Mini Project Report

Projects Management System for RGUKT Basar

Submitted by

Aslam SK (ID: B121407)
Harish Kumar D (ID: B121465)
Bhagvan G (ID: B121582)
Rakesh k (ID: B121303)
Ramakrishna (ID: B121532)

Under Supervision of:
Mr. Samit Kumar Pradhan
Mr. K. Parshuram

Department of Computer Science & Engineering
Rajiv Gandhi University of Knowledge Technologies
Basar-504107, INDIA

April, 2017

Certificate

It is certified that the work contained in this report titled “Projects Management System for RGUKT Basar” is the original work done by **Aslam SK** (ID: B121407)**,Harish Kumar D** (ID: B121465),**Bhagvan G** (ID: B121582),**Rakesh k** (ID: B121303),**Ramakrishna** (ID: B121532) and has been carried out under our supervision.

_____

Mr. Samit K. Pradhan
Assistant Professor
Department of CSE
RGUKT, Bassar

_____

Mr. K. Parshuram
Lab Instructor
Department of CSE
RGUKT, Basar

Date:14-4-2017

Acknowledgement

We would like to express our deepest appreciation to all those who provided us the possibility to complete this project. A special gratitude we give to our project guide, **Mr. Samit K. Pradhan**,Assistant Professor in Computer science department whose contribution in stimulating suggestions and encouragement, helped us to coordinate the project and our sincere thanks to **Mr. K. Parsuram**, Lab Instructor who constantly motivate us to do this project and make it successful.

We are also grateful to respected **Mr. Ranjith Kumar HOD (CSE)** for permitting us to utilize all the necessary facilities of the institution.

Lastly, we would like to express our deep gratitude towards classmates and our indebtedness to our parents for providing us the moral support and encouragement.

Aslam SK (ID: B121407)
Harish Kumar D (ID: B121465)
Bhagvan G (ID: B121582)
Rakesh k (ID: B121303)
Ramakrishna (ID: B121532)

Abstract

The main objective of this project is to develop a system which manages all the projects details of our campus students and it provides a complete environment to the students to demonstrate their projects

This Projects Management System would store the details of all the projects of students like Academic projects,Techfest projects,Internship projects etc. As well as other students can upvote and comment the projects top projects and recent projects would be displayed in the website based on rankings.

Contents

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

This Projects Management System would store the details of all the projects of students like Academic projects,Techfest projects,Internship projects etc. As well as other students can upvote and comment the projects Top projects and recent projects would be displayed in the website based on rankings.

To reduce the of complexities we find a new solution that is ProjectHub,basically in this project is to facilitate students in the campus to communicate with each other regarding Project related queries. Any one can answer to solve the problems while doing projects with a valid user login.

CHAPTER 2

SoftwareRequirementSpecification

1. Introduction

1.1 Purpose

The main objective of this project is to develop a system which manages all the projects details of our campus students and it provides a complete environment to the students to demonstrate their projects.

1.2 Document Conventions

Heading:	Font Size: **18** Font Style : Bold Font : Times new Roman
Sub Heading:	Font Size: **16** Font Style : Bold Font : Times new Roman
Content:	Font Size: **14** Font : Times new Roman

1.3 Intended Audience and Reading Suggestions

The audience precisely will be:-

1. All the Students in the campus
2. The software developers(Administrators)

Notification section will be helpful to students to get notified and plan accordingly.

1.4 Product Scope

This Projects Management System would store the details of all the projects of students like Academic projects,Techfest projects,Internship projects etc. As well as other students can upvote and

comment the projects Top projects and recent projects would be displayed in the website based on rankings.

2. Overall Description

This project is to facilitate students in the campus to communicate with each other regarding Project related queries. Any one can answer to solve the problems while doing projects with a valid user login.

2.1 Product Perspective

This system allows projects from various streams of Engineering where all the projects may not have the same type of description.

So this projects management system would maintain different categories for projects from different streams, which also make users to search more efficiently to save time and effort.

2.2 Product Functions

● The System would provide the facility of viewing details of all the projects.
● It allows users to encourage the projects by upvoting and comments.
● Users can propose for new projects
● Allows users to ask and answers the queries regarding projects.

2.3 User Classes and Characteristics

The major User classes in the System would be :

i) Student

- Student need to login with his/her username and password if they want to share their project details or ask questions.
- Students can see any project details without login to the site but they can not ask, answer, upvote and comment.

ii) Administrator

- The Admin has the supreme power of the application.
- Admin provides approval to the Student questions.
- Admin is responsible for maintaining and updating the whole system.

2.4 Operating Environment

This web application can be deployed on linux or windows machine with Apache Server and MySQL server.

This application can be accessed by user through a machine having any web browser with html javascript support. The client devices must preferably have browsers like IE9 or above, Mozilla firefox (version 3.5 or above) or Opera 10 or chrome (version 29 or above) or safari installed in their OS. Specified versions are preferred to get HTML 5 output.

2.5 Design and Implementation Constraints

- User system should be connected to LAN.
- User should install any one of the above specified web browsers.

2.6 Assumptions and Dependencies

- We are assuming that the user should have some basic knowledge of computer.
- Students should be from any branch or any year.

3. External Interface Requirements

3.1 User Interfaces

It is specially designed for students. The home page allows the students

login into the website and see the current notifications going in the campus placement section. The menu list shows the functions that students can do.

The user can click on any one of the options and is taken to the screen of their choice. Students can only use the functions after logging into the website.

3.2 Hardware Interfaces

The program will communicate with hard drive (the filesystem and database) via the appropriate PHP code. The user can communicate through browser using keyboard and a display through graphical interface displayed on user's screen.

3.3 Software Interfaces

The software interface uses a LAMP (Linux, Apache, MySQL and PHP) set-up.

The product will host a local Apache web server where the user interface will be displayed via a web browser. PHP will also be used to create background colors, border colors and text display.

2.7 Communications Interfaces

The ProjectHub shall use the HTTP protocol for communication over the Local Area Network(LAN) only. Students can access this software through a web browser in their system that is connected to LAN.

4. Specific Requirements

4.1 Functionality

i.Login

User can login into the website with a valid username and password.

Use case name	Login
Actor	Student, Faculty.
Brief Description	This use case allows the user log into the ProjectHub Website and do some functions of the system.
Goals	To login the system or website.
Triggers	User accesses the ProjectHub Website.
Pre-Condition	User should be in the homepage that display username and password to login.
Post-Condition	User can ask questions or post answers, view the past questions or answers and can alow upvote and comment.
Basic Flow	User accesses this sytem website. User gives the details and click on the login button.
Alternative Flow	None.
Exception	The login is abandoned if user gives wrong details.
Quality	Login takes very less time to load.

ii.Search

User can search for particular projects in the projectHub.

Use case name	Search
Actor	Student, Faculty.
Brief Description	This use case allows the user to search for any project regarding any stream of engineering in the campus and choose their required one.
Goals	To search for the projects in the website
Triggers	User would click search button.
Pre-Condition	Any one can search without even logged in.
Post-Condition	User can view the interested projects.
Basic Flow	User opens this website. Enter the search query User clicks on the search. User search his/her interested projects and view the details.
Alternative Flow	User can search the interested projects.
Exception	The user may abandon the operation at any time.
Quality	User can filter within 5 seconds of time.

iii)Upload

Users can upload their porjects details they can also upload a video which explain the project.

Use case name	Upload
Actor	Student.
Brief Description	This use case allows user to upload the project details.
Goals	To upload the details of the project in the website
Triggers	User would click upload button.
Pre-Condition	User should be logged in.
Post-Condition	User will get notified on successful submission.
Basic Flow	User opens this website. User logs into this website. User clicks on the upload. User fill the required details User would upload by a button.
Alternative Flow	None.
Exception	The user may abandon the operation at any time.
Quality	Takes time based on the size of details

iv) Ask Question

User can ask their questions and clarify their doubts regarding anything.

Use case name	Ask Question
Actor	Student
Brief Description	This use case allows the user to ask any question regarding projects and clarify the doubts.
Goals	To post the questions in the website
Triggers	User would click ask a question button.
Pre-Condition	User should be logged in.
Post-Condition	User questions are posted in the website.
Basic Flow	User opens this website. User logs into this website. User clicks on the ask a query. User writes his/her query and submits the question.
Alternative Flow	None.

v) Post Answer

User can post answers to any question which have been posted in the past.

Use case name	Answer a question
Actor	Student, Faculty.
Brief Description	This use case allows the user to answer any question regarding projects and share their knowledge.
Goals	To post the answers in the website
Triggers	User would click answer button.
Pre-Condition	User should be logged in.
Post-Condition	User answers are posted in the website.
Basic Flow	User opens this website. User logs into this website. User view the questions User clicks on the answer. User writes his/her answer and submits the answer.
Alternative Flow	None.
Exception	The user may abandon the operation at any time or if the answer is unnecessary it will be canceled.
Quality	User can submit the answer within 2 seconds of time.

Vi) Upvote

Users can appreciate other users by upvoting on their projects.

Use case name	Upvote
Actor	Student, Faculty
Brief Description	This use case allows the user to to appreciate other's projects
Goals	To give an upvote
Triggers	User would click upvote button.
Pre-Condition	User should be logged in.
Post-Condition	Project upvotes would updated in the website.
Basic Flow	User opens this website. User logs into this website. User Views the projects. User upvote the project.
Alternative Flow	None.
Exception	Cancel the upvote by clicking on the button again

Vii) Comment

Users can comment on the projects by either suggesting somthing to improve or by encouraging.

Use case name	Comment
Actor	Student, Faculty
Brief Description	This use case allows the user to to appreciate or give feedback on other's projects
Goals	To comment
Trigers	User would click post comment button.
Pre-Condition	User should be logged in.
Post-Condition	Project comments would updated in the website.
Basic Flow	User opens this website. User logs into this website. User Views the projects. User Comment the project.
Alternative Flow	None.
Exception	Project owners can disable comment.

Viii) Profile

User can view thier profile which consists of thier activities in the website.

