

We wish to place our deep sense of gratitude to **Sri. K. V. Vishnu Raju**, Chairman of SVES, for his constant support on each and every progressive work of ours.

We express our heart full thanks to our principal **Dr. G. Srinivasa Rao**, for providing us with all facilities necessary to carry out this project successfully.

We wish to express our sincere thanks to Vice Principal **Dr. P. Srinivasa Raju**, for being a source of an inspiration and constant encouragement.

We are privileged to express our sincere gratitude to honorable Head of The Department **Dr. P. Kiran Sree**, for giving his support and guidance in our endeavors.

We express our deep sense of gratitude and sincere appreciation to PRC members, **Dr. VVR Maheshwara Rao, Mrs. P R Sudha Rani, Dr. D. Dharmaiah** for giving his support and guidance in our endeavors.

We express our deep sense of gratitude and sincere appreciation to our guide **Mr.P.Sunil**, Assistant Professor (CSE), for his indispensable suggestions, unflinching and esteemed guidance throughout the project.

It has been great pleasure doing project work at Shri Vishnu Engineering College for Women as a part of our curriculum, we express our profound gratitude to who gave us an opportunity to work in this esteemed organization.

PROJECT TEAM:

Nunnaboyina Shanthi Sri	17B01A05C1
Nelli Kavya Sri	17B01A05A9
Medipati Divya Sarada	17B01A0596
Neerukonda Anupama Rani	17B01A05A7

ABSTRACT

Support Hub is a Web based application that provides a platform for students and faculty to clarify their doubts through a web portal.

Support Hub is a resource intended to provide the Students or end users with information and support related to a subject / technical information.

The purpose of this application is usually to clarify doubts. In this application faculty members will be given rights to solve the tickets raised by the students.

A typical Support Hub can effectively perform several functions. It provides a single point of contact for users to gain assistance in solving queries, getting answers to questions.

In this application students and faculty are logged into portal. After logging into the portal students are given opportunity to raise ticket for each and every doubt they have.

Then it will be the task of the faculty to address the ticket and clarify the corresponding student doubts. After clarifying the doubts the ticket which is generated, will be deleted. Students have the feasibility of raising the ticket to one concerned faculty without sending to all the remaining faculty.

Annexure – 3

Contents

INTRODUCTION

1. INTRODUCTION

Support Hub is a Web based application that provides a platform for students and faculty to clarify their doubts through a web portal. It is a resource intended to provide the Students or end users with information and support related to a subject / technical information. The purpose of this application is usually to clarify doubts. In this application faculty members will be given rights to solve the tickets raised by the students. A typical Support Hub can effectively perform several functions. It provides a point of contact for users to gain assistance in solving queries, getting answers to questions. In this application students and faculty are logged into the portal. After logging into the portal students are given the opportunity to raise tickets for each and every doubt they have. Then it will be the task of the faculty to address the ticket and clarify the corresponding student doubts. After clarifying the doubts the ticket which is generated, will be deleted. Apart from students and faculty we also have an admin page where each and every action is noticed and grants permission. When a student or faculty is registered then we undergo a two-step verification process. In the first step, a mail is sent to the email id, provided by the user, in which we have a link click here which validates the first step. After that in the admin page, admin checks the details of the registered user and makes sure that they belong to the community, if so then approves the registration, else denies it. This ensures a best two-step verification process. Admin also contains the details of the students and faculty who generated and answered the question. If there are multiple faculty answering particular student questions then they will be appended with faculty names. This not only manages to have a clear answer but also arranges all in a neat sequential way.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

Every organization has at least one help desk system, where complaints and queries can be stored, resolved and saved for future reference. But as this Help Desk Management System is worked manually, it will take a lot of time and labor to solve all the problems of the customers. And even to handle too many customers at one time. Manually the work is even increased, just like writing the whole data, query and other requirements, apart from sending them messages or replying to them manually. Customers often ask the same questions or raise the same types of incidents, prompting the support team to repeatedly provide the same answers, which increases repetitive and time-wasting tasks.

2.2 PROPOSED SYSTEM:

The goal of the project is to provide a medium between students and faculty to raise and clarify their doubts through an online portal. In this project we are going to create a web application which enables students to raise their doubts as tickets. When a student asks a question, a ticket is generated at the student end and it is also viewed at the faculty end. Students can then delete or update the ticket on the basis of his/her progress. We have three modules: Student, Faculty, Admin. Student module helps students to create, view, and edit their tickets. Faculty module helps faculty to view the tickets generated by students in particular subjects dn allows them to answer the doubts. More than one faculty can answer a question and they will be displayed in the student module one after the other. Admin module validates the student and faculty registration and login. After successful registration a mail is sent to the user which consists of a link and the user needs to login into the website using that link. When a faculty answers a question, it will be displayed in the student module with the name of the faculty who answered the question. We can also get the information of the questions asked in each subject, questions asked by each student, and answers given by the faculty.

2.3 FEASIBILITY STUDY:

Preliminary investigation examines project feasibility, the likelihood the application will be useful to the user. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging traditional desktop centric applications. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- ➔ Technical Feasibility
- ➔ Operation Feasibility
- ➔ Social Feasibility

TECHNICAL FEASIBILITY:

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- ➔ Does the necessary technology exist to do what is suggested?
- ➔ Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?
- ➔ Can the system be upgraded if developed?
- ➔ Are there technical guarantees of accuracy, reliability, ease of access and data security?

OPERATIONAL FEASIBILITY:

User-friendly

Users need to just select a subject. Then the user either raises a ticket or answers a query. When faculty just clicks on a particular subject then all the tickets generated in that subject will be displayed on the screen.

Portability

The application will be developed using standard open source technologies like web applications etc. These technologies will work on all web platforms. Hence portability problems will not arise.

Availability

This software will always be available.

SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user.
This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM REQUIREMENTS SPECIFICATION

3.SYSTEM REQUIREMENTS SPECIFICATION

3.1 SOFTWARE REQUIREMENTS:

- ➔ VS code editor
- ➔ GitHub
- ➔ Heroku
- ➔ MongoDB
- ➔ Mongoose

3.2 HARDWARE REQUIREMENTS:

- ➔ Processor: I3
- ➔ Ram: 4GB

3.3 FUNCTIONAL REQUIREMENTS:

- ➔ React JS
- ➔ HTML
- ➔ Bootstrap
- ➔ CSS
- ➔ Express

➔ Node js

➔ Socket

➔ MongoDB Atlas

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 INTRODUCTION:

After the analysis, it must be clear what must be done. In the system we plan to do it. The objectives of the system design are to decide in general how the system should be implemented. Individual physical analysis we prepare a blueprint for the system. Actually, system design is the process of planning a new business system or one to replace or complement an existing system. Analysis of what the system should do and design specifies how to accomplish the objectives. System design begins with a search for alternative solutions. The design of an information system produces the details that state how a system will meet the requirements identified during system analysis. This process is known as logical design and the process of development program software is referred to as physical design. The design process begins by identifying reports and other outputs the system will produce. Software design serves as the foundation for all software engineers and the software maintenance states that follow.

4.2 UML Diagrams:

An Overview of UML:

The UML is a language for

- Visualizing
- Specifying
- Construction
- Documentation

RELATIONSHIPS IN THE UML:

There are four kinds of relationships in UML:

1. Dependency
2. Association
3. Generalization
4. Realization

1. CLASS DIAGRAMS:

Class diagrams are the most common diagrams found in modeling object-oriented systems. A class diagram shows set of classes, interfaces and collaborations and their relationships. Graphically, a class diagram is a collection of vertices and arcs.

Contents:

- Classes
- Interfaces
- Collaboration
- Dependency, generalization and association relationships.

2. USE CASE DIAGRAMS:

Use Case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of the systems. Use case diagrams are central to modeling the behavior of the system, a sub system, or a class. Each one shows a set of use cases and actors and relationships.

Common Properties:

A use case diagram is just a special kind of diagram and shares same common properties, as do all other diagrams a name and graphical contents that are a projection into the model. What distinguishes a use case diagram from all other diagram is it particular content.

Contents:

Use case diagram commonly contain:

- Use Cases
- Actors
- Dependency, Generalization and association relationships.

Like all other diagrams, use case diagrams may contain packages, which are used to group elements of your model into large chunks. Occasionally, you will want to place instances of use cases in your diagrams, as well, especially when you want to visualize a specific executing system.

3. INTERACTION DIAGRAMS:

An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may not be dispatched among them. Interaction diagrams are used for modeling the dynamic aspects of the system. A sequence diagram is a table that shows objects arranged along the X-axis and messages, ordered in increasing time, along the Y-axis and messages, ordered in increasing time, along the Y-axis.

Contents:

- Objects
- Links
- Messages

4. SEQUENCE DIAGRAMS:

A sequence diagram is an interaction diagram emphasizes the time ordering of the messages. Graphically, a sequence diagram is a table that shows objects arranged along the X-axis and messages, ordered increasing time, along the Y-axis.
Typically, you place the objects the initiates the interaction at the left and increasingly more sub-routine objects to the right .Next, you the messages that these

object send and receive along the Y-axis, in ordering of increasing time from top to bottom. This gives the reader a clear visual cue to the flow of control over time.

Sequence diagrams have two increasing features:

- There is the object lifeline. An object lifeline is the vertical dashed line that represents an object over a period of time. Most objects that appear in the interaction diagrams will be in existence from top of the diagram to the bottom.

- There is a focus of control. The focus of control is tall, thin rectangle that shows the period of time during which a object is performing an action, either directly or through the subordinate procedure. The top of the rectangle is aligns with the action the bottom is aligned with its completion.

5. COLLABORATION DIAGRAMS:

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

Following are some of the use cases enlisted below for which the collaboration diagram is implemented:

- To model collaboration among the objects or roles that carry the functionalities of use cases and operations.
- To model the mechanism inside the architectural design of the system.
- To capture the interactions that represent the flow of messages between the objects and the roles inside the collaboration.
- To model different scenarios within the use case or operation, involving a collaboration of several objects and interactions.
- To support the identification of objects participating in the use case.

6. ACTIVITY DIAGRAMS:

An activity diagram is essentially a flow chart showing flow of control from activity. They are used to model the dynamic aspects of a system. They are used to model the flow of an object as it moves from state to state at different points in the flow of control.

An activity is an ongoing non-atomic execution within a state machine. Activities ultimately result in some action, which is made up of executable atomic computations that in a change of state of distinguishes a use case diagram from all other kinds of diagrams is its particular content.

7. STATE CHART DIAGRAMS:

A state chart diagram shows a state machine. State chart diagrams are used to model the dynamic aspects of the system. For the most part this one whose behavior is best characterized by its response to events dispatched from outside its context.

A state chart diagram show a state machine emphasizing the flow of control from state to state. A state machine is a behavior that specifies the sequence of states an object during its lifetime in response to events together with its response to the events.

