**Course COMP1640**

**Enterprise Web Software Development**

****

**Submitted To Submitted By**

Dr. Ray Stoneham Username: **mb6883g**

Course leader Submission Date 12th April 2018

University of Greenwich, UK

**Group Name – “TeamX”**

**Group members**

|  |  |  |
| --- | --- | --- |
| **SL.** | **Name** | **Role** |
| 1 | Mohammad Mahmudur Rahman Babu | Tester |
| 2 | Rubel Mahmud | Programmer |
| 3 | Naim Islam | System Analyst and Designer |
| 4 | Zahid Hasan | Database Designer |

**Important Links**

|  |  |
| --- | --- |
| System URL | <https://ideaportal.rubelmahmud.me/> |
| Screen cast URL | [https://youtu.be/Y5fSpWwKnj0](https://l.facebook.com/l.php?u=https%3A%2F%2Fyoutu.be%2FY5fSpWwKnj0&h=ATPoUMWzuKKaAPoT3EnjqJjeIBbWFcfx6CdyE_x7DjOjmDzsM3eK1CUgQ3r_0FW6MA90-GTDExIHr8NXqIuwdzli4VFrHpyR7e_gJIZwLd-wKzhY7h67Ww) |
| Repository | [goo.gl/4TqFW1](https://l.facebook.com/l.php?u=http%3A%2F%2Fgoo.gl%2F4TqFW1&h=ATN85SvvbxFaog8Ylqag09Onyv5sKr83zwo33NSx3nxHHMkNY1dkywpceMBLzdo3kWohcghjBh9x6LslQaO9TsSokbn3mGaB3Z3oKhSTq9bWcldVgiv-tw) |

**Credentials**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Pin** | **Usermail** | **Password** |
| Admin |  |  |  |
| QAM |  |  |  |
| QAC(IFY) |  |  |  |
| QAC(L4DC) |  |  |  |
| QAC(L5DC) |  |  |  |
| QAC(IFY) |  |  |  |
| Supervisor |  |  |  |
| Student |  |  |  |

# Introduction

This course work is focused on developing Enterprise Web Software. A group of members works on several task. They lead in different role according to their knowledge on that particular task. Agile methodology is used to develop the system. Product backlog is listed. Every task is completed according to product backlog. Here we have developed a web based idea sharing system. This as a role based system. There are two types of user. Users of the system get access to the system on their role basis.

# Description of Final Built System

A web based secured idea sharing system is developed. It is a role based system. There are two types of user and these are student and staff. Staff types included Admin, Quality Assurance Coordinator of different department (QAC), a Supervisor and a Quality Assurance Manager (QAM). Admin can add new user and manage user information when QAM is able to manage category, generate different reports and user summary, QAC is able to manage ideas and generate reports that are related to their department.

**Home page**

System URL leads to the system home page. There is login option for users.