Use case name	Profile
Actor	Student, Faculty
Brief Description	This use case allows the user to to view their user details
Goals	To view their profile
Triggers	User would click Myprofile link.
Pre-Condition	User should be logged in.
Post-Condition	Profile details.
Basic Flow	User opens this website. User logs into this website. User clik on myprofile.
Alternative Flow	None.

- **Add New Information**
  - Adminstrators can add the new company details.
  - Adminstrators can extend the existed company details.
  - Adminstrators can add the new exam papers of any company.

4.2 Usability

- The software must have a simple and User friendly Interface.
- The navigation to various pages should make it more convenient to the users so as to save time and confusion.

4.3 Reliability

- The system shall provide 100% access reliability.
- The system shall generate error messages when the user attempts to enter invalid data or ask wrong questions.

4.4 Performances

- The users must get the response within seconds i.e. the response time of a particular function should be minimum.
- The system would exhibit high performance because it would be well optimized.

3. Other Nonfunctional Requirements

3.1 Performance Requirements

The Online temple management system that we are going to develop will be used

to solve the all requirements of devotees. It also provides the Guide and Transportation System for the devotees. It will solve the problems of the devotees.

3.2 Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

3.3 Security Requirements

We are going to develop a secured database for the devotees. There are different categories of users namely administrator, devotees etc. Depending upon the category of user the access rights are developed. The devotees who are login into the web page and the user submitted bank account details should belongs to the one user only. Administrator can be able to update the data and delete the record etc.

3.4 Software Quality Attributes

The quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database.

Other Requirements Other requirements includes maintaining devotee information, managing pooja schedule, managing donations, generating various types of reports for temple management like priest scheduling report, donation history report and year donation summary report.

CHAPTER 3

DETAILED DESIGN DOCUMENT

1.0 Introduction

- This section provides an overview of the entire design document. This document describes all data, architectural, interface and component-level design for the software.

1.1. Goals and Objectives

- This software will be useful for RGUKT. Through this application students can demonstrate their projects.

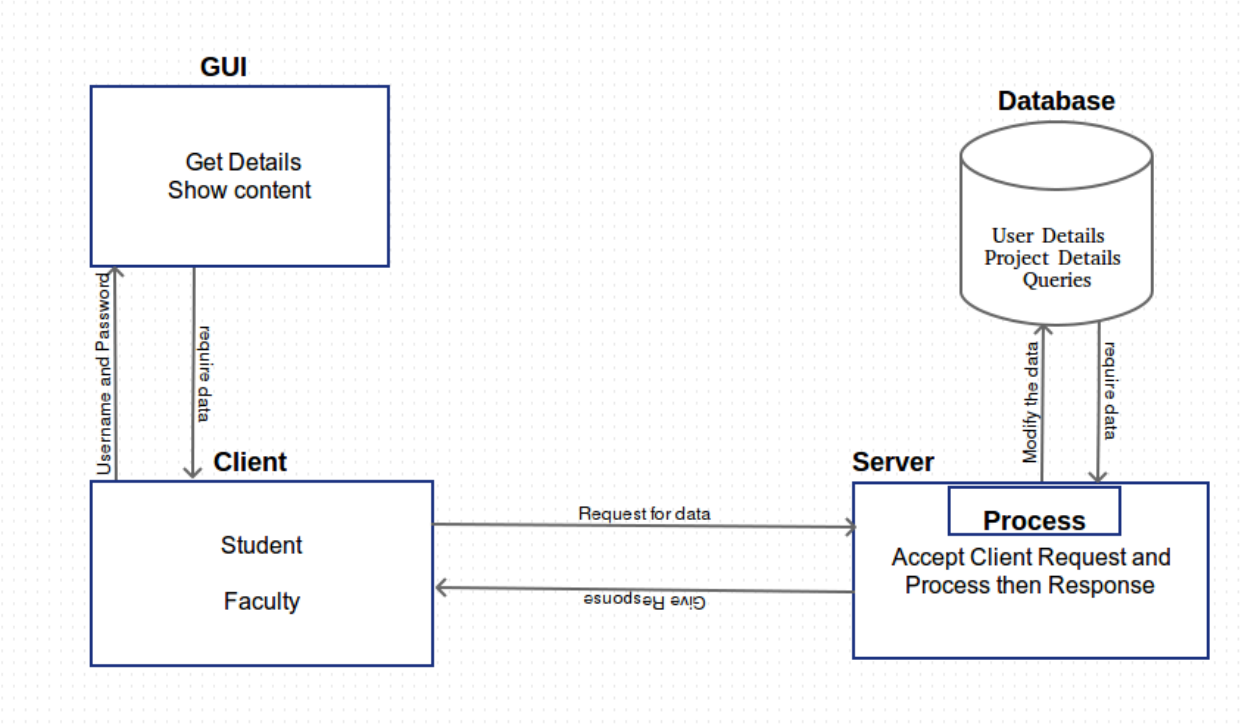
1.2. Scope of Solution

This Project Management System would store the details of all the projects of students like Academic projects, Techfest projects, Internship projects etc. As well as other students can upvote and comment the project.

Top projects and recent projects would be displayed in the website based on rankings.

2.0 Architecture Overview

2.1 Architecture diagram

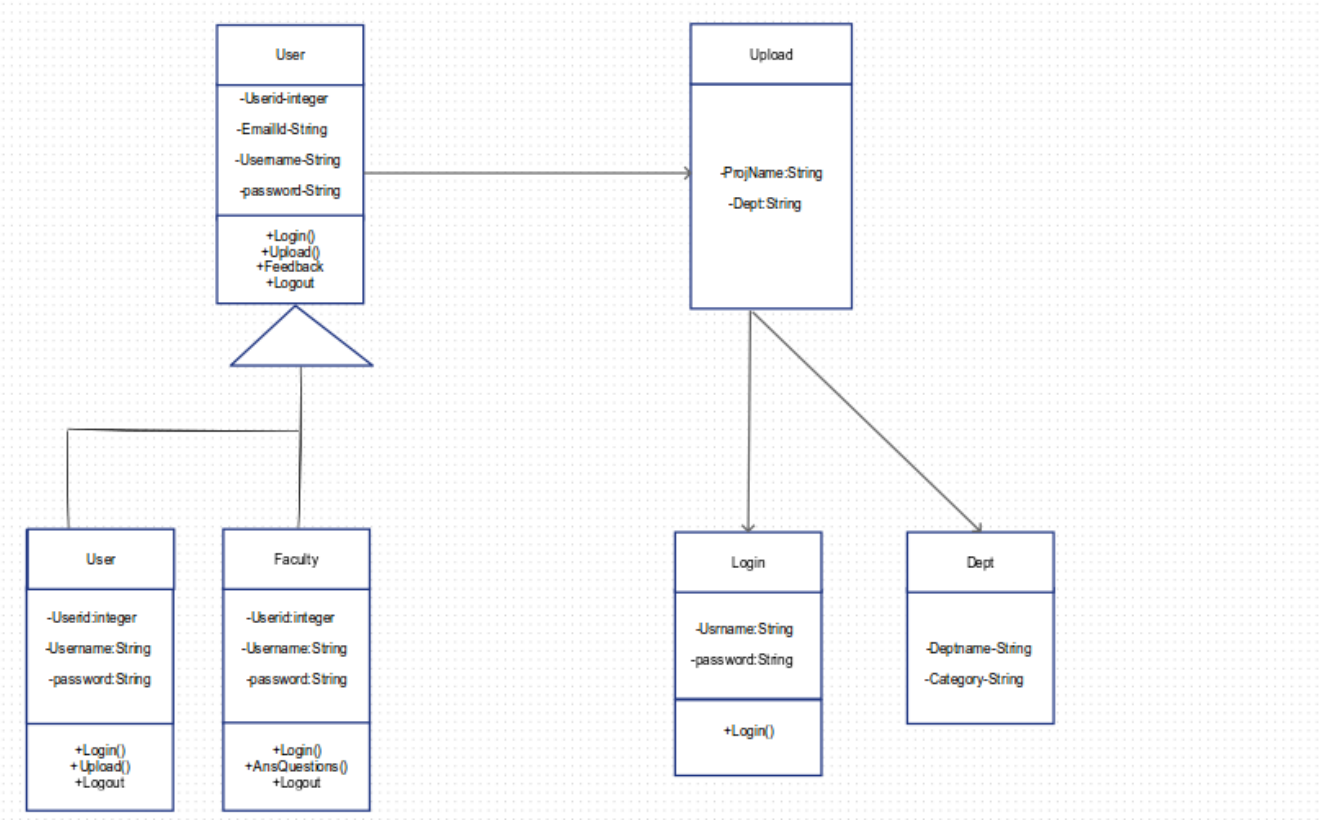


Architecture diagram

2.2 Description of Components

We divided our project into 4 modules. Those are
1. Graphical user interface
2. Database
3.Server
4.Client

2.2.1 Class Diagram :



class diagram

2.2.2 Home

This componet describes layouts like home page that is nothing but here a login page, on our website..etc. This can be showed to student/faculty, it shows top and recent projects to the user.