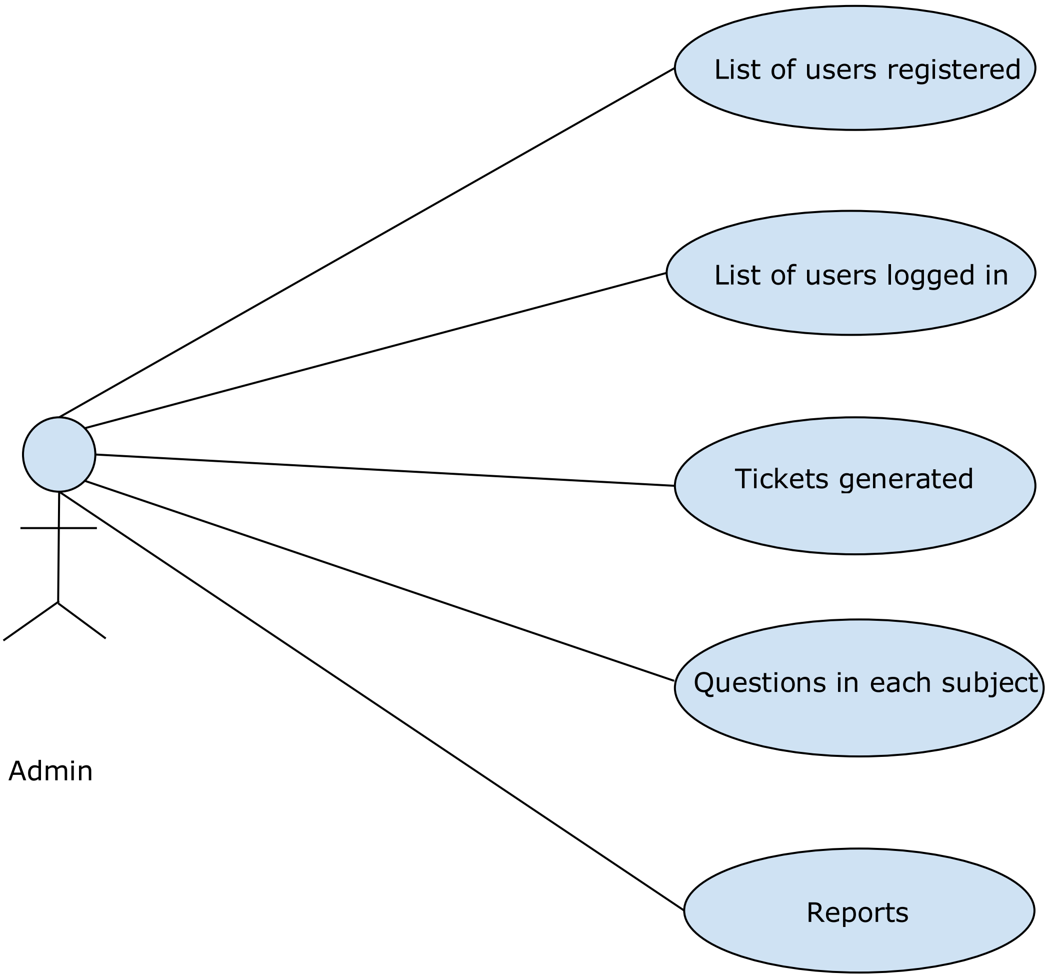
Graphically a state chart diagram is a collection of vertices and arcs. State chart diagram commonly contain:

- Simple states and compound states.
- Transitions, including events and action

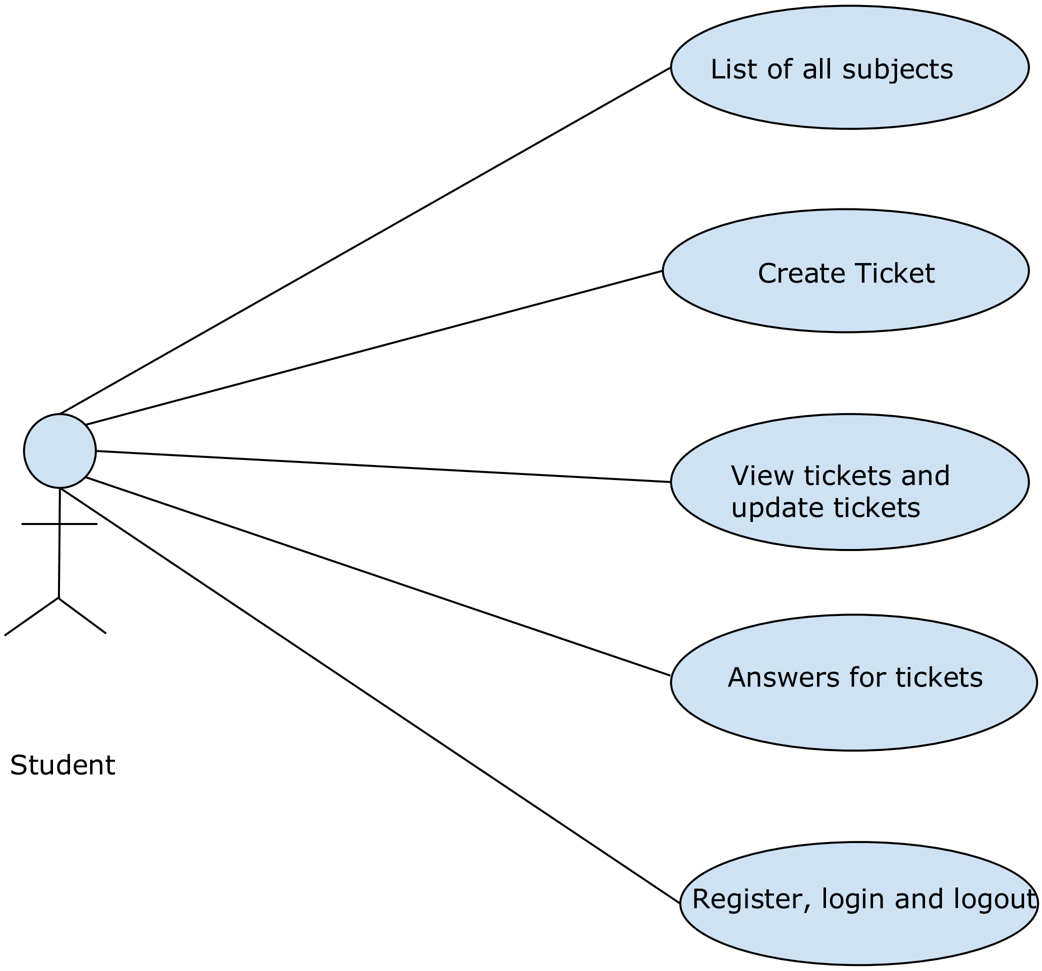
UML Diagrams:

Use case diagram:

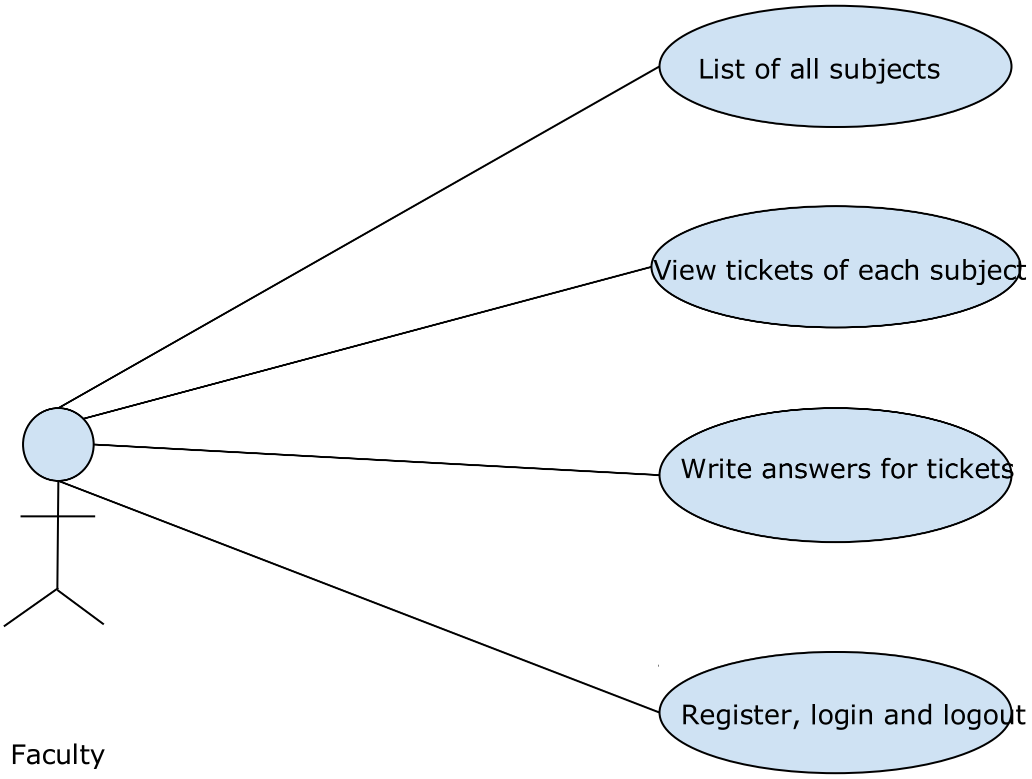
Admin:



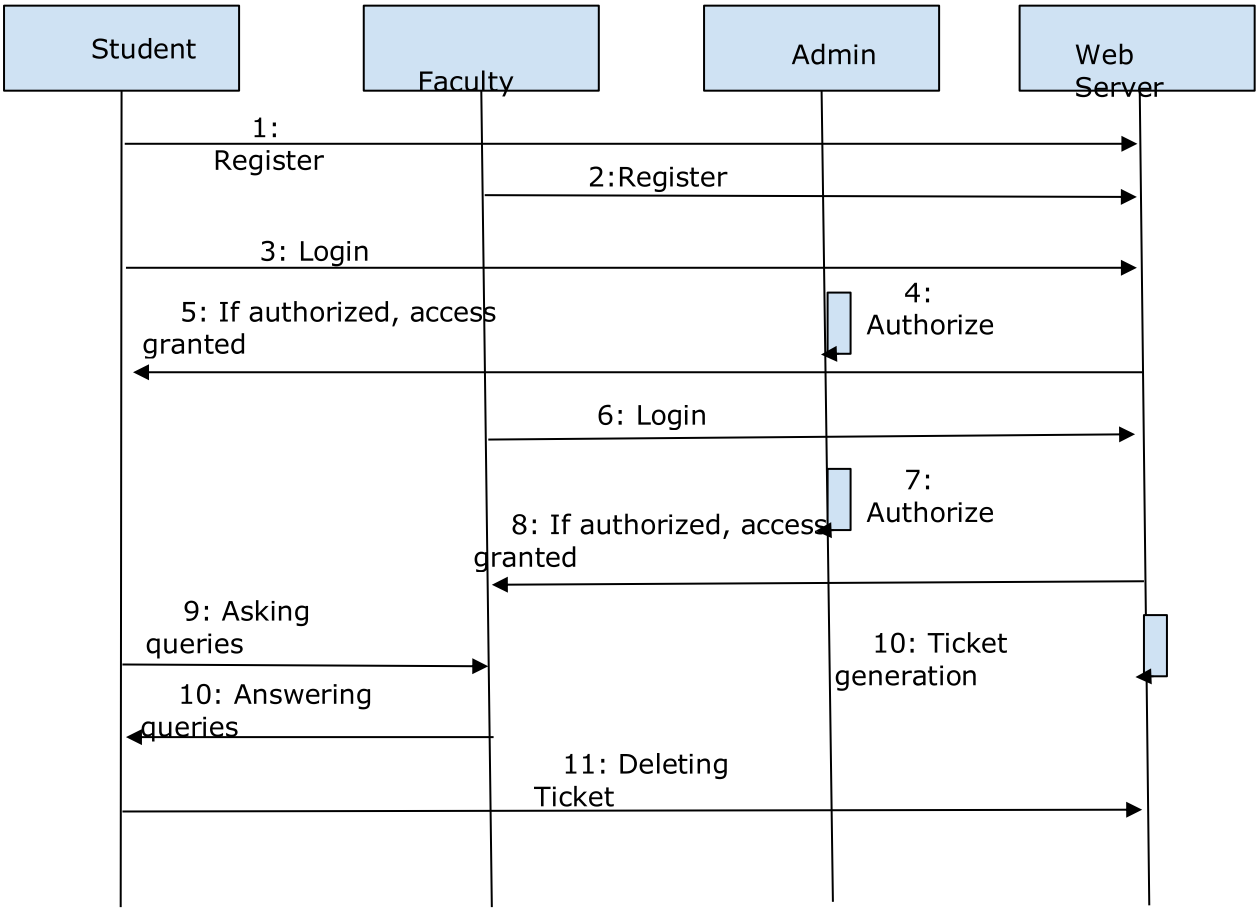
Student:


List of all subjects
Create Ticket
View tickets and update tickets
Answers for tickets
Register, login and logout
Student