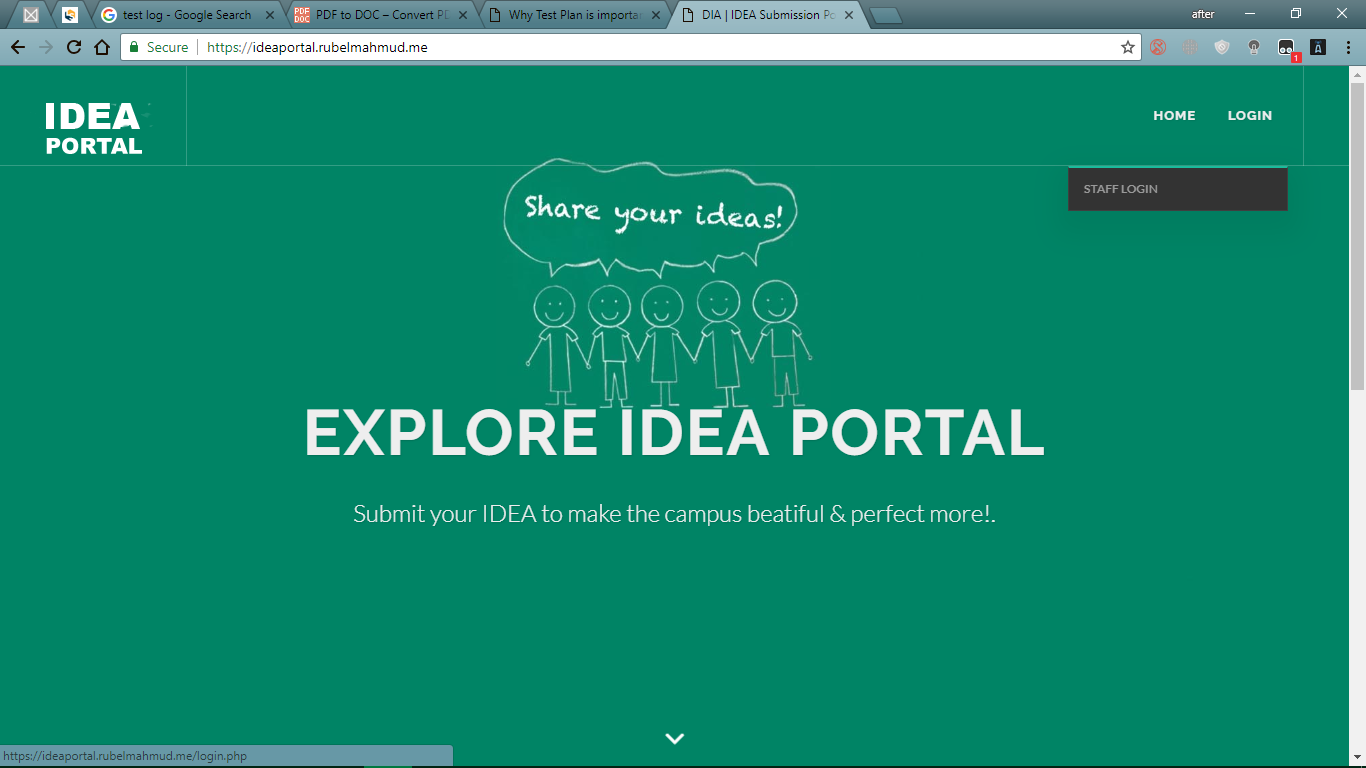


Figure: System home interface

**Staff Login**

Staff could be able to login to the system providing user mail and password.

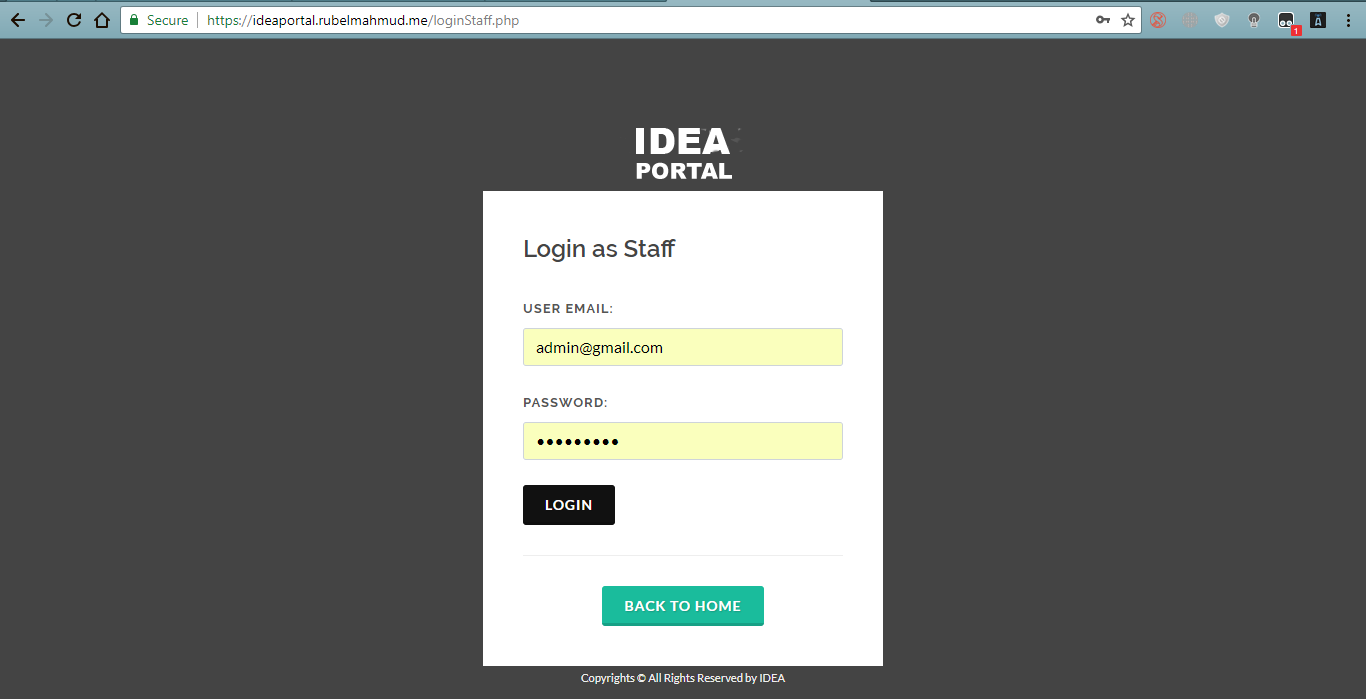


Figure: User login interface

**System user list**

All user lists with student and staff information.