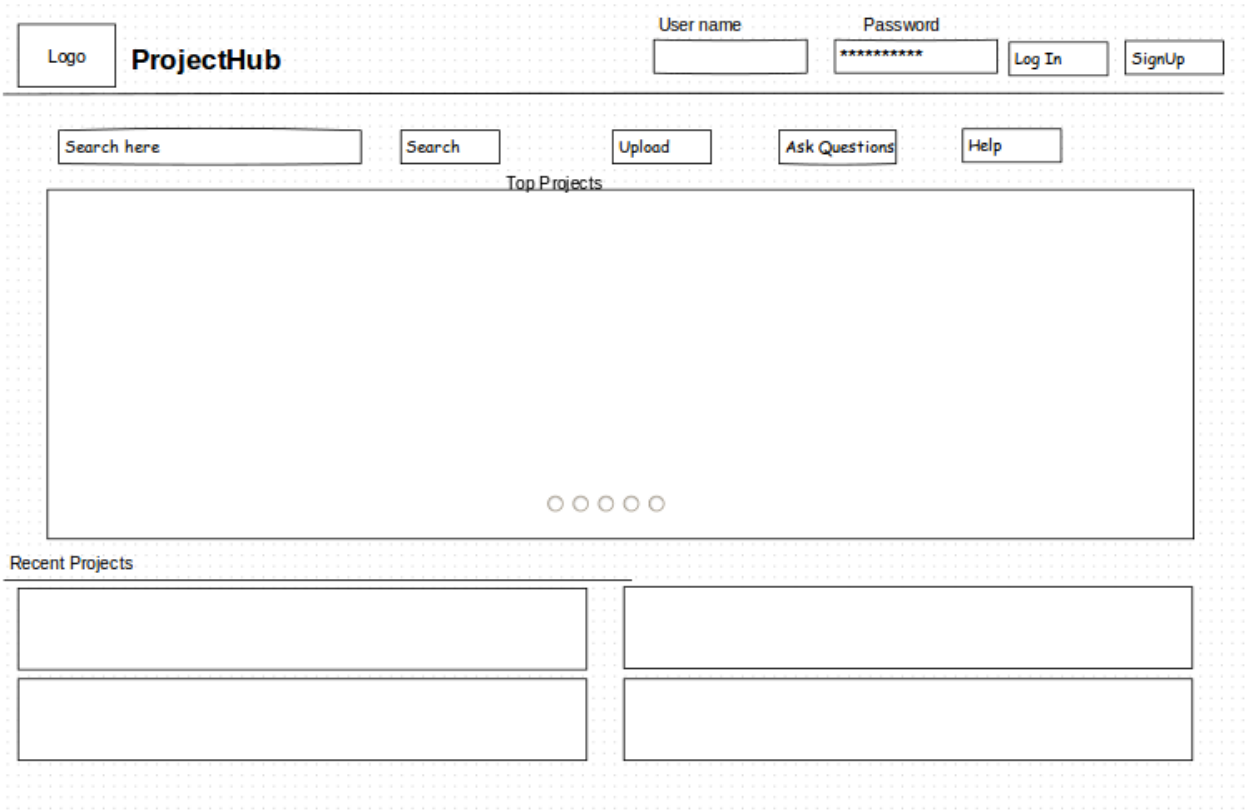


Figure:Home page

Attributes: username,password

Method#1: login

precondition : user must have username and password

postcondition: logged in

Algorithm:

```
flag=login(username,password)
if flag is true
        “login successfull”
else “either username or password is invalid”
```

2.2.3 Student/Faculty

Sign up:

This component creates new user accounts for the user.

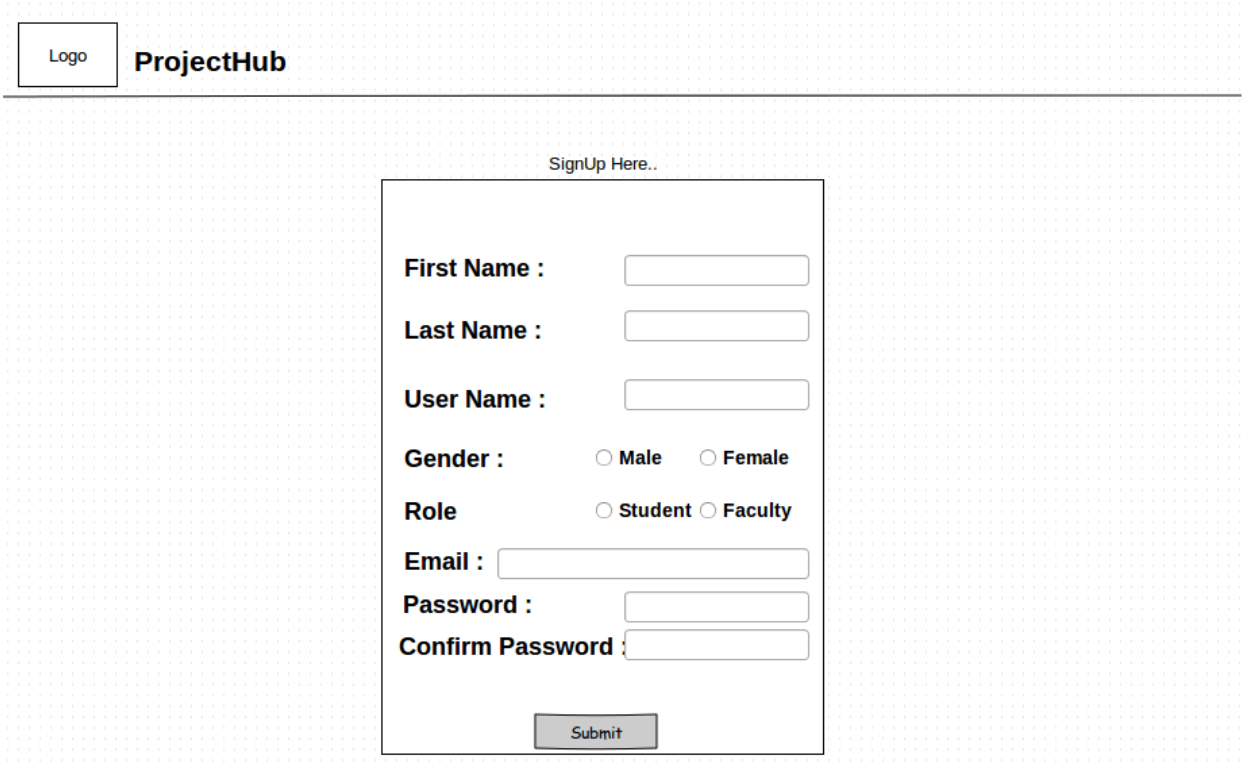


Figure : sign up

Attributes: Name,username,gender,role,email,password,confirmpassword

Method#2:signup

Precondition: User should be student/faculty

Post condition: Created account for user

Algorithm:

flag = signup(Name,username,gender,role,email,password,confirmpassword**)**
if flag is true
    return “registered successfully”
else
    return “enter details properly”

Change password:

This component allows users to change their passwords:

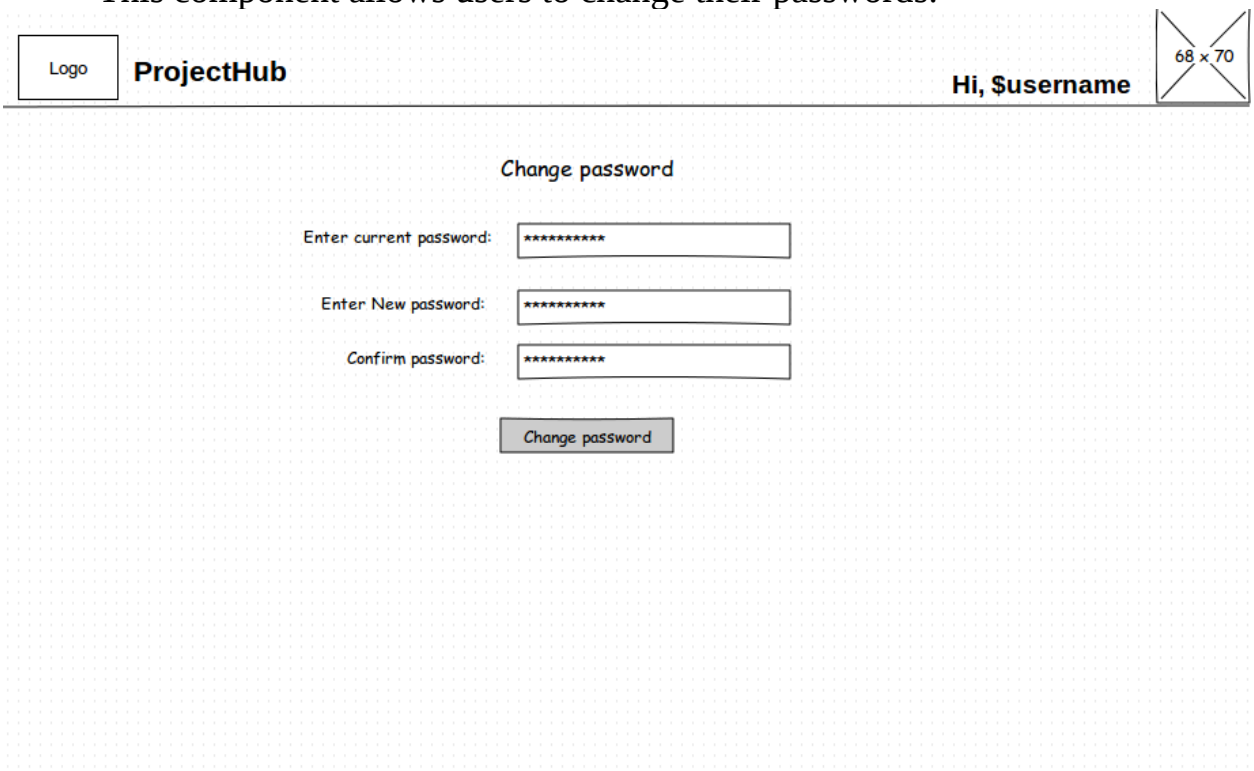


Figure: change password

Attributes: current password, new password,confirm password

Method#2: changepassword

Precondition: User must be logged in

Post condition: Change the password.

Algorithm:

flag = changepassword(current password, new password,confirm password**)**
if flag is true
    return “password changed successfully”
else
    return “enter details properly”

Profile:

This component allows user to view their profile.