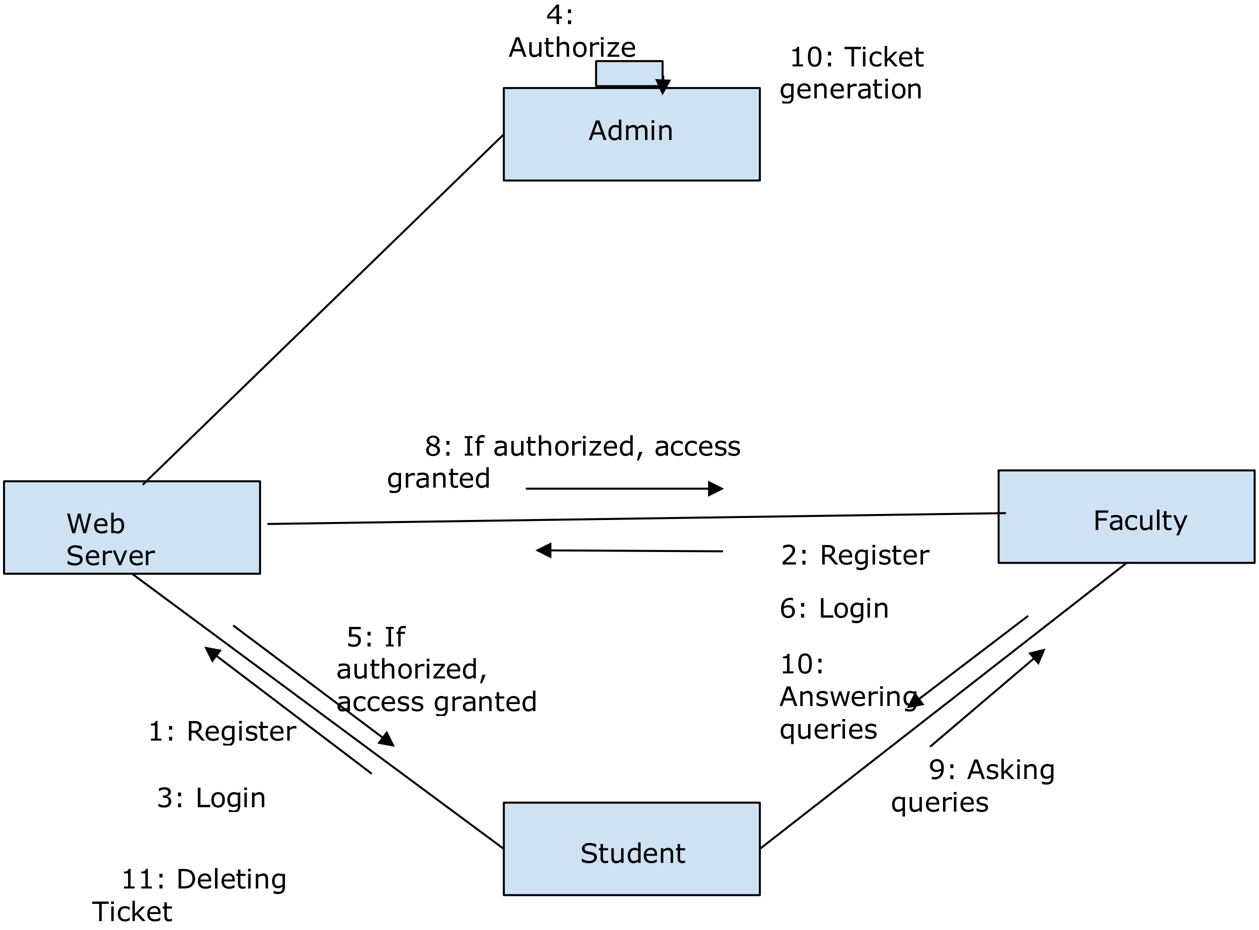
Faculty:


List of all subjects
View tickets of each subject
Write answers for tickets
Register, login and logout
Faculty

Sequence diagram:

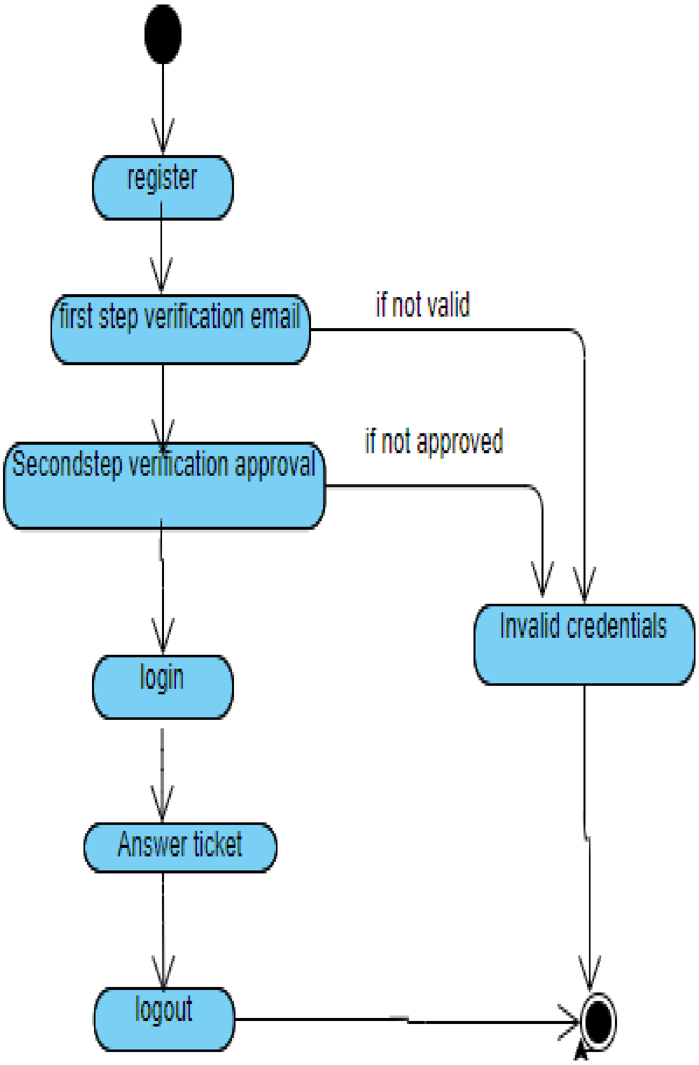


Collaboration diagram:

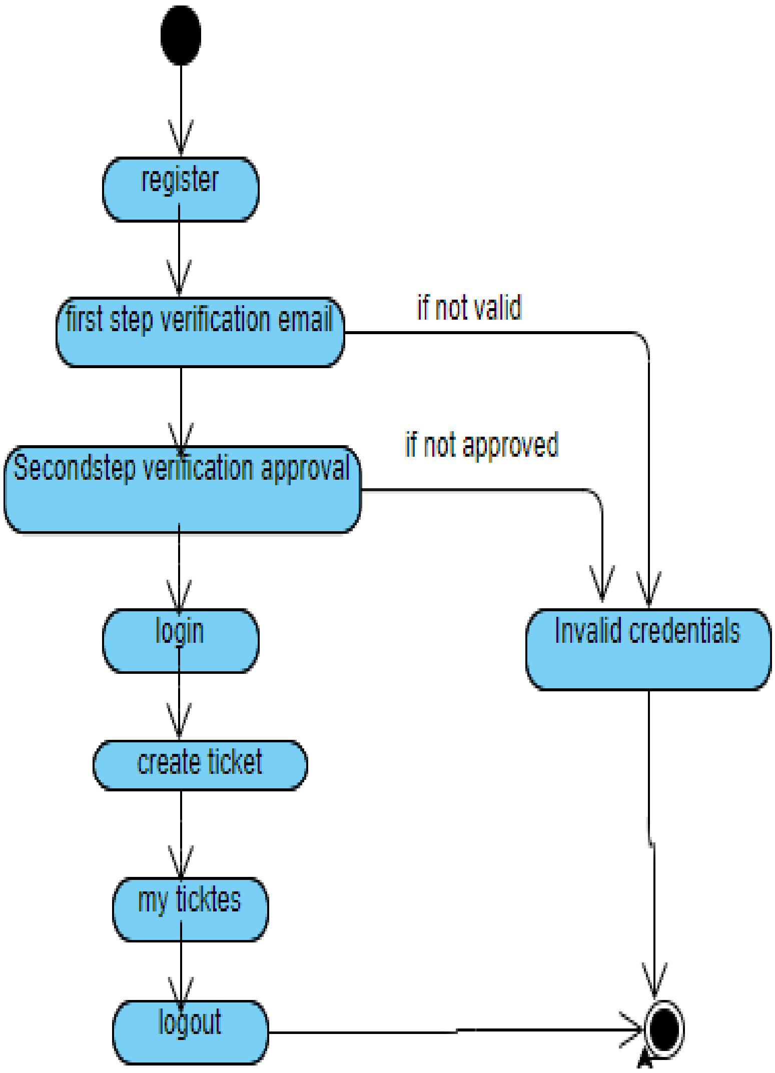

4: Authorize
10: Ticket generation
Admin
8: If authorized, access granted
Web Server
Faculty
2: Register
6: Login
10: Answering queries
5: If authorized, access granted
1: Register
3: Login
11: Deleting Ticket
9: Asking queries
Student

State Chart Diagram:

Faculty:



Student:

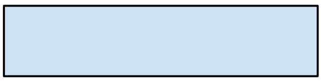


4.3 Database Design

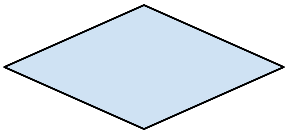
4.3.1 ER DIAGRAMS

An entity relationship diagram(ERD) shows the relationships of entity set stored in a database. An entity in this context is an object, a component of data. In other words, ER diagrams help you to explain the logical structure of databases. At first glance, an ER diagram looks very much like a flowchart. It specializes in symbols, and its meaning of those symbols that make it unique.

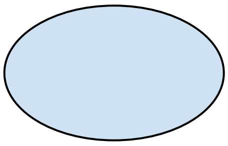
Symbols and Notations



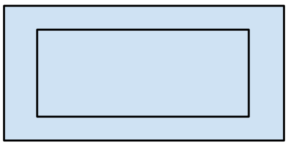
Entity



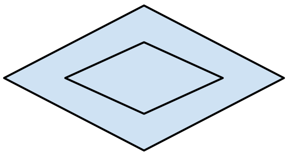
Relationship



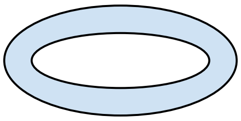
Attribute



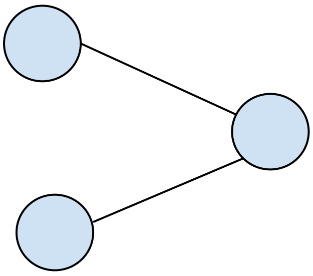
Weak Entity



Weak Entity Relation



Multi valued Attribute



Composite Attribute

Entity relations are of different types.

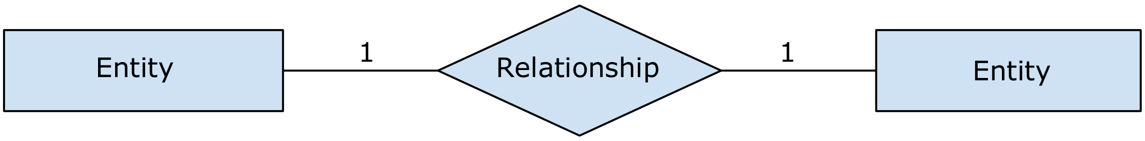
1.One-One

2.One-Many

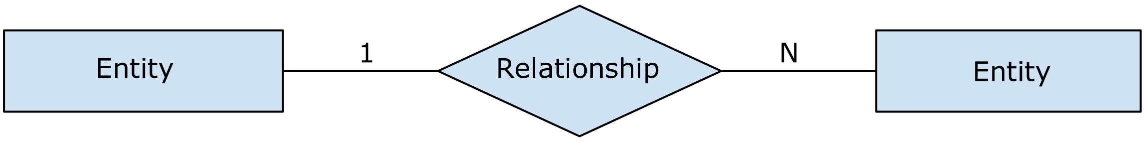
3.Many-One

4.Many-Many

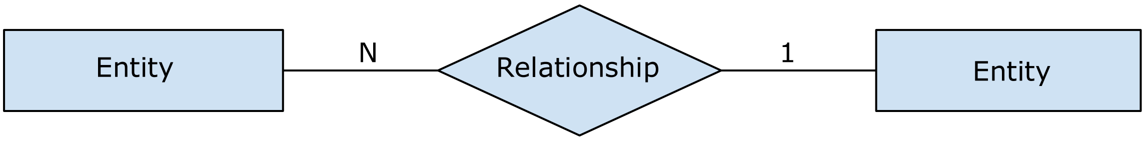
One-to-One: When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts a one-to-one relationship.