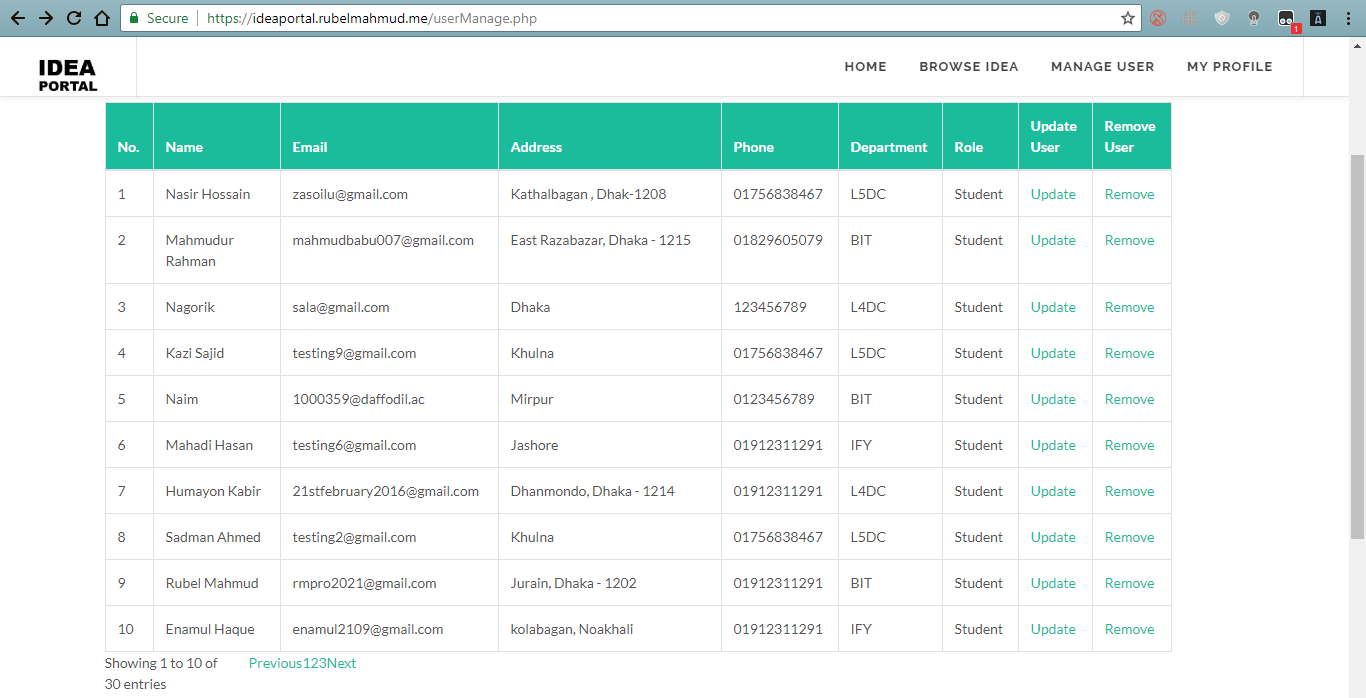


Figure: All user list view

**User summary**

Total user summary of the system

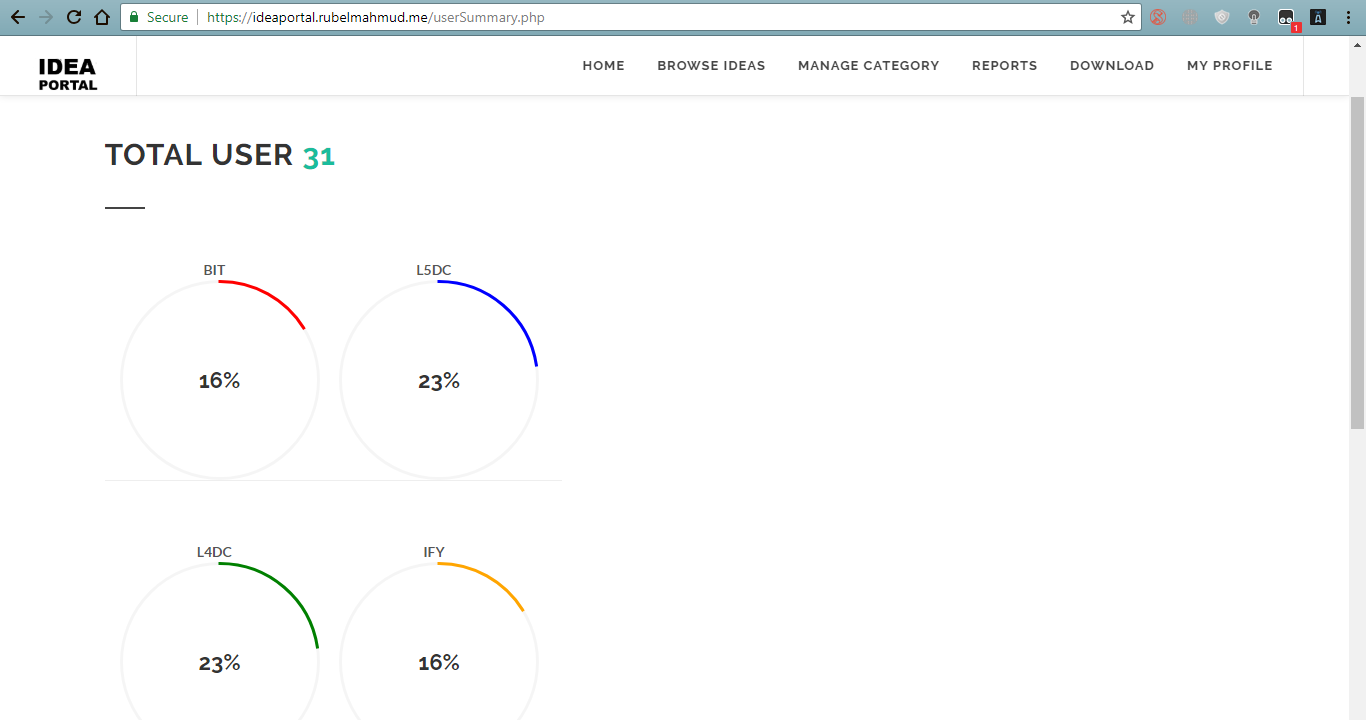


Figure: Summary of total user of the system

**Add Category**

A new category could be added from this page.