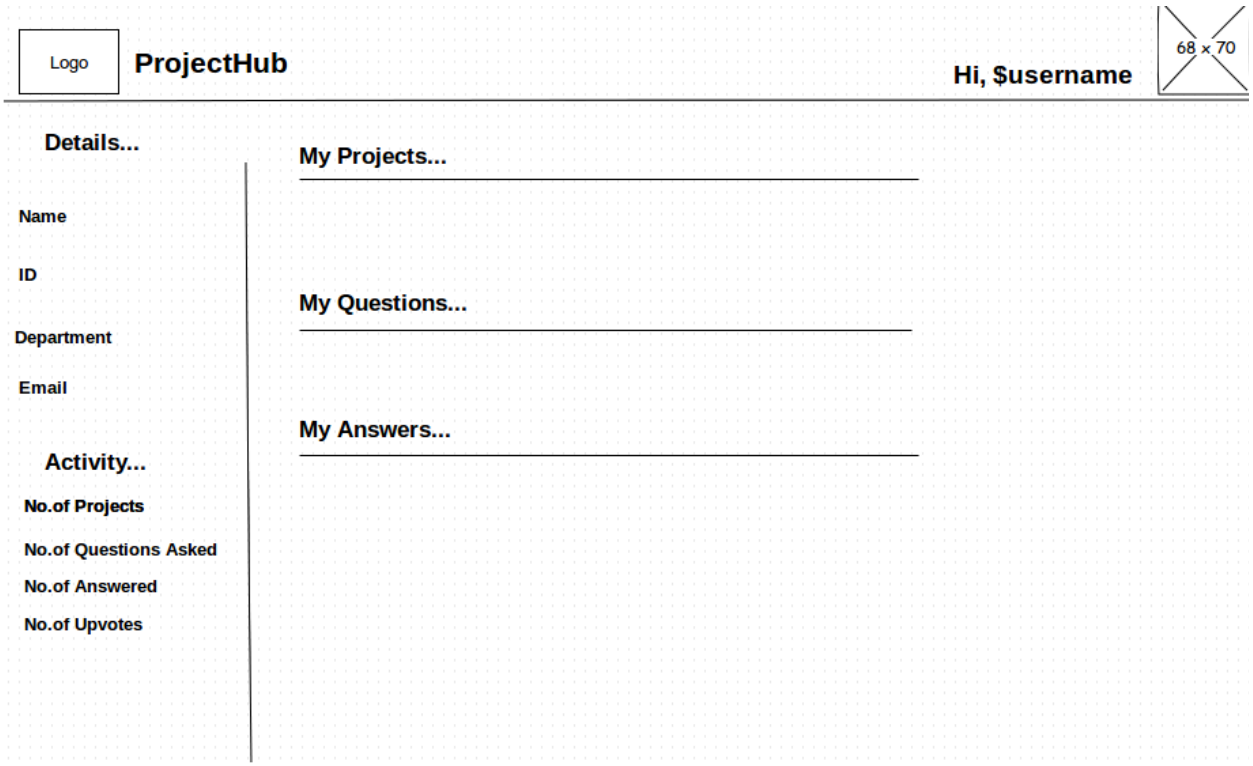


Figure : profile

Attributes:

Method: check()

Precondition: User must be logged in

Post condition: view user profile

Algorithm:

```
flag = check()
  if(user logged in)
          return 1;
  else return 0;
```

Search_page:

This component respond to the user queries to view projects list with small description

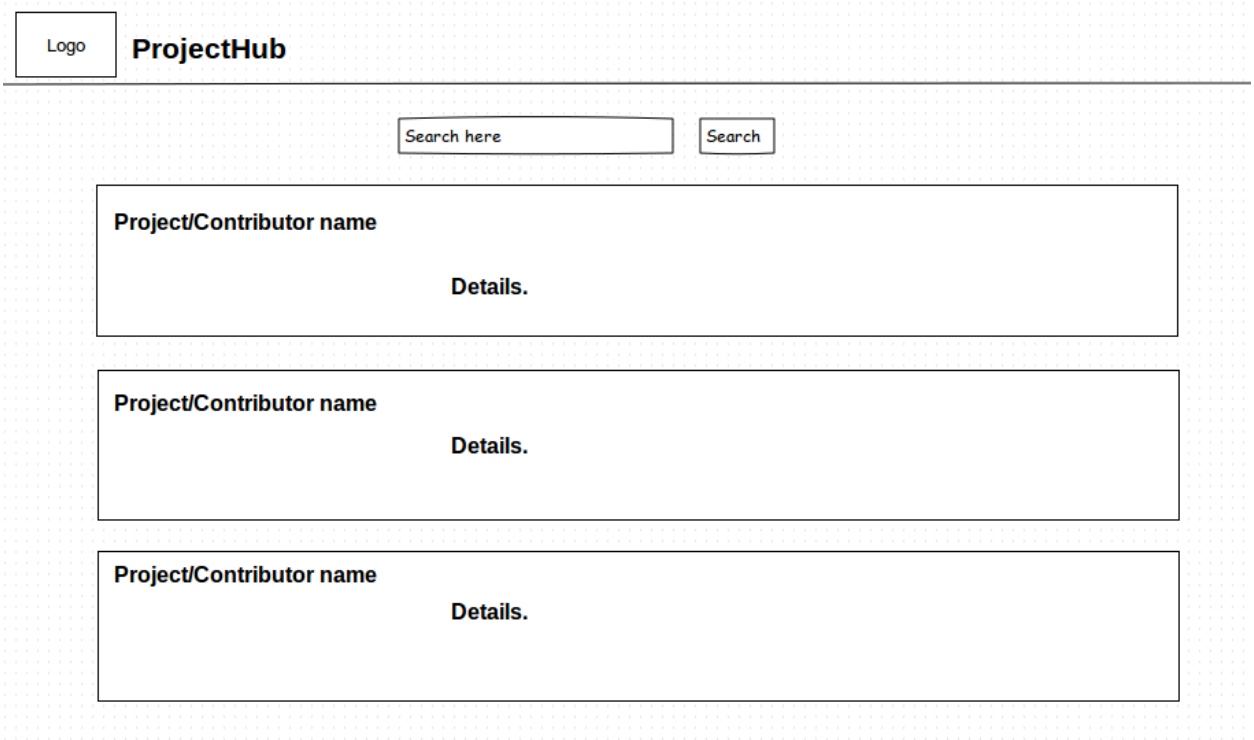


Figure: search

Attributes:search _qurey

Method: search(string)

Precondition: no need to be logged in

Post condition: view the different projects list

Algorithm:

flag = search(string**)**
if(string is valid)
view related projects list
else

show “No projects found”

Upload:

This component allows users to upload their projects:

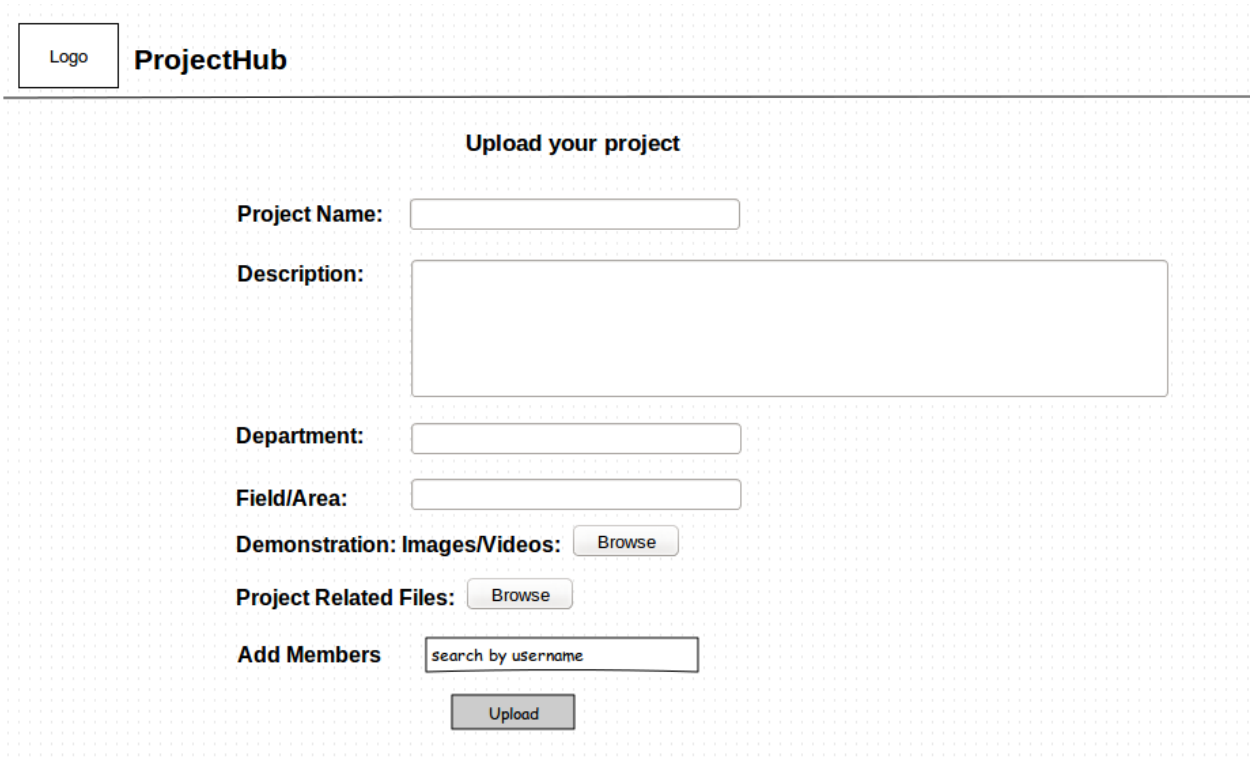


Figure : upload

Attributes:name,description,department,area,files,members

Method: upload**(**name,description,department,area,files,members**)**

Precondition: user must be logged in

Post condition : project is uploaded

Algorithm:

flag = upload**(**name,description,department,area,files,members**)**
if(fields are non-empty)
upload the project details
else
fields should be filled properly.

View_page:

This component allows to view the project details in detail.