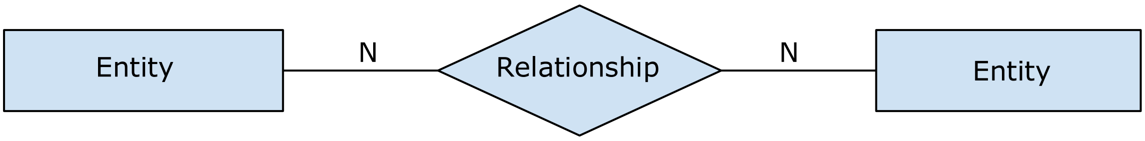
One-to-Many: When more than one instance of an entity is associated with a relationship, it is marked as '1:N'.The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts a one-to-many relationship.



Many-to-one: When more than one instance of an entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts a many-to-one relationship.



Many-to-many: The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many.


Entity
N
Relationship
N
Entity

SYSTEM IMPLEMENTATION

5. SYSTEM IMPLEMENTATION

5.1 INTRODUCTION:

Support Hub is a resource intended to provide the student or end user with information and support related to a particular subject. The purpose of a Support Hub is usually to solve problems or provide guidance about subjects and doubts. Institutions usually provide a Support Hub to their students through websites. A typical help desk can effectively perform several functions. It provides a single point of contact for users to gain assistance in getting answers to questions, and solve known problems. Support Hub is a Web based application that provides a platform for students and faculty to clarify their doubts through a web portal. Support Hub is a resource intended to provide the Students or end users with information and support related to a subject / technical information. The purpose of this application is usually to clarify doubts. In this application faculty members will be given rights to solve the tickets raised by the students. A typical Support Hub can effectively perform several functions. It provides a point of contact for users to gain assistance in solving queries, getting answers to questions. In this application students and faculty are logged into the portal. After logging into the portal students are given the opportunity to raise tickets for each and every doubt they have. Then it will be the task of the faculty to address the ticket and clarify the corresponding student doubts. After clarifying the doubts the ticket which is generated, will be deleted. Apart from students and faculty we also have an admin page where each and every action is noticed and grants permission. When a student or faculty is registered and then logged into the website then a confirmation mail is sent to the student or faculty in order to make sure that their email id is working and also validates the users logged into the website. The goal of the project is to provide a medium between students and faculty to raise and clarify their doubts through an online portal. In this project we are going to create a web application which enables students to raise their doubts as tickets. When a student asks a question, a ticket is generated at the student end and it is also viewed at the faculty end. Students can then delete or update the ticket on the basis of his/her progress. We have three modules: Student, Faculty, Admin. Student module helps students to create, view, and edit their tickets. Faculty module helps faculty to view the tickets generated by students in particular subjects dn allows them to answer the doubts. More than one faculty can answer a question and they will be displayed in the student module one after the other. Admin module validates the student and faculty registration and login. After successful registration a mail is sent to the user which consists of a link and the user needs to login into the website using that link. When a faculty answers a question, it will be displayed in the student module with the name of the faculty who answered the question. We can also get the information of the questions asked in each subject, questions asked by each student, and answers given by the faculty.

5.2 PROJECT MODULES:

This is the phase where we get an overview of our website. Implementation phase consists of modules, technical skills and functionalities. Module is a collection of source files and build settings that allow us to divide our project into discrete units of functionality. Importance of modules is that we can have better evaluation and more focused revision and improvement. Modifying a module is easier and more cost effective than revising a course or one large module. Technical Skills are the key requirement of a project, because we get the entire project development and enhancement from this field. Project needs to be developed in stages, hence the requirement of technical skills is also deferred from stage to stage. Functionalities is the use of knowledge a developer has making the user interface friendly and bug free. Support Hub consists of 3 modules. They are : Student module, Faculty module, Admin module. Each module has a set of files of source code which helps in better evaluation and efficient working of project

- **Admin:**

Admin module consists of admin login which consists of username and password. Admin has the right of validating the users who are registered. Since we are taking two-step verification, the second step takes place at the admin module. In this module admin has the right to approve or deny the users registered in the website. Admin also consists of the actions performed in websites like - questions asked by each student, answers given by each faculty etc.

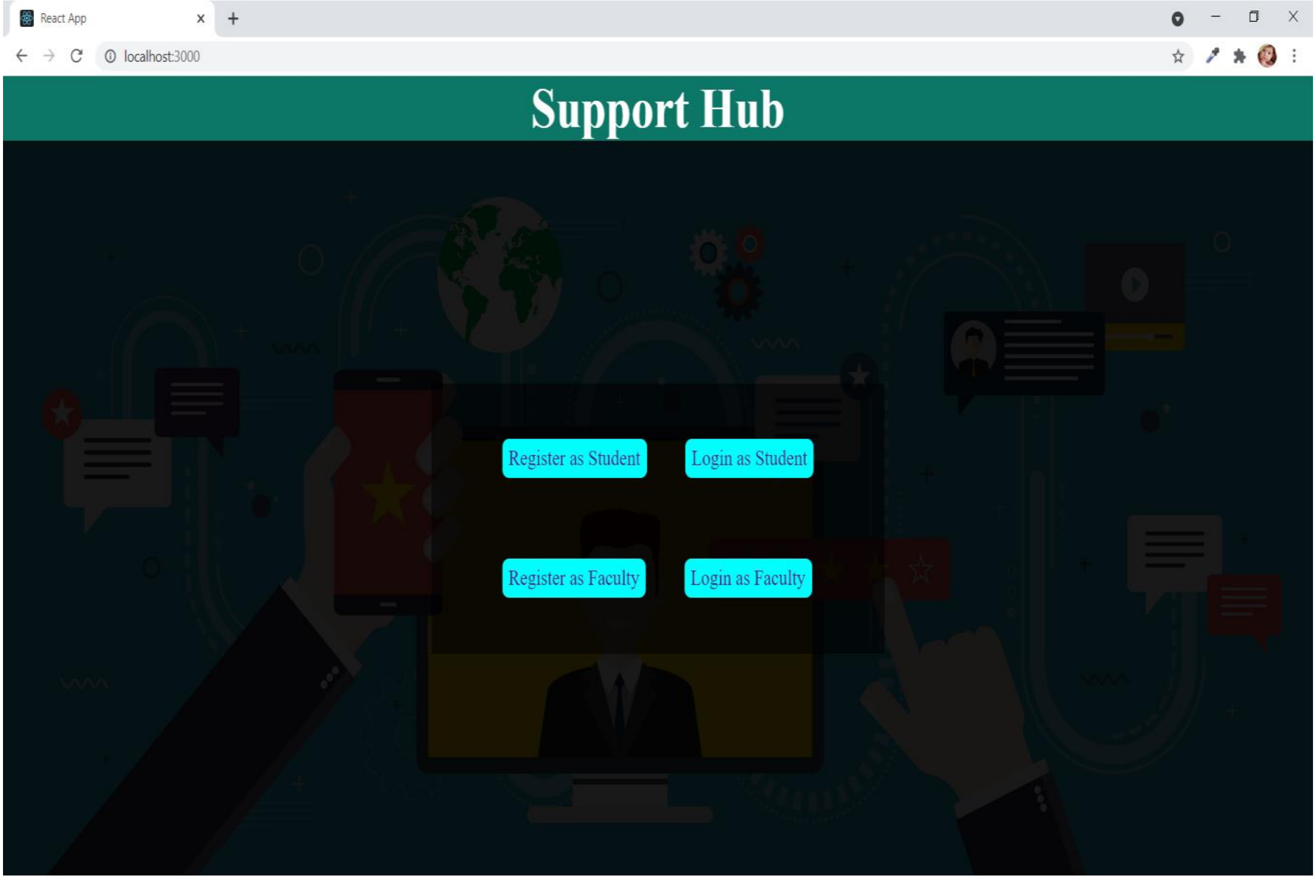
- **Student:**

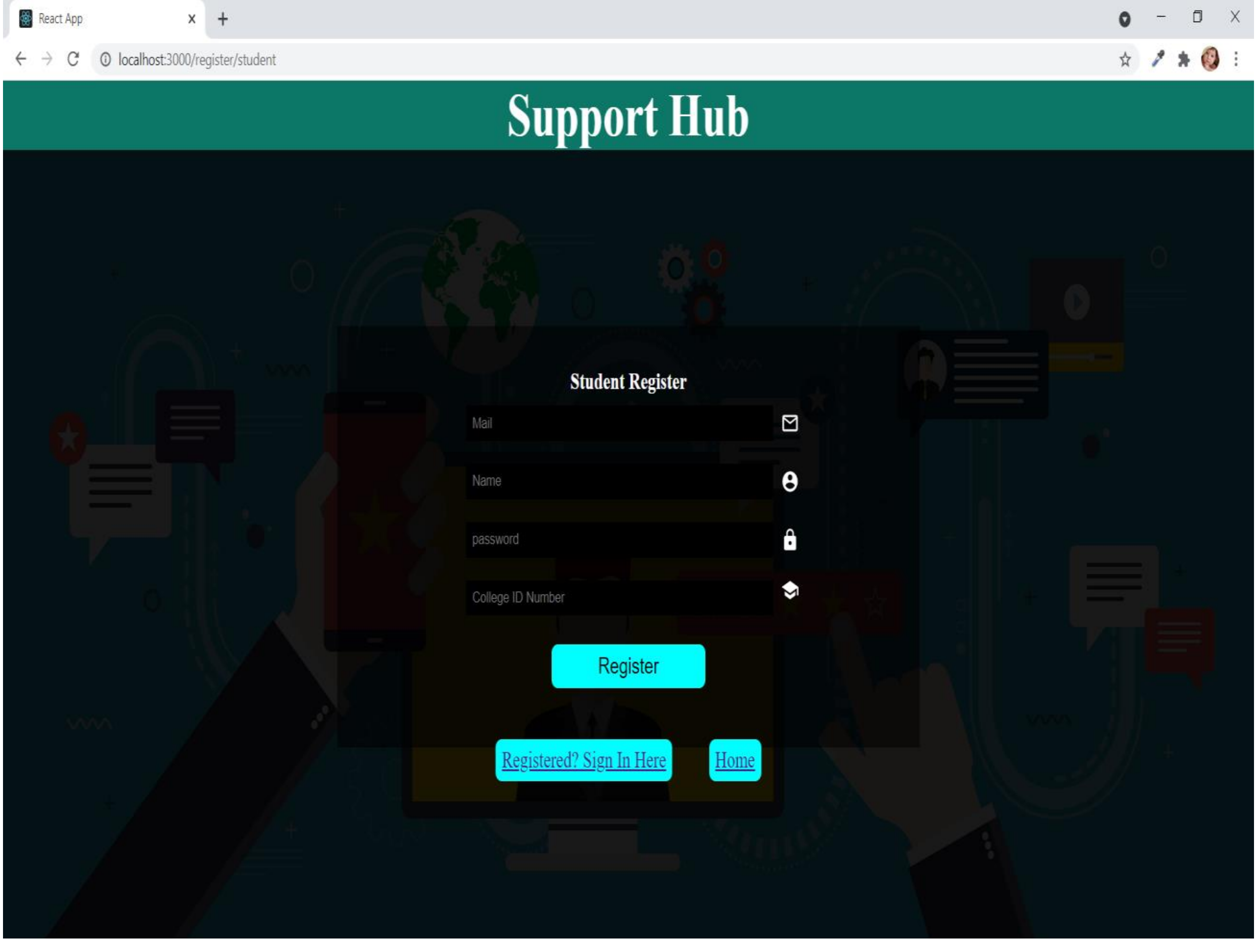
Student module consists of student registration, login, and students web portal. Students initially register in the web portal, thereafter login into the portal. When a student is logged into the portal then his/her data is stored in the database. Then for each other login, students are granted permission regarding the credentials in the database. Student module also consists of the design of the student portal. There would be a slight difference between student and faculty portal. Student portal consists of the details where ticket creation and tickets generated are shown. Apart from that, students are also provided with a dropdown where all the related subjects are displayed. Students need to pick the subject in which he/she has a doubt. After doubt is clarified then the student also has the option of deleting the ticket generated. If a student gets more than one response for the ticket generated, then they are appended on the student side with bullets and with the name of the faculty who answered the question. This increases readability and also makes the page look more neat and clean. Students can also view all tickets generated by him by clicking on the mytickets section.