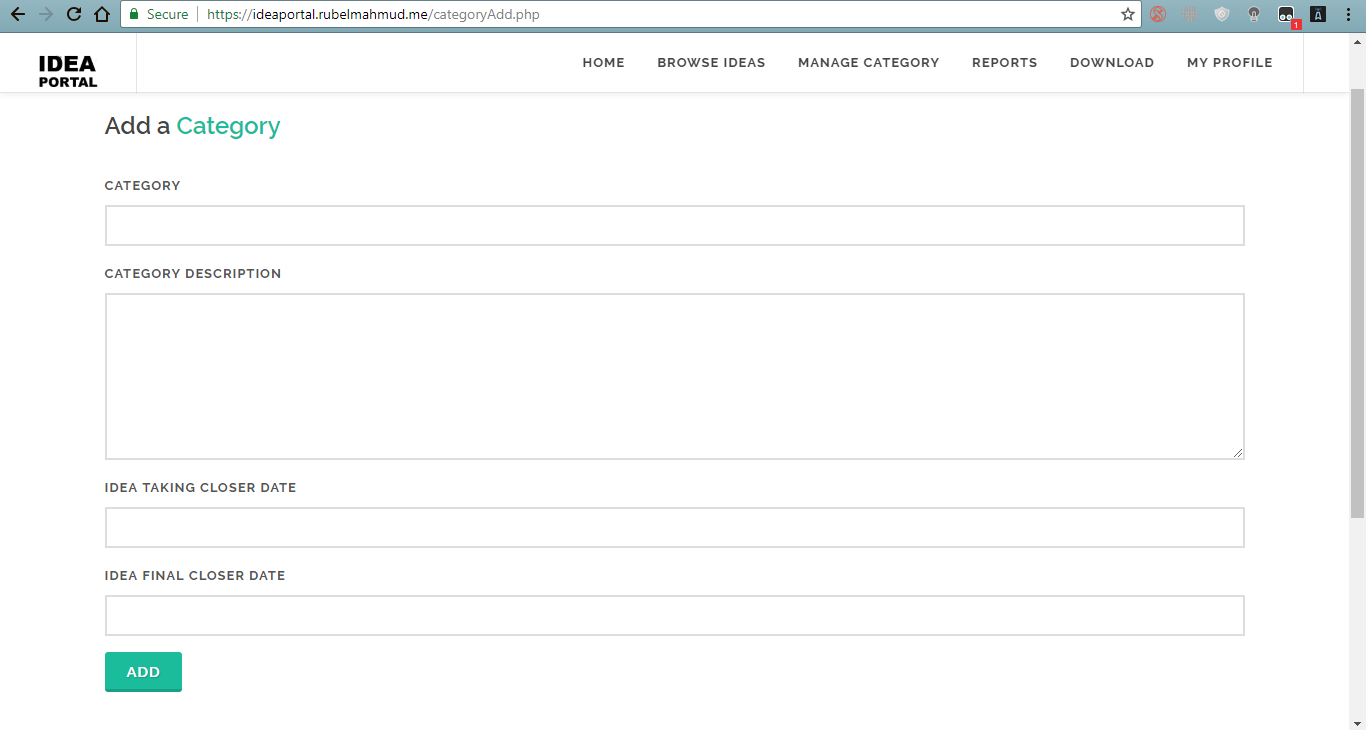


Figure: Category adding interface

**Category List**

All category lists can view.