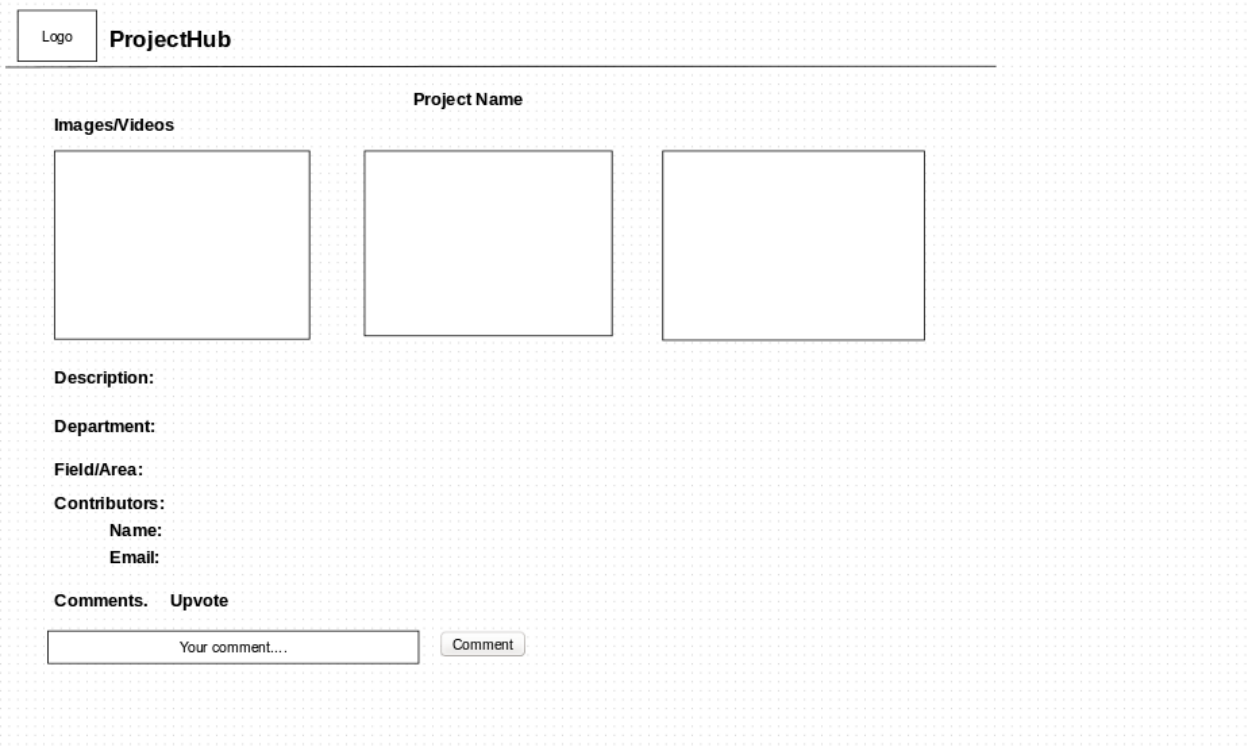


Figure : view_page

Attributes:

Method: view(project_link)

Precondition: user need not be logged in

Post condition :view the project detials

Algorithm:

```
flag =view(project_link)
  if(if data is availabe)
        view project details
  else
        project not found.
```

View_user:

This comment allows users to view details about other users

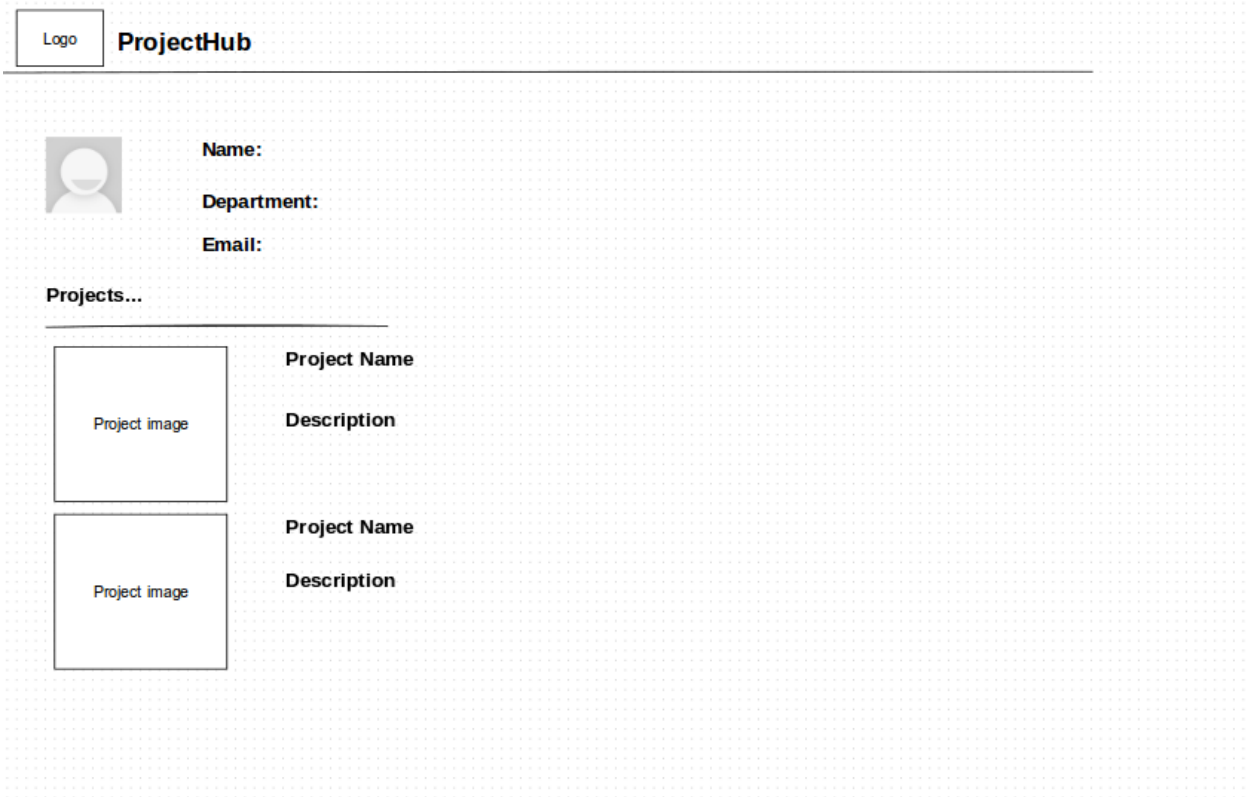


Figure: view user

Attributes:

Method: view_user(user_link)

Precondition: user need not be logged in

Post condition :view the user detials

Algorithm:

```
flag =view(user_link)
  if(if data is availabe)
        view user details
  else
```

2.2.4 Server:

Implementation of functions on server side module

Method#1: Login

l

```
login(username,password)

if(user name exist in database)
        if(password matches)
                return true
        else
                return false
else
        return false
```

Method#2: SignUp

signup(Name,username,gender,role,email,password,confirm password**)**

```
if(details are valide)
        create a user account in user table using given information
else
        user account can not be created
```

Method#3: Change Password

changepassword(current password, new password,confirm password**)**

```
if(current password is true)
        if(newpassword == confirm password)
                update the password in user details table;
        else
                show message “re enter the passwords”
else
        show message “incorrect password”
```

Method#4 :Profile

```
chekc();

    if(user is logged in)
        return true;
    else
        return false;
```

Method#5: Search for projects

```
    search(string_query)

    if(string found in projects description)
        return the projects list

    else
        return null;
```

Method#6:Upload

```
    upload(name,description,department,area,files,members)

    if(valid fields)

        update the projects table with given details
    else
        enter details correctly.
```

Method#7: View project

```
    view(project_link)

    if(project id is present in table)
        view project details

    else
        no results found;
```

Method#8: view_user


```
view(user_link)

if(user id is present in user table)
        view user details

else
        no results found;
```

3. Data Architecture

3.1 Description

Data collected from users are stored in the database in form of tables with which we manipulate corresponding records.

3.2. Entity-Relation Diagram

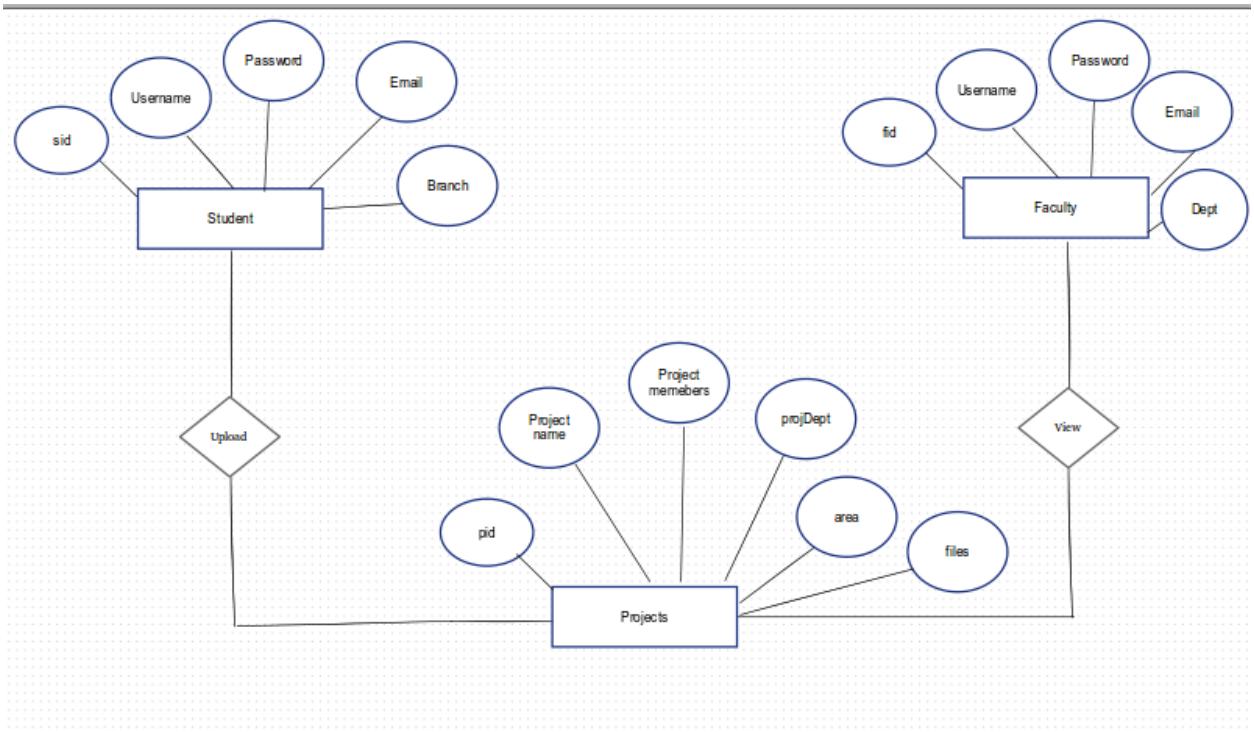


Figure : ER Diagram

3.3 Data Dictionary

Table name: Student

Field Name	Field Type	Length/size	Extra Information
Id	Int	11	AUTO_INCREMENT
Fname	Varchar	100	-
Lname	Varchar	100	-
Email	Varchar	100	-
Username	Varchar	30	-
Password	Varchar	100	-
Id no.	Varchar	100	-
Branch	Varchar	100	-
Role	Int	11	Default : 1
Ph	Varchar	100	-