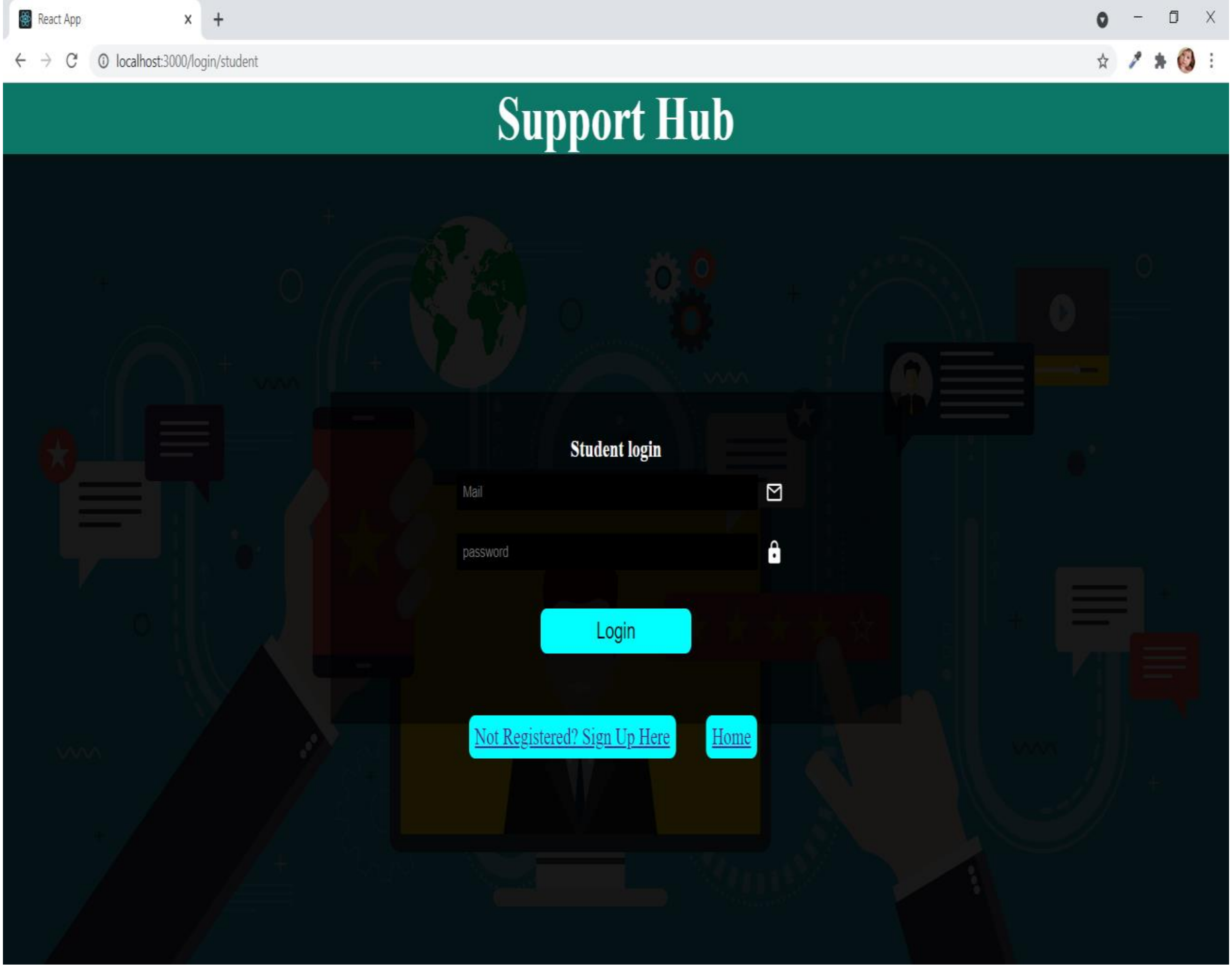
- **Faculty:**

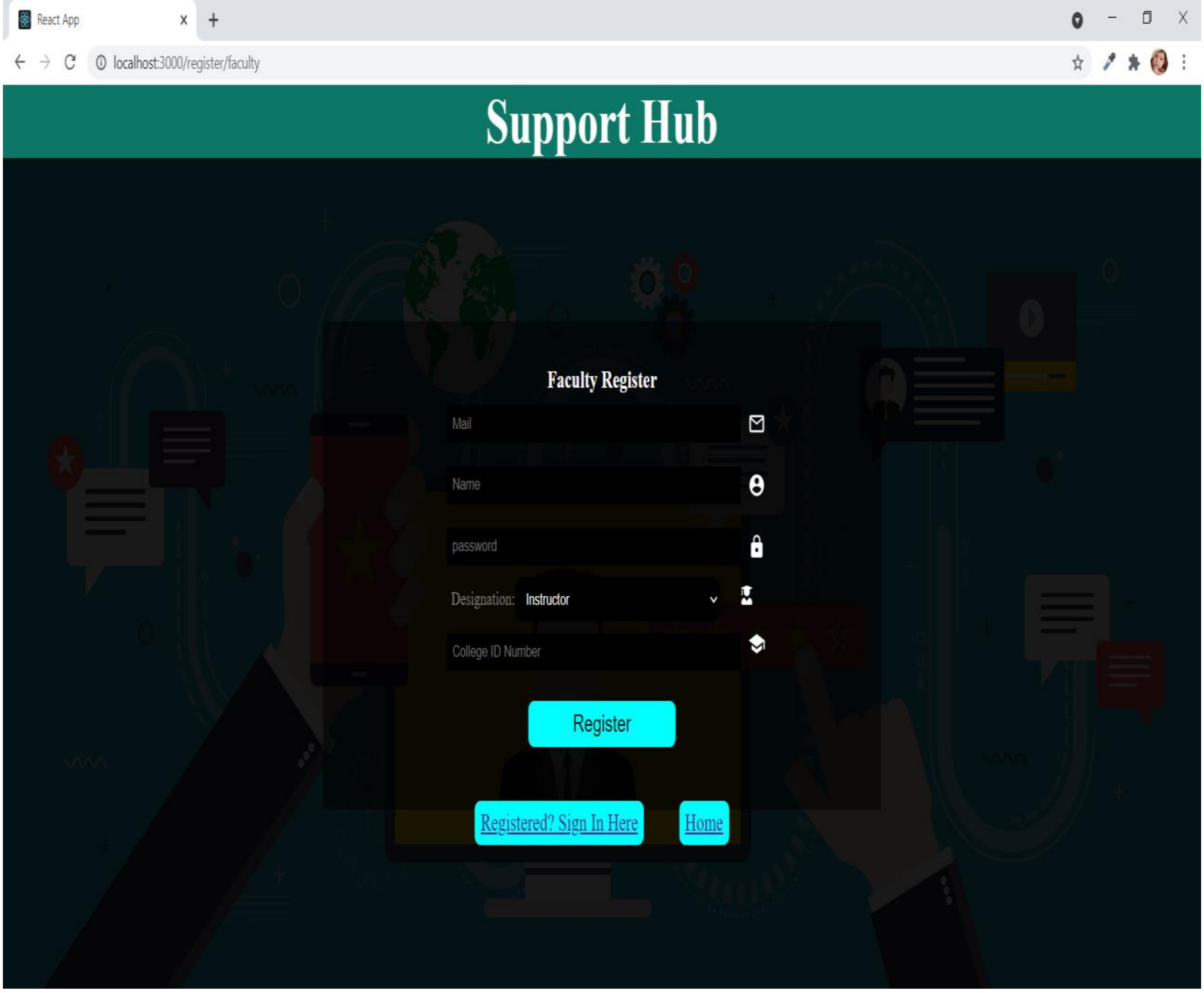
Faculty also follows the same authentication procedure as that of the students. Faculty should give their name, designation and email id to login into the website. The only difference will be the tickets. Faculty are provided the subjects in which they want to solve the doubts, then can choose any of them. Then doubts generated related to that subject are displayed on the faculty portal. Faculty can address the doubts of the students by typing their answer in the text field. If faculty want to rewrite or resubmit the answer that can also be done. All the faculty answers are displayed on the student side.

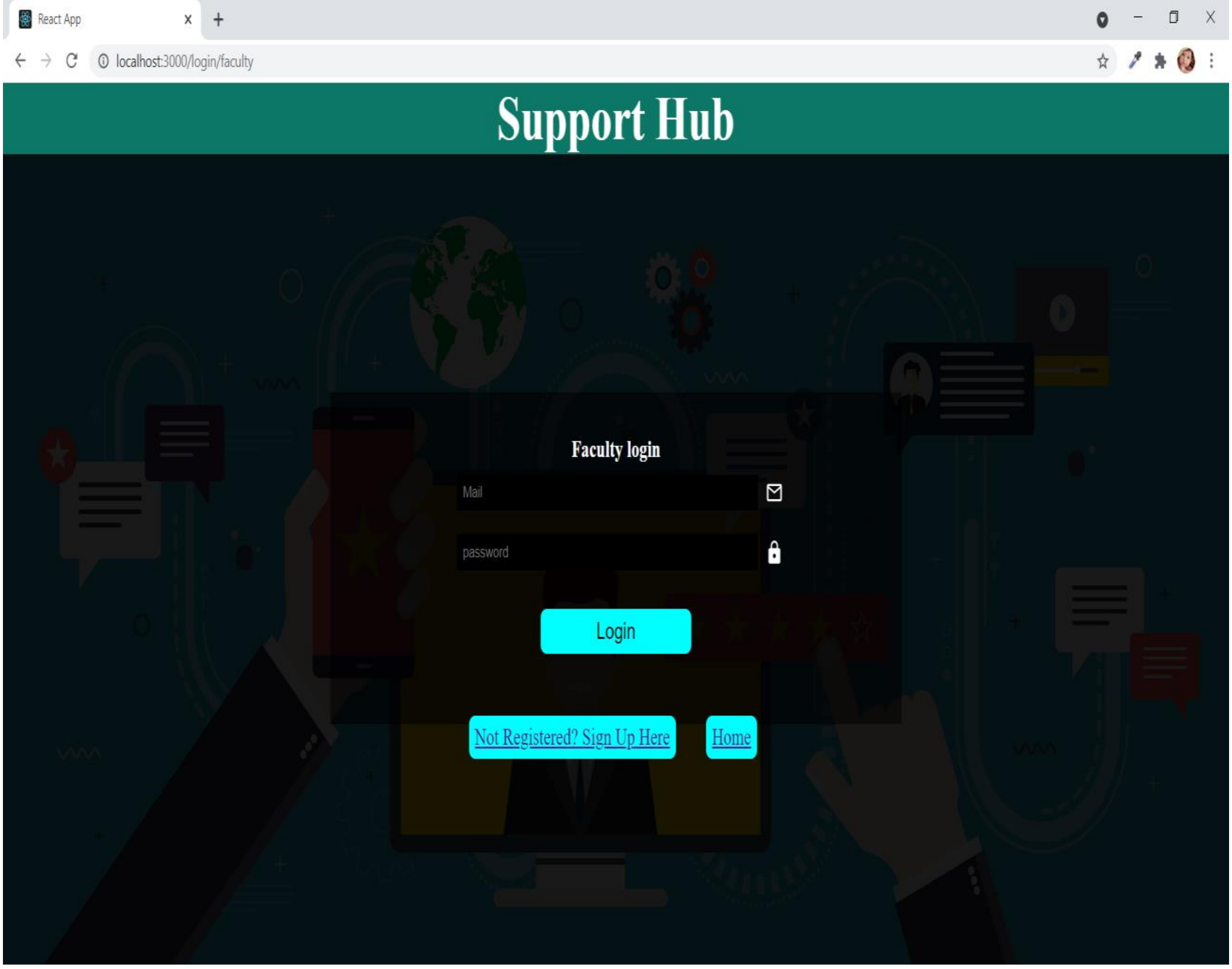
5.3 SCREENS:


React App
localhost:3000
Support Hub
Register as Student
Login as Student
Register as Faculty
Login as Faculty

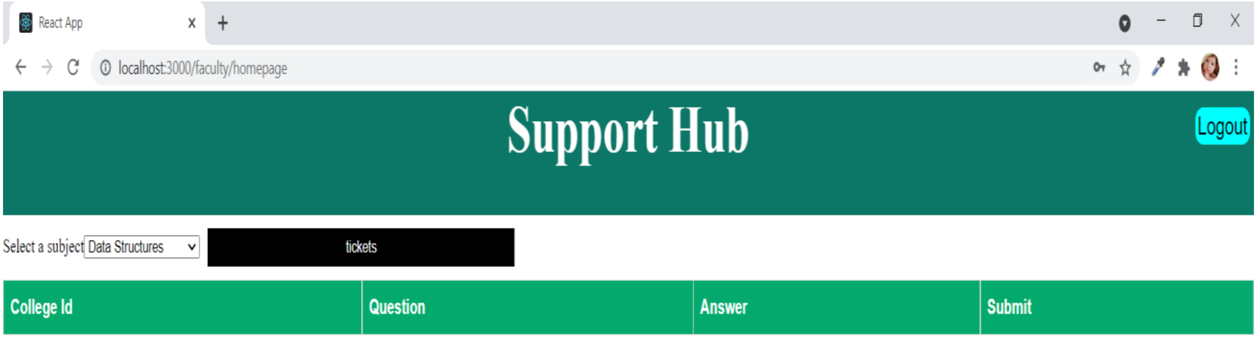

React App
localhost:3000/register/student
Support Hub
Student Register
Mail
Name
password
College ID Number
Register
Registered? Sign In Here
Home


React App
localhost:3000/login/student
Support Hub
Student login
Mail
password
Login
Not Registered? Sign Up Here
Home

React App
localhost:3000/register/faculty
Support Hub
Faculty Register
Mail
Name
password
Designation: Instructor
College ID Number
Register
Registered? Sign In Here
Home



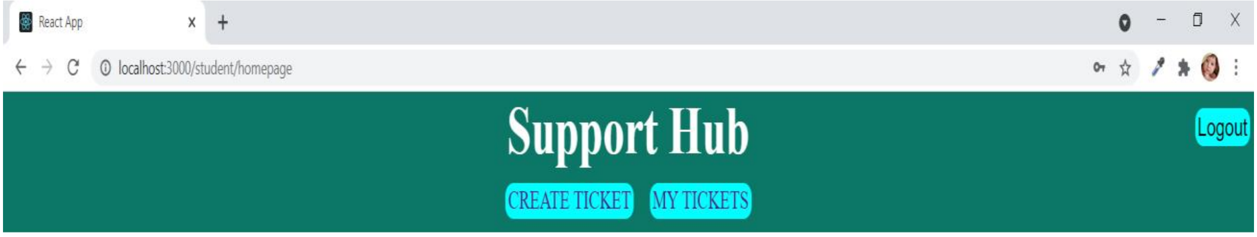
React App
localhost:3000/login/faculty
Support Hub
Faculty login
Mail
password
Login
Not Registered? Sign Up Here
Home



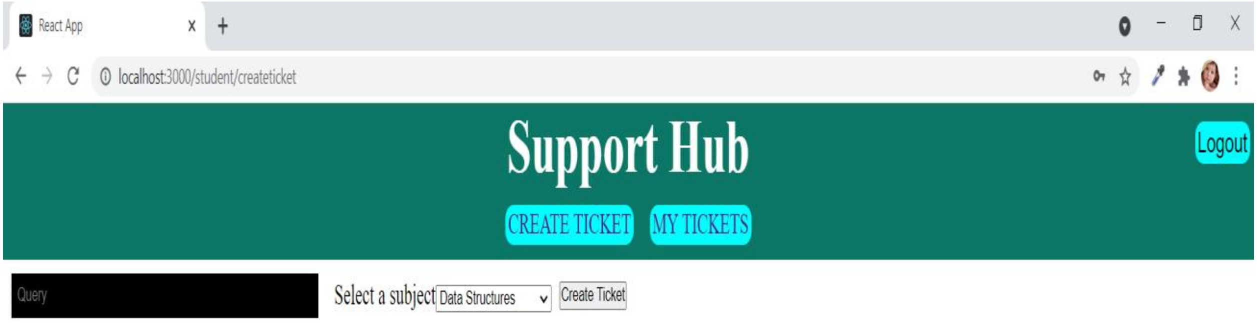
Support Hub

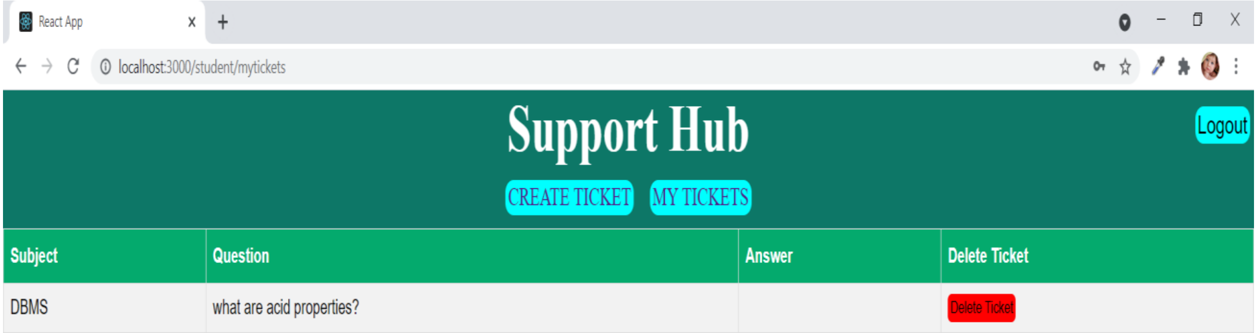
Select a subject Data Structures

College Id	Question	Answer	Submit

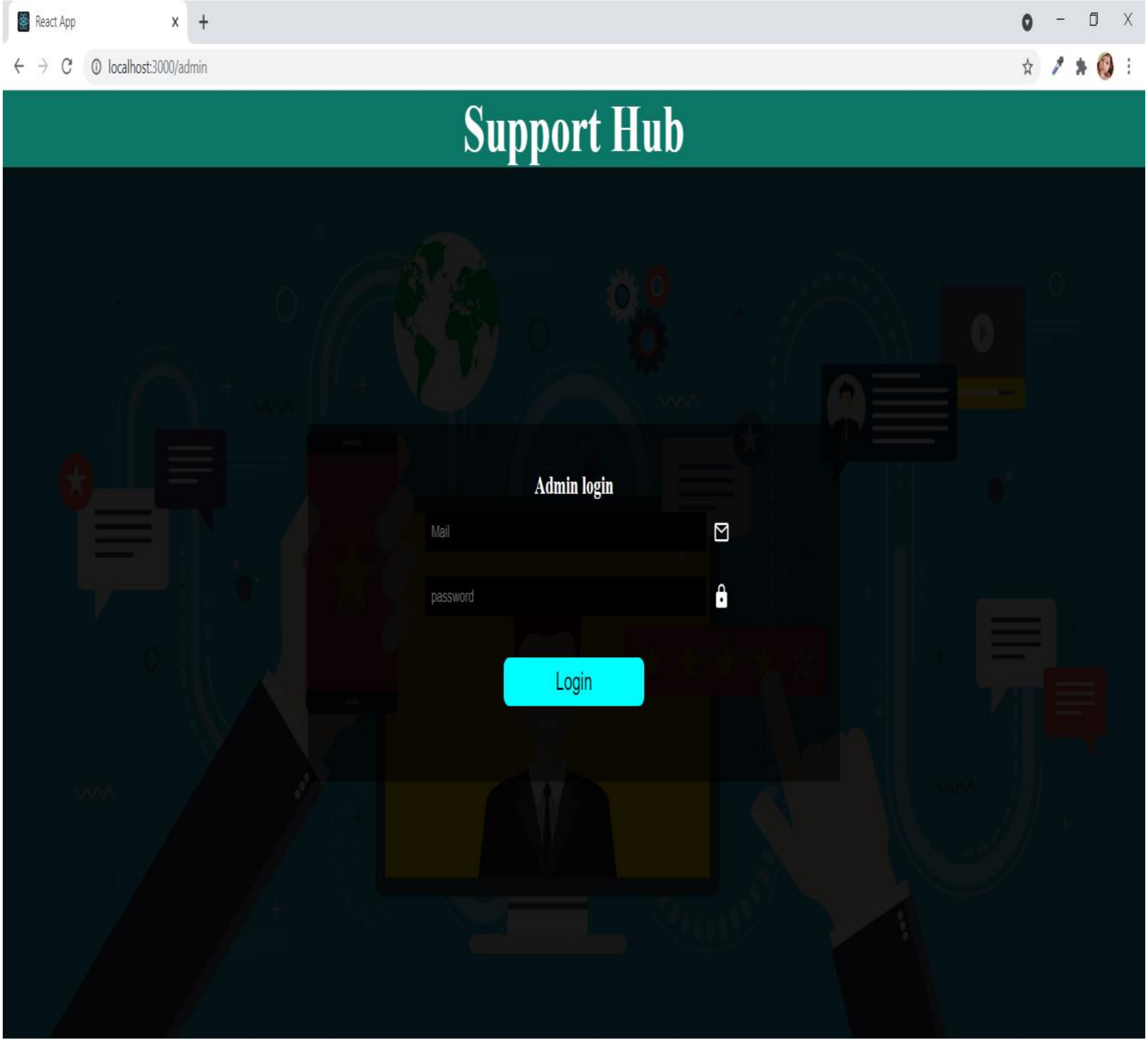


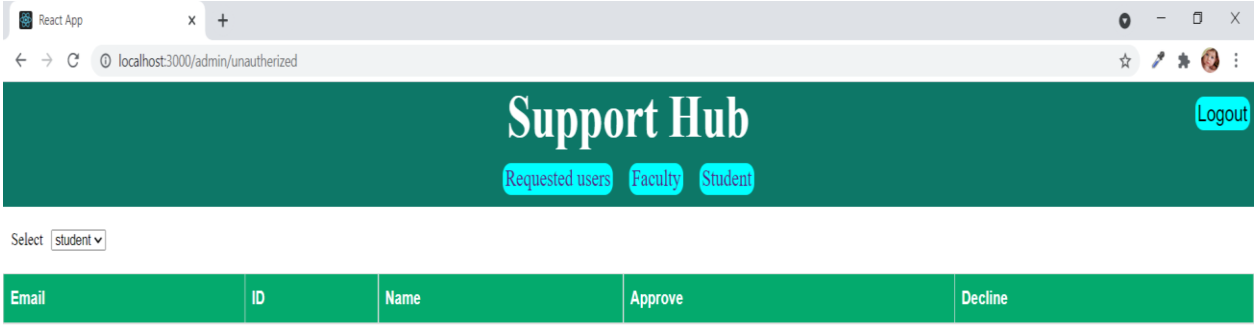
React App
localhost:3000/student/homepage
Support Hub
Logout
CREATE TICKET
MY TICKETS

React App
localhost:3000/student/createticket
Support Hub
Logout
CREATE TICKET
MY TICKETS
Query
Select a subject
Data Structures
Create Ticket



Subject	Question	Answer	Delete Ticket
DBMS	what are acid properties?		Delete Ticket