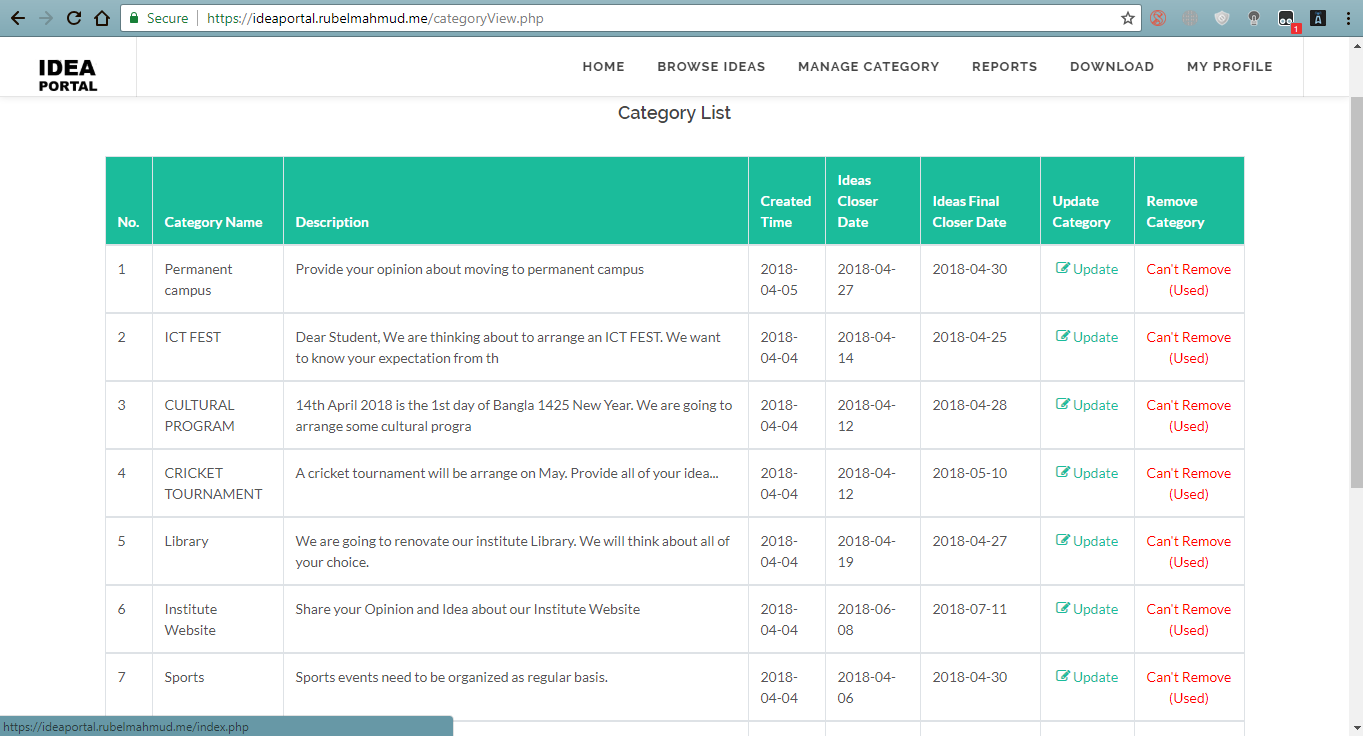


Figure: All category list

**Idea Submit**

Idea could be submitted providing information.

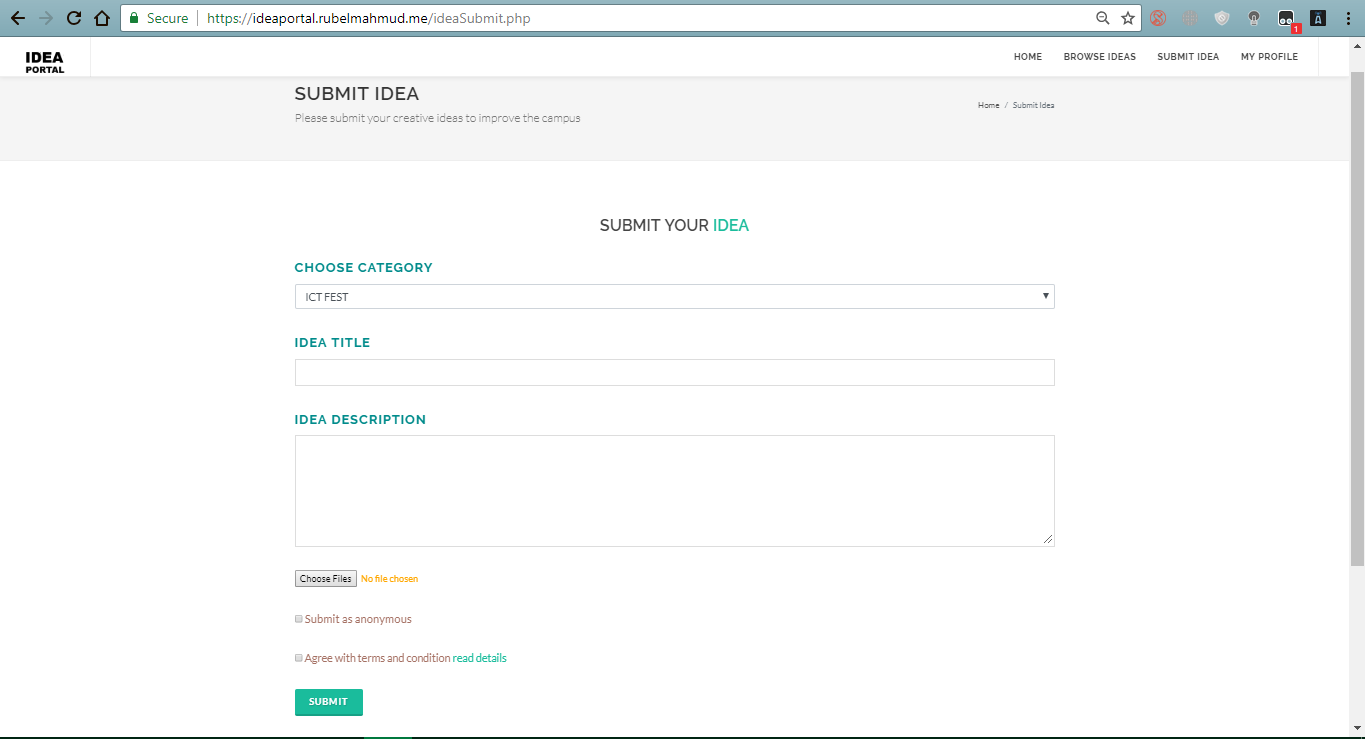


Figure: Idea submit form

**Idea List**

All submitted idea in a list view.