Table name: Answer

Field Name	Field Type	Length/size	Extra Information
Id	Int	11	AUTO_INCREMENT
Answer	Text	-	-
Qid	Int	11	-
Username	Varchar	30	-
Date	Timestamp	-	Default : CURRENT_TIMESTAMP

Table name: Comments

Field Name	Field Type	Length/size	Extra Information
Id	Int	11	AUTO_INCREMENT
Pid	Int	11	-
Username	Varchar	30	-
Comment	Text	-	-
Date	Timestamp	-	Default : CURRENT_TIMESTAMP

Table name: Faculty

Field Name	Field Type	Length/size	Extra Information
Id	Int	11	AUTO_INCREMENT
Fname	Varchar	100	-
Lname	Varchar	100	-
Username	Varchar	30	-
Password	Varchar	100	-
Email	Varchar	100	-
Id no.	Varchar	100	-
Branch	Varchar	100	-
Role	Int	11	Default : 0
Ph	Varchar	100	-

Table name: Likes

Field Name	Field Type	Length/size	Extra Information
Pid	Int	11	-
Username	Varchar	30	-

Table name: Members

Field Name	Field Type	Length/size	Extra Information
Id	Int	11	AUTO_INCREMENT
Pid	Int	11	-
Username	Varchar	30	-

Table name: Project

Field Name	Field Type	Length/size	Extra Information
Pid	Int	11	AUTO_INCREMENT
Name	Varchar	100	-
Branch	Varchar	100	-
Field	Varchar	100	-
Files	Varchar	300	-
Members	Varchar	200	-
Date	Timestamp	-	Default : CURRENT_TIMESTAMP
Des	Longtext	-	-
File1	Varchar	100	-
File2	Varchar	100	-
File3	Varchar	100	-
Vfile	Varchar	100	-
Sfile	Varchar	100	-
Likes	Int	11	Default : 0

Table name: Question

Field Name	Field Type	Length/size	Extra Information
Qid	Int	11	AUTO_INCREMENT
Qname	Varchar	250	-
Username	Varchar	30	-
Date	Timestamp	-	Default : CURRENT_TIMESTAMP

CHAPTER 4
IMPLEMENTATION

SCREEN SHOTS :