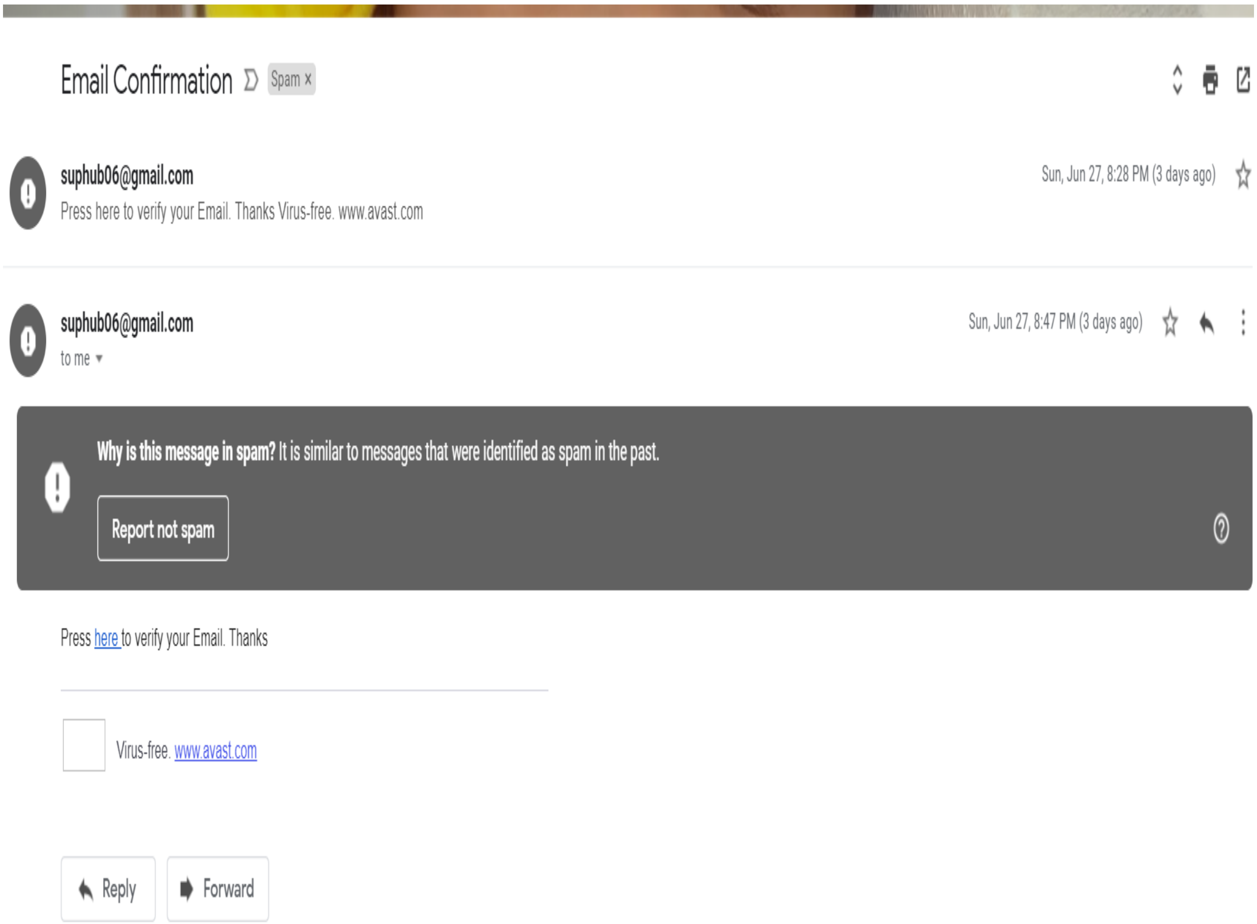
React App
localhost:3000/admin
Support Hub
Admin login
Mail
password
Login



Support Hub

Select student

Email	ID	Name	Approve	Decline



Email Confirmation

suphub06@gmail.com

Press here to verify your Email. Thanks Virus-free. www.avast.com

Sun, Jun 27, 8:28 PM (3 days ago)

suphub06@gmail.com

to me

Sun, Jun 27, 8:47 PM (3 days ago)

Why is this message in spam? It is similar to messages that were identified as spam in the past.

Press here to verify your Email. Thanks

Virus-free. www.avast.com

SYSTEM TESTING

6. SYSTEM TESTING

6.1 INTRODUCTION:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TESTING METHODS:

The following are the Testing Methodologies:

- **White box Testing**
- **Black box Testing**
- **Unit Testing**
- **Integration Testing**

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1.Top-Down Integration:

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration:

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

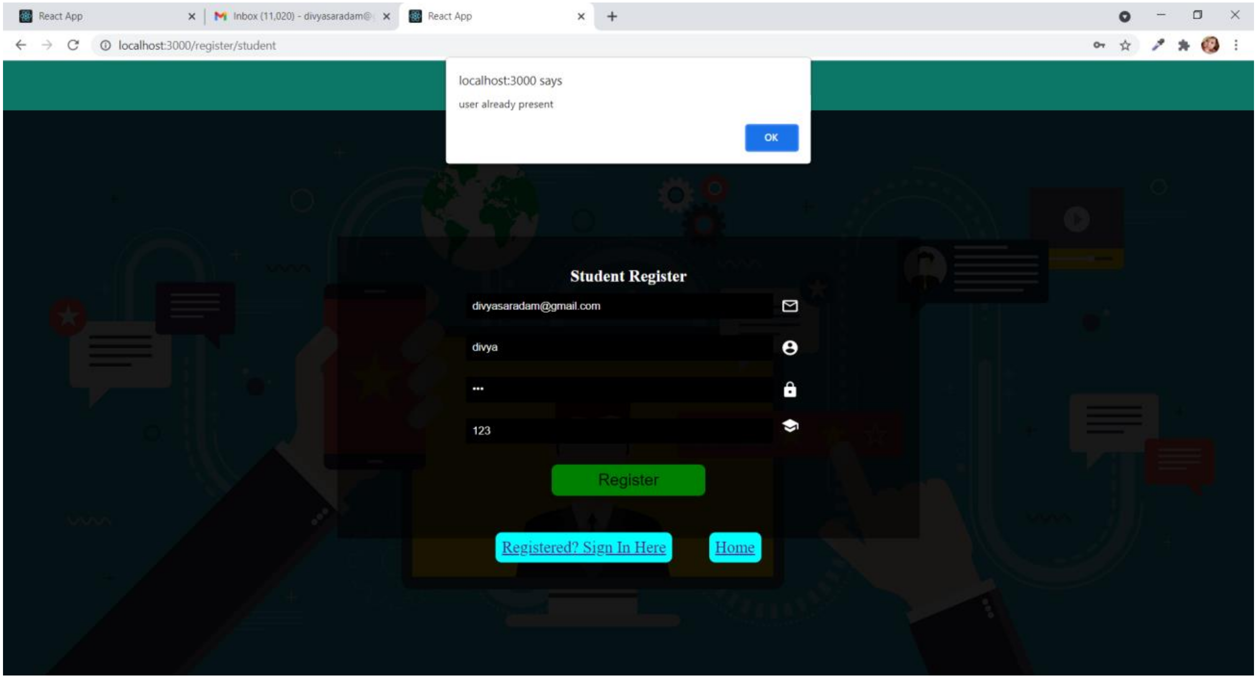
- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

6.3 TEST CASES:

Table 6.1: TEST CASE FOR REGISTRATION

Test Case1: User Registration	
Test Objective: To check whether a user is indeed a valid e-mail and is successfully registered or not.	
Test Description:	
Requirement Verified: Yes	
Test Environment: System should require internet.	
Test Setup/Pre-conditions:	
Actions	**Expected Results**
User enters details and clicks submit	Registration successful.
Pass: Yes	**Fail:** No
Problem/Issues: None	
Notes: Successfully tested and executed.	



localhost:3000/register/student

localhost:3000 says

user already present

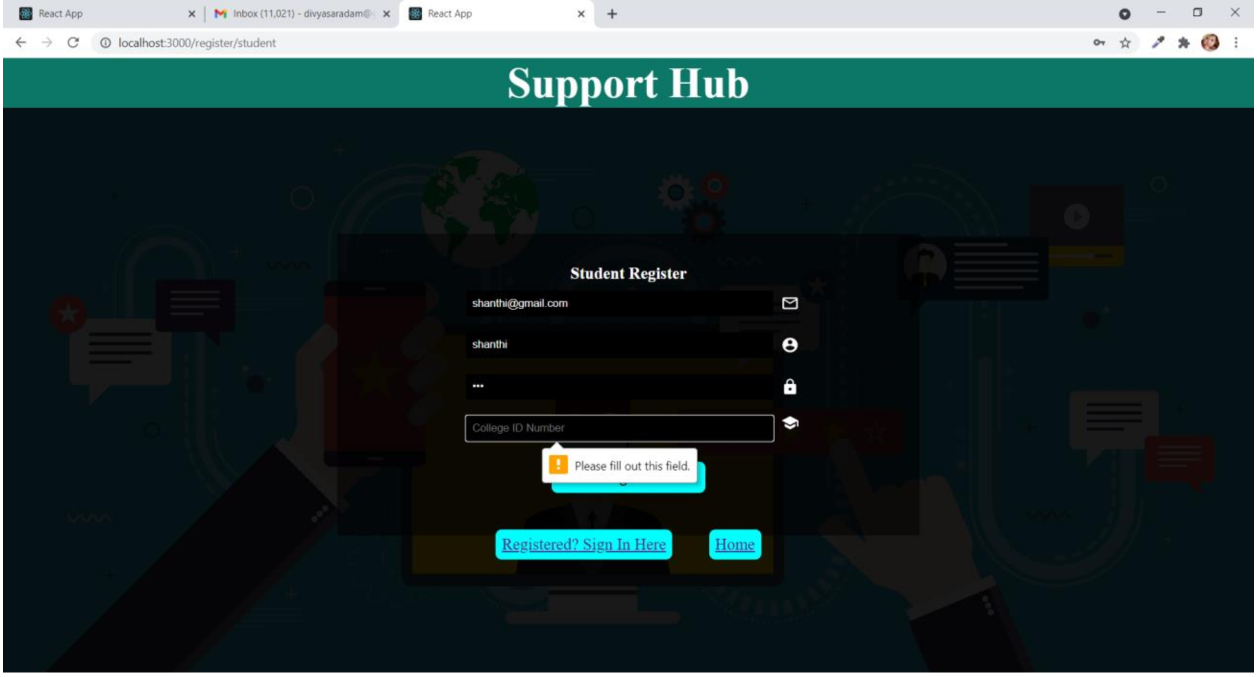
Student Register

divyasaradam@gmail.com

divya

•••

123



React App
Inbox (11,021) - divyasaradam@
React App
localhost:3000/register/student

Support Hub

Student Register

shanthi@gmail.com

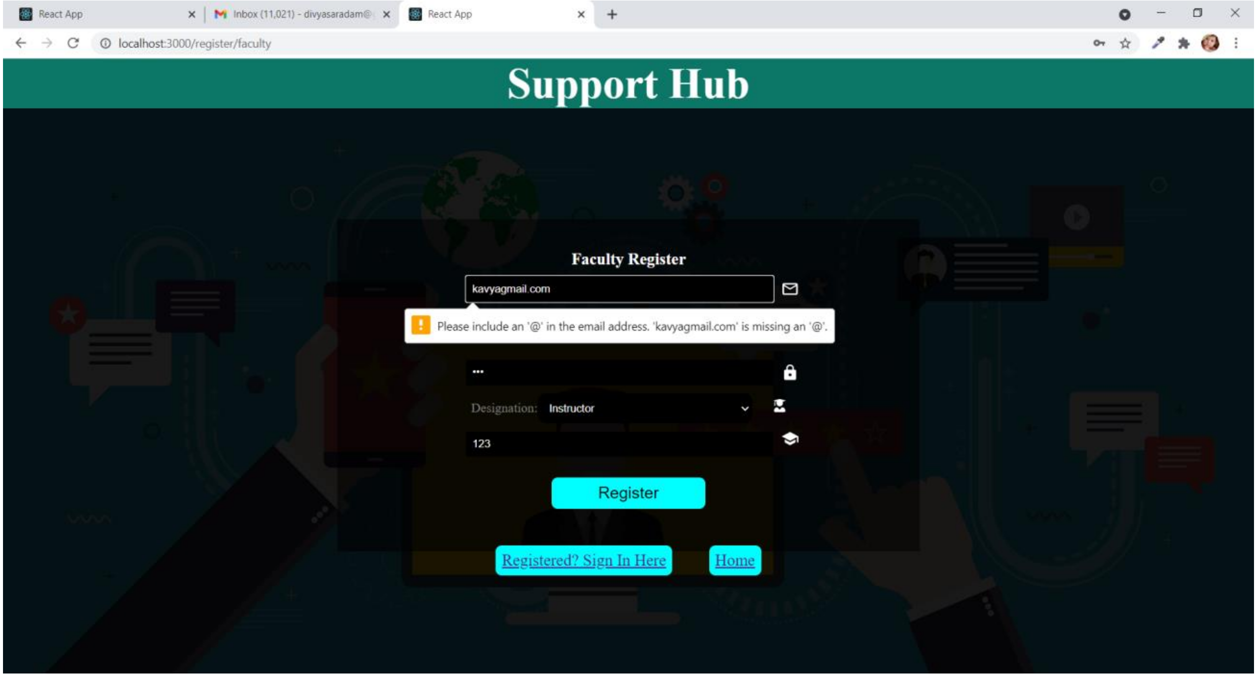
shanthi

•••

College ID Number

Please fill out this field.