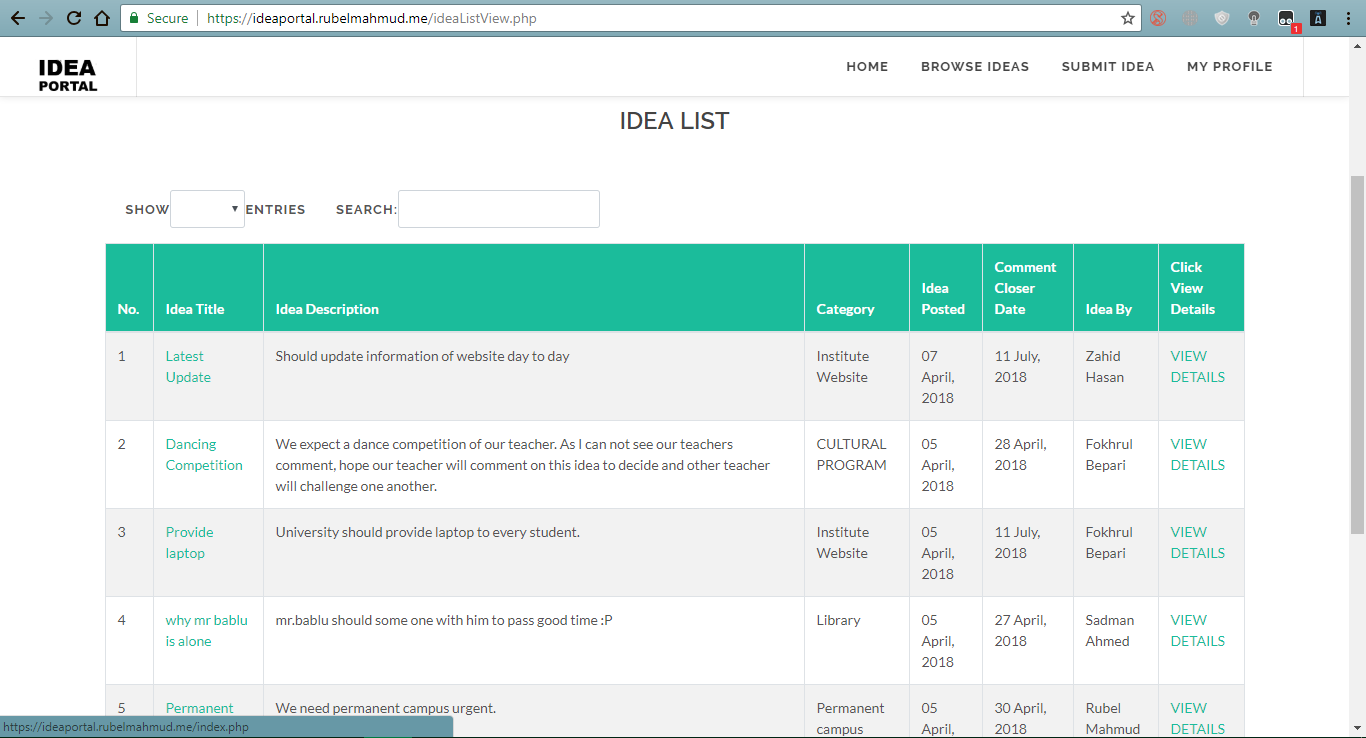


Figure: Submitted Idea with description

**Comment and thumps up/down**

Commenting on ideas and like/unlike an idea.