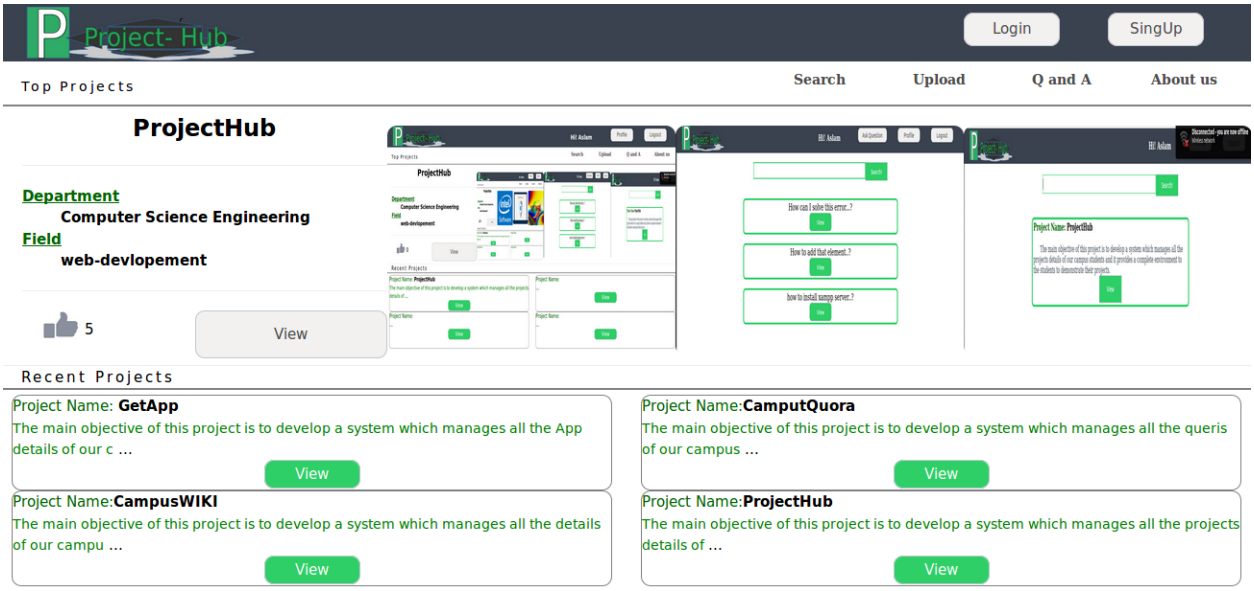


Figure 10 : Home Page

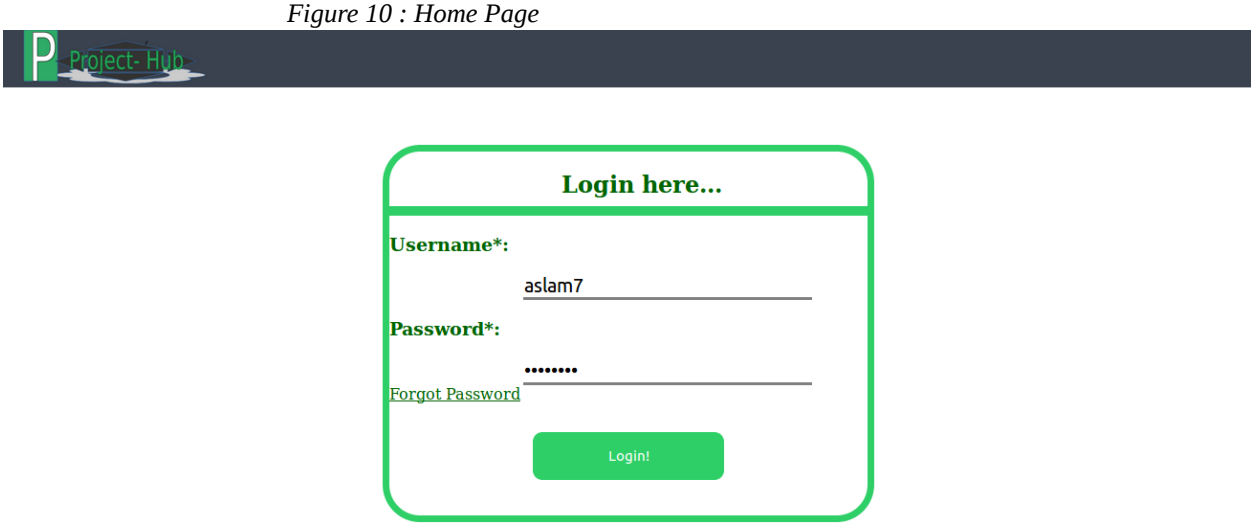


Figure 11 : log in

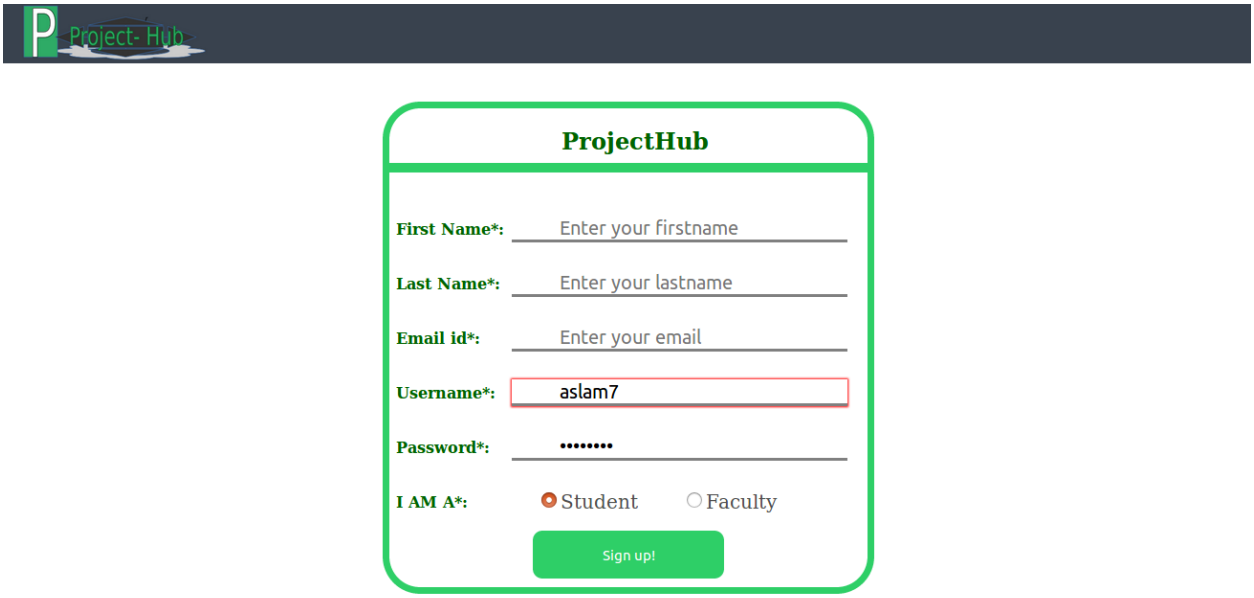


Figure 12 : sign up

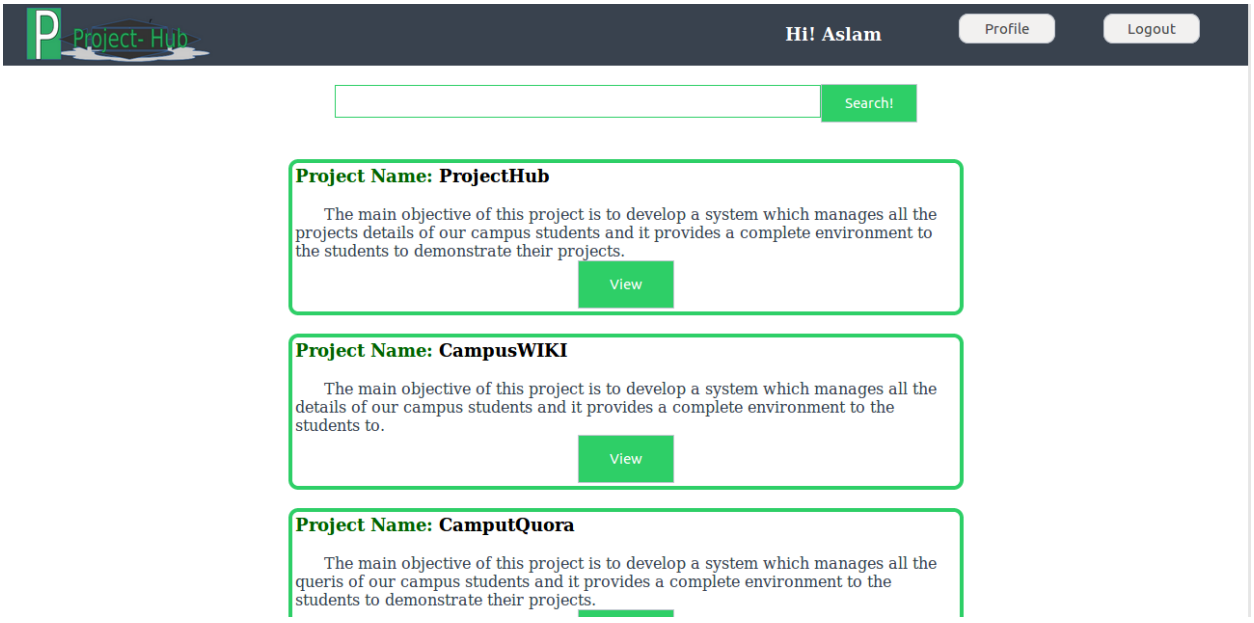


Figure 13 : search page

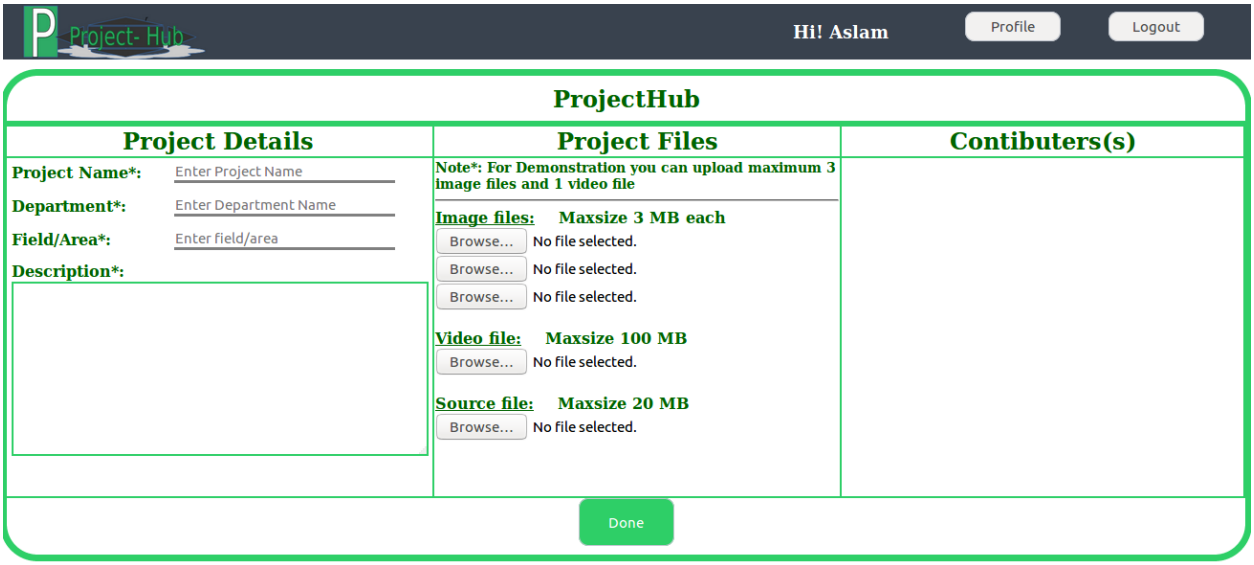


Figure 14 : upload

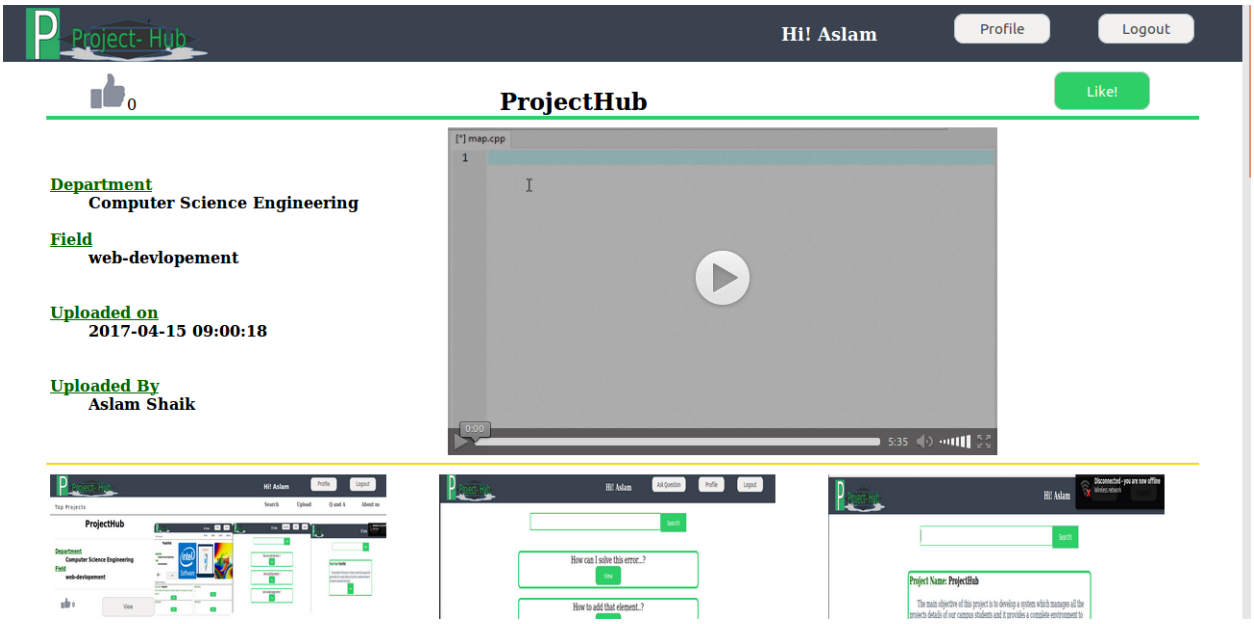


Figure 15 :view

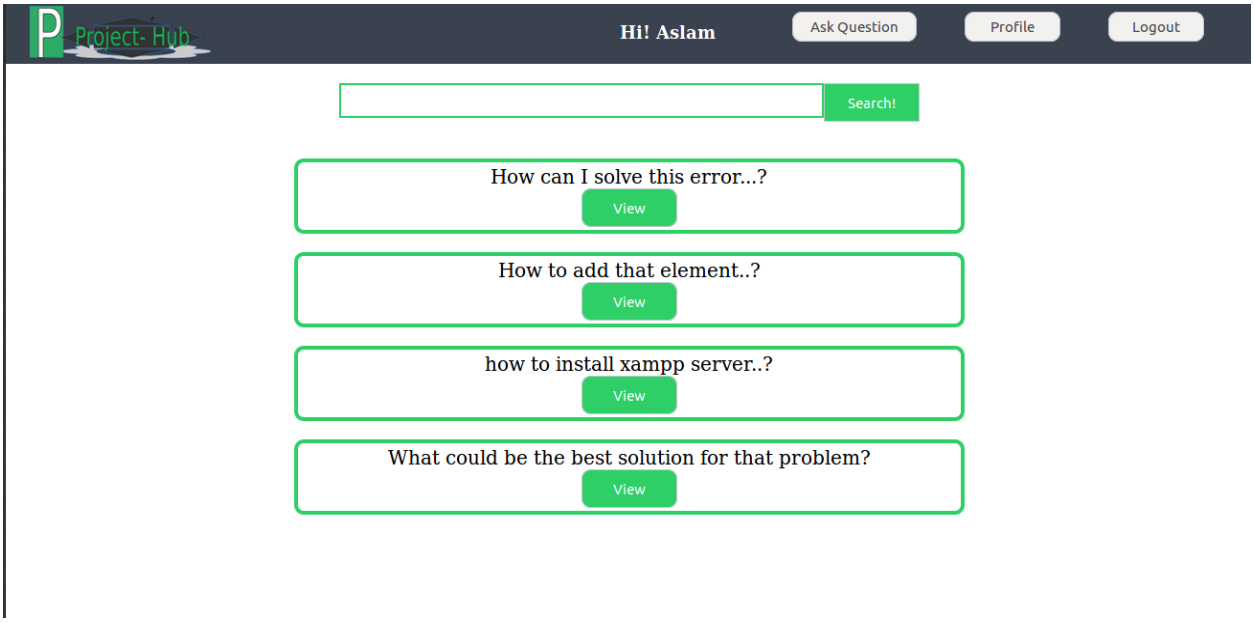


Figure 16: questions and answers

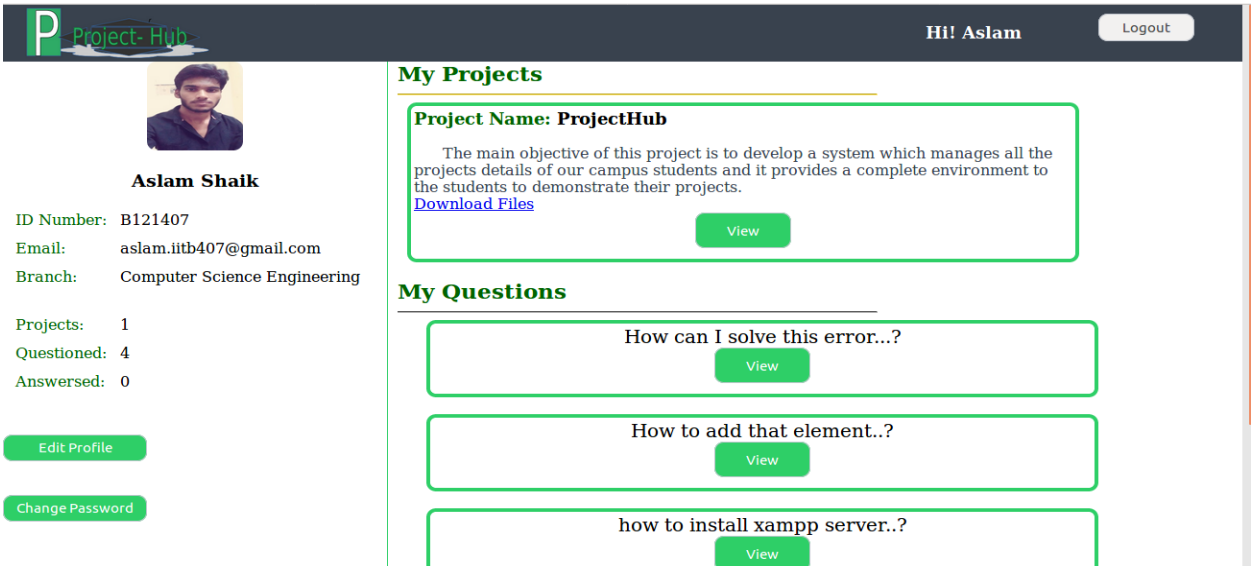


Figure 17: profile

CHAPTER 5

SOFTWARE TEST DOCUMENT

5.1 UNIT TESTING DOCUMENT(UTP)

1.Unit testing

Introduction:It is a level of Software testing where indvidual modules are tested.

1.Login

Introduction:The User has to login to the site before he wants to do any updates..

System overview:The User has to login with valid username and password then he is able to see home page of ProjectHub.

Test case id	Purpose	Inputs	Test procedure	Expected output	Actual result	Pass/Fail
GUI TS1	To check whether user is able to login successfully or not?	Correct username, correct password	1.Enter correct username and correct password in respective fields. 2.click on Login button.	User must successfully login to the webpage.	User successfully logged in to webpage	pass
GUI TS2	To check whether unregistered user is not able to login to the site or not? Login must be failed.	Incorrect username, Incorrect password	1.Enter incorrect username and incorrect password in respective fields. 2.click on Login on.	Proper error must be displayed and prompt to enter login again.	Error message was shown and stays in the same page.	pass
GUI TS3	Test with empty username and empty password suchthat login must be failed	Empty username,empty password	1.Enter empty username and empty password in respective fields. 2.click on login button.	Proper error must be displayed and prompt to enter login again	Error message was shown.	pass
GUI TS4	Test with empty username and valid password such that login must get failed.	Empty username and valid password	1.Enter empty username and valid password in respective fields. 2.click on Login button.	Proper error must be displayed and prompt to enter login again	Error message was shown.	pass
GUI TS5	Test with valid username and empty password such that login must get failed.	Valid username and empty password	1.Enter empty username and valid password in respective fields. 2.click on Login button.	Proper error must be displayed and prompt to enter login again	Error message was shown .	pass

GUI TS6	Check whether the passwors entered is visible or not? password should not be visible.	Valid/unvalid username and password	1.Enter password with some characters 2.click on submit button.	The password entered sholud be in bullets or asterisks	The password was shown in asterisks	pass
GUI TS7	Check if the login function handles case sensitivity	Enter case changed username / password	1.Enter case change password/username 2.click on submit button.	The user should not able to login to webpage.	The user was successfully logged into webpage.	fail

2.Sign Up

Introduction:

The User has to sign up if he don’t have an account.

System Overview:

The User has to sign up with valid details.The username may contain any special characters or digits and the files to be upload in described format.The password should contain atleast 8 characters.The phone number should start with 7,8 or 9 and contaions exactly 10 digits.

Test case id	Purpose	Inputs	Test procedure	Expected output	Actual result	Pass/Fail
GUI TS1	To check whether user is able to signup with correct details ?	Correct details	1.Enter all correct details. 2.Enter signup button.	User should be able to register.	User successfully registered.	pass
GUI TS2	To check the behavior of form by not filling any fields into the field	Empty form	1.Do not enter anything into the fields 2.click on sign up button.	no error will be displayed and enter signup form again	No Error message was shown but not able to redirect to same page again.	pass
GUI TS3	Check the behavior of form by not filling up the name text field but by filling up rest of the other fields.	Empty name and correct details of rest.	1.Enter empty name and fillup rest of other fields correctly. 2.click on signup button.	no error will be displayed and enter signup form again	No Error message was shown but not able to redirect to same page again.	pass
GUI TS4	Check the form by filling other text fields except email number text field.	Valid details and empty email	1.Enter empty email and fillup rest of fields correctly. 2.click on signup button.	no error will be displayed and enter signup form again	No Error message was shown but not able to redirect to same page again.	pass
GUI TS5	Check the form by filling other text fields except password text field.	Valid details and empty password	1.Enter empty password and fillup details correctly 2.click on signup button.	no error will be displayed and enter signup form again	No Error message was shown but not able to redirect to same page again.	pass

3.Upload

Introduction:In this function the user must be logged in as student to upload the project detials and related files.