Registered? Sign In Here
Home



React App
Inbox (11,021) - divyasaradam@
React App
localhost:3000/register/faculty
Support Hub
Faculty Register

kavyagmail.com

Please include an '@' in the email address. 'kavyagmail.com' is missing an '@'.

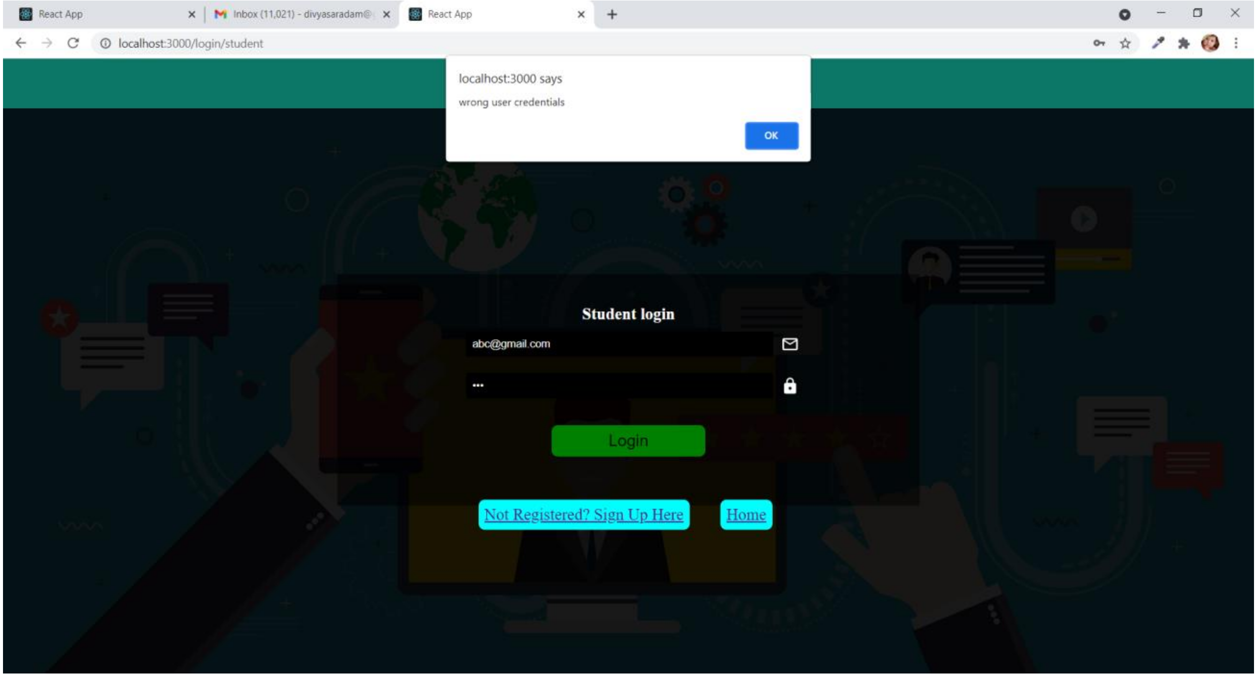
•••

Designation: Instructor

123

Table 6.2: TEST CASE FOR LOGIN

Test Case1: User Login	
Test Objective: To check whether a user is valid or not.	
Test Description: The user enters username and password. If password and username are valid, and if the user completes two-step verification then the user will be successfully logged in.	
Requirement Verified: Yes	
Test Environment: System should require internet.	
Test Setup/Pre-conditions:	
Actions	**Expected Results**
User enters details and clicks submit	Registration successful.
Pass: Yes	**Fail:** No
Problem/Issues: None	
Notes: Successfully tested and executed.	



React App
Inbox (11,021) - divyasaradam@
React App
localhost:3000/login/student

localhost:3000 says

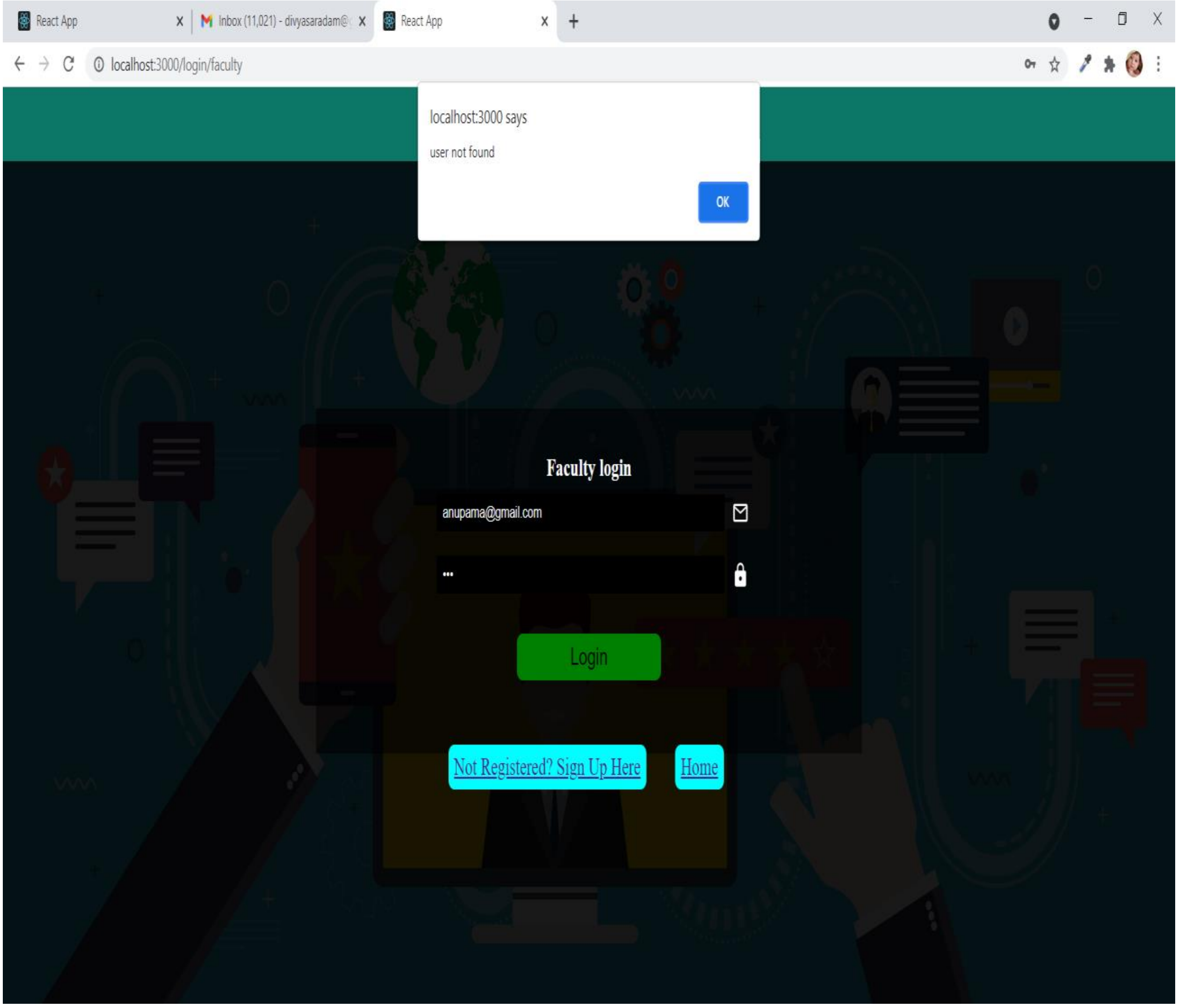
wrong user credentials

Student login

abc@gmail.com

•••

Login
Not Registered? Sign Up Here
Home



React App
Inbox (11,021) - divyasaradam@
React App
localhost:3000/login/faculty

localhost:3000 says

user not found

Faculty login

anupama@gmail.com

•••

Login
Not Registered? Sign Up Here
Home

Table 6.3: TEST CASE FOR TICKET GENERATION

Test Case1: Ticket Generation	
Test Objective: To check whether a ticket is successfully created or not.	
Test Description: The student selects subjects and enters their doubt. If a ticket is successfully generated then an alert box is generated.	
Requirement Verified: Yes	
Test Environment: System should require internet.	
Test Setup/Pre-conditions:	
Actions	**Expected Results**
Student selects subjects and enters their doubt and clicks submit.	Ticket generated successfully
Pass: Yes	**Fail:** No
Problem/Issues: None	
Notes: Successfully tested and executed.	

Table 6.4: TEST CASE FOR SUBMITTING ANSWERING

Test Case1: Submitting Answers	
Test Objective: To check whether an answer is successfully submitted or not.	
Test Description: The faculty selects subjects and views doubts. If faculty wants to answer a particular doubt, they will enter the answer and press the submit button. If the answer is submitted successfully then an alert box is generated.	
Requirement Verified: Yes	
Test Environment: System should require internet.	
Test Setup/Pre-conditions:	
Actions	**Expected Results**
Faculty selects subjects and enter the answer and clicks submit.	Answer submitted successfully
Pass: Yes	**Fail:** No
Problem/Issues: None	
Notes: Successfully tested and executed.	

CONCLUSION

7. CONCLUSION

Finally in this website, students and faculty are allowed to login and clarify their doubts. Basically when there is any doubt which needs any assistance we can generate a ticket and that tickets are stored by subject classification. When a faculty wants to deal with any other subject then faculty will choose the particular subject in the drop down. On selecting the subject all the doubts present in the particular subject are displayed to the faculty with respective college ID's of students. Any faculty can answer the student's question until and unless the student deletes the ticket. If there are more than one faculty addressing a particular doubt then in the students view, the answers of all the faculty are appended to the answer field. Answers are appended with the faculty name in the answer field. If a student who raised a doubt is satisfied with a particular answer then he/she can delete the ticket where it disappears from both student and faculty modules. All the details regarding the number of questions asked in a subject, or number of questions asked by one student are maintained in the admin module. Admin module is where all the authentications are done. Simply using the Support Hub enhances faculty and students to explore and solve doubts in a more professional way.

BIBLIOGRAPHY

8:BIBLIOGRAPHY

[1]https://www.npmjs.com/package/nodemailer

[2]https://www.npmjs.com/package/axios

[3]https://www.npmjs.com/package/react-router-dom

[4]https://material-ui.com/components/icons/

[5]https://nodejs.dev/learn/make-an-http-post-request-using-nodejs

[6]https://reactjs.org/docs/hooks-effect.html

[7]https://reactjs.org/docs/hooks-state.html

[8]https://docs.atlas.mongodb.com/

[9]https://jwt.io/

[10]https://expressjs.com/

[11]https://www.npmjs.com/package/bcryptjs

[12]https://www.npmjs.com/package/dotenv