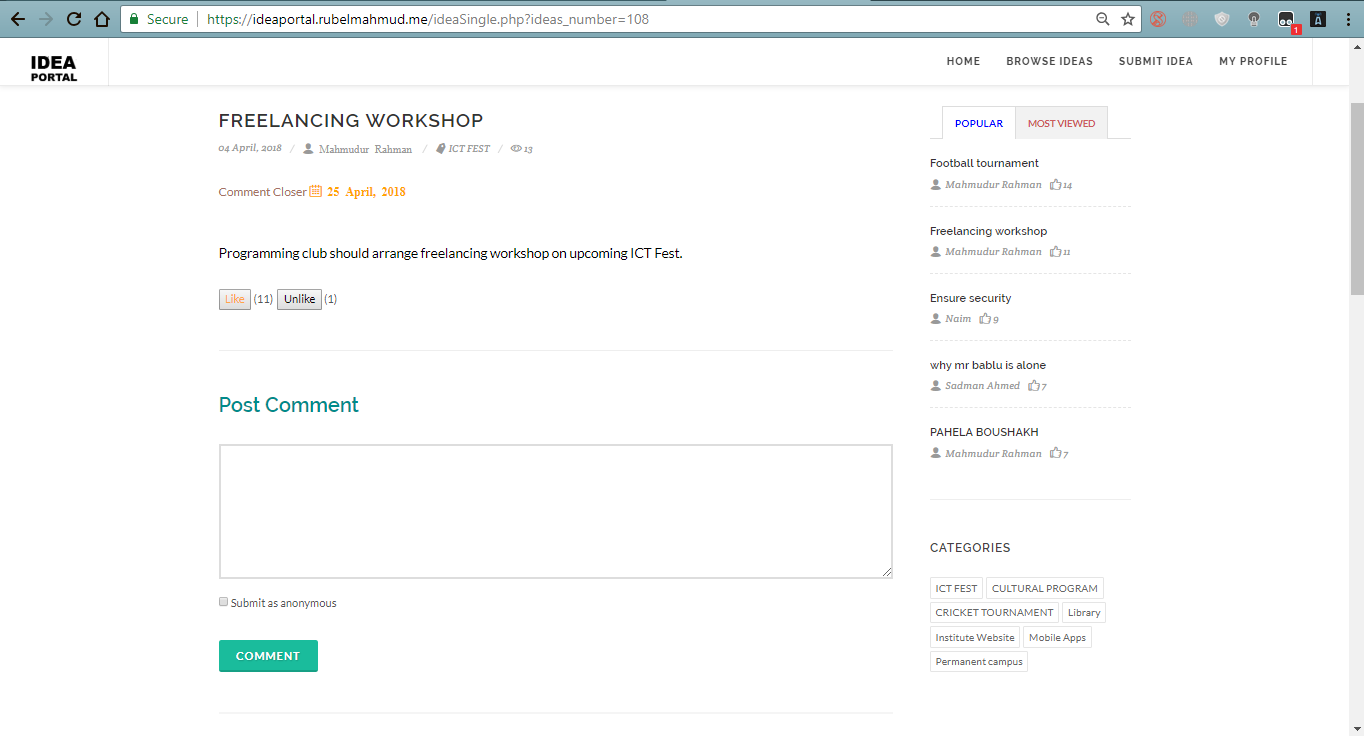


Figure: Thumps up/ down feature

## Users Role and their access to the system

The users of the system get access to the system according to their role. The role based access to the system are given below

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S/N** | **Features** |  |  | | **Roles** |  | |
| **Admin** | **QAM** | **QAC** | | | **Student** |
|  |  |
| 1 | Sign in to the system | ✔ | ✔ | ✔ | | | ✔ |
|  |  |  |  |  | | |  |
| 2 | Add user | ✔ | ✘ | ✘ | | | ✘ |
|  |  |  |  |  | | |  |
| 3 | Manage user | ✔ | ✘ | ✘ | | | ✘ |
|  |  |  |  |  | | |  |
| 4 | Add category | ✘ | ✔ | ✘ | | | ✘ |
|  |  |  |  |  | | |  |
| 5 | Manage category | ✘ | ✔ | ✘ | | | ✘ |
|  |  |  |  |  | | |  |
| 6 | Submit idea | ✘ | ✘ | ✘ | | | ✔ |
|  |  |  |  |  | | |  |
| 7 | Browse Ideas | ✔ | ✔ | ✔ | | | ✔ |
|  |  |  |  |  | | |  |
| 8 | Comment on Ideas | ✔ | ✔ | ✔ | | | ✔ |
|  |  |  |  |  | | |  |
| 9 | See students comment | ✔ | ✔ | ✔ | | | ✔ |
|  |  |  |  |  | | |  |
| 10 | See staffs comments | ✔ | ✔ | ✔ | | | ✘ |
|  |  |  |  |  | | |  |
| 11 | Manage ideas | ✔ | ✔ | ✔ | | | ✘ |
|  |  |  |  |  | | |  |
| 12 | Generate report | ✘ | ✔ | ✘ | | | ✘ |
|  |  |  |  |  | | |  |
| 13 | Send individual email | ✘ | ✘ | ✔ | | | ✘ |
|  |  |  |  |  | | |  |

# Evaluation of product and process

The system was developed properly following the requirements. Every user role was well defined and security of the system was highly prioritized. The developed system is able to perform every required task properly.

# Strength of the product

The built system has met all the required functionality. Besides fulfilling all the requirements the system has some features that have made the system functionally strong. Some strength of the system are given below-

**Secure system**

Security was prioritized during development of the system. Enterprise level software needs to be secure. User login is secured by providing separate login platform. Staff login system is high secured with 2 steps login. The system session destroy confirm the security of user session. Validation and cookies was assured for security purpose.

**Role based system**

The system has role based functionalities and access to the information. Student and other staffs have role based access to different part of the system.

**Responsive design**

One of the strength of the system is responsiveness of the system. The system is compatible in all types of devices as well as all browsers.

**Mail notification system**

User of the system will get mail notification after completing several part of system.

**Reliable system**

The system is reliable. User can provide any contribution without any hesitation.

**User friendly interface**

User will find it easy to complete their task. User friendliness was confirmed during the development of the system.

**Reports generate**

Staff will be able to generate reports of different information. A list of student contribution document on could be downloaded in a zip format by QAM.

**Weakness of product**

Besides strength of the system there is some minor weakness that could have take place in the system. But the weakness will not lessen our system functionality and performance. Some weakness are given below-

**Resetting Password**

User cannot reset their password and Usermail.

**Reply to comment**

No reply option to reply to a comment made on a idea.

# Further Development

The system is developed according to the requirements. This is earliest version of the system. Some features and requirements could be added on the future according to user experience. The weaknesses that are identified would be tried to fulfill. We will try to focus on some requirements that should be developed in future-

**System security**

The developed system is well secured. But within the updating technology newer threat could arise. So we will try to enable more security of the system.

**Reply to comment**

Reply to other users comment contribution could be added during the future development of the system.

**Messaging**

User could be able to exchange messages between them. A messaging feature could be added on the further development.

**Unauthorized user blocking**

Unauthorized login attempt would be blocked by identifying maximum number of wrong login information.

**Update user profile**

User personal information editing option could be added during the future development of the system. Updating on some information and uploading user image could be possible by user.