System Overview:The user shoud enter valid project details to upload his/her project successfully.

Test case id	Purpose	Inputs	Test procedure	Expected output	Actual result	Pass/Fail
1	Check the field of project name.	Correct login	1.Load the file. 2.Enter upload button.	The user should get the message.	User should get the message about error and success.	pass
2	Test the project name with empty name.	Empty name.	1.Enter empty name. 2.Enter submit button.	The user must not able to error message.	Proper Error message was shown and redirected to same page.	pass
4	To check Done button working properly or not	Any input	1.Enter any input 2.click on done button	The user should be redirected to same page.	Working properly.	pass
5	Check the field of Department.	Correct login	1.Load the file. 2.Enter upload button.	The user should get the message.	User should get the message about error and success.	pass
6	Check the field of field/area.	Correct login	1.Load the file. 2.Enter upload button.	The user should get the message.	User should get the message about error and success.	pass
7	Check the size of the file.	Correct login	1.Load the file. 2.Enter upload button.	The user should get the message.	User should get the message about error message if file exceeds 3MB otherwise success.	pass
8	Check the size of the video.	Correct login	1.Load the file. 2.Enter upload button.	The user should get the message.	User should get the message about error message if file exceeds 100MB otherwise success.	pass
9	Check the size of the source file.	Correct login	1.Load the file. 2.Enter upload button.	The user should get the message.	User should get the message about error message if file exceeds 20MB otherwise success.	pass

4.Question

Introduction:In this function the user can Question projects.
System Overview:The user has to enter comment on desired project.

Test case id	Purpose	Inputs	Test procedure	Expected output	Actual result	Pass/Fail
1	Check the minimum length of the string.	valid question	1.select valid question. 2.Enter your question	Proper confirm message should be shown to user.	Your Question uploaded successfully message should be shown otherwise error message should be shown.	pass
1	Check the maximum length of the string.	valid question	1.select valid question. 2.Enter your question	Proper confirm message should be shown to user.	Your Question uploaded successfully message should be shown otherwise error message should be shown.	pass

5.Comment

Intoduction:In this function the user can comment on the projects.
System Overview:The user has to select desired project and then comment on the project.

Test case id	Purpose	Inputs	Test procedure	Expected output	Actual result	Pass/Fail
1	Check the description box with no characters.	valid project	1.select valid project. 2.Enter your comment	Proper confirm message should be shown to user.	Your comment uploaded successfully message should be shown otherwise error message should be shown.	pass
2	Check the description box with minimum characters.	valid project	1.select valid project. 2.Enter your comment	Proper confirm message should be shown to user.	Your comment uploaded successfully message should be shown otherwise error message should be shown.	pass
3	Check the description box with maximum characters.	valid project	1.select valid project. 2.Enter your comment	Proper confirm message should be shown to user.	Your comment uploaded successfully message should be shown otherwise error message should be shown.	pass

1.Integration Testing

Introduction:Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

Test case id	**Purpose**	**Inputs**	**Test procedure**	**Expected output**	**Actual result**	**Pass/Fail**
1	To check whether user is directing to homepage after loin to site.	Correct username,correct password	1.Enter correct username and correct password in respective fields. 2.click on submit button.	User must successfully login to the webpage and prompt to home page.	User was directed to homepage.	pass
2	To check whether user is able to logout after loggedin to site.	Correct password and correct username.	1.Login to the site. 2.click on logout button.	User should be able to logout after login to home page.	User successfully loged out of homepage.	pass
3	To check whether user is able to enter functionalites of system after loged into homepage.	Empty username,empty password	1.login to home page. 2.Enter any function.	User should be able to enter any functionality.	User successfuuly entered into functionalities.	pass
4	To check whether user is able to log in after successfully registered.	Correct details.	1.Register with correct details. 2.Try to login with registered usename and password	User must successfully login to the webpage and prompt to home page.	User successfully loged into site.	pass

Conclusion

The main motto of this project is to develop a system which manages all the projects details of our campus students and it provides a complete environment to the students to demonstrate their projects and help other students to improve their practicle knowledege.