# Evaluation of Team

It was a great experience to work with some hardworking, skillful and tremendous people. The group environment was friendly and working style was as enterprise level. Everyone was very helpful and dedicated. Everyone contributed a lot from their side. Without their effort and contribution it was impossible to develop the system. No one was treated as less important in the group. They have excellent knowledge at their field. I would love to work with them in near future. Though a range of score cannot describe about them, I have evaluated their work with a range of score and commented about their effort in group work-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Member**  **Name** | **Role** | **Evaluation Items** | | | | **Comment on**  **Team Member** |
| **Participation**  **In Group Meeting** | **Helping Hand** | **Work Quality** | **Task**  **Accomplish on time** |
| **Rubel Mahmud** | Programmer | 8/10 | 10/10 | 9/10 | 9/10 | Programmer is one of the most important members of the team. He played is very role perfectly. Though there were some irregularities attending in team meeting, he overcame with his hardworking skill. I will love to work again in his team. |
| **Naim Islam** | Analyst & Designer | 10/10 | 8/10 | 9/10 | 9/10 | Was Punctual in attending team meeting and his skill on his field is very good. I hope he will be successful in his work field. |
| **Zahid Hasan** | Database Designer | 8/10 | 9/10 | 10/10 | 8/10 | Amazing team member with great skill of work in designing database. Delivered a good quality full database that helped to reach our goal. |
| **Mahmudur Rahman** | Tester | 10/10 | 9/10 | 8/10 | 9/10 | Was punctual on joining team meeting and tried to help other group member. Testing plan and testing was performed in enterprise level. |

# Self Evaluation

As the system was developed by a group of members, everyone tried their best to deliver the best outcome. Every member played an important role from their side. It is true that anyone cannot describe about himself neutrally. As our group was consist of four members, the other three could better evaluate about me. But, from my side I can assure that I tried my best to help my team members and tried to act my role perfectly. In my group my role was as tester. As a tester I tried to give my best effort in group work and my individual task. I tried to participate every group meeting and provided my opinion on any group decision. Though my role was tester, I helped analyst in analysis and identify product requirements. I produced a test plan that helped the developer to develop the system in a right way. According to the test plan I tested every task that was completed following product backlog. After completing the development of the system, the whole system was tested and produced a test report. Though I had some lacking of skill, I overcame this by taking help from my group members and from internet. In this group work I tried to contribute my best with honesty.

# Lesson Learnt

This coursework has taught me many things to work on a group, work with enterprise software development and follow product backlog. First of all I have learned to communicate with a group of people and express myself to other. I have learnt that any large work could be completed easily if we work as a team.

To perform my testing role I had to study about software quality testing, criteria and security. That will help me to utilize in the future. I have learnt how to secure a system and how to protect a system from any kind of threat. The most important thing that I have learnt is working in enterprise software development environment. I have learnt how to document a test plan and execute different test case according to test plan. The experience that I have gathered it will help me to utilize in my professional life. I faced few problems during testing of the system and solved them properly that enlarged my problem solving skill. At last I want to thanks my group members and my course leader for giving me opportunity to work on this friendly environment.

# Conclusion

All of us tried to develop the system according to all the requirements. All the requirements were completed. Every user role is ensured properly. User will find it easy to use the system. This course was very important and the experience and knowledge that I have gathered will be helpful for future. Working in a group was enjoyable.

# Appendix: A

## Test Plan

Testing is important in development lifecycle of any software. Test plan describe the list of testing that will be performed after development of particular part of the system. The initial test plan contains brief ideas about testing before we perform different test case. Test plan ensure that functional and design requirements are progressing according to the documentation. A test plan keeps a software development in the right track and helps to reach the goal. The following tests will be conducted during testing-

* + **Functional Testing**
  + **Security Testing**
  + **Performance Testing**
  + **Responsive testing**
  + **Integration Testing**
  + **Role based access to the system**
  + **Usability testing**
  + **Validation testing**
  + **Database testing**
  + **Platform Compatibility testing**
  + **Cross Browser Testing**
  + **Crowd Testing**
  + **Sprint Backlog Testing**

# Appendix: B

## ****Testing****

Testing ensures a system quality. Testing measures system security, performance and usability. After completing the development system is tested according to the test plan. Various tests are executed providing data. Here the system test that are processed are given-

* + **Functional Testing**
  + **Security Testing**
  + **Performance Testing**
  + Responsive testing
  + **Integration Testing**
  + Role based access to the system
  + **Usability testing**
  + Validation testing
  + **Database testing**
  + Platform Compatibility testing
  + Cross Browser Testing
  + Crowd Testing
  + Sprint Backlog Testing

## Test Environment

**Device:** Laptop ( Acer ).

**Operating system:** Windows 10 Pro ( 64 bit).

**Browser:** Google Chrome Version 67.0.3386.1 (Official Build) dev (64-bit) , Mozilla Firefox 60.0b8 (64-bit) , Microsoft Edge , Safari.

## Test Cases

**Functional Test**

The functional test will conducted to test all the functionalities working properly.The following functionalities will be tested-

* Manage user
* Manage Category
* Set closure date
* Set idea final closure date
* Comment anonymously
* Email notification
* Generate report
* Download zip files of uploaded idea supporting document
* Send Email
* Submit Idea
* Comment on idea
* Thumbs up/down on an idea.
* Upload Idea supporting document
* Proper pagination

**Test case for functionalities**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 1.1 | Functionality  Testing | QAM Add category and set closure date and final closure date | The closure date and final closure date will be saved with category information in category list | Closure date and final closure date is saved successfully | None |
| 1.2 | Functionality  Testing | Commenting on a student’s idea | The idea author student will get a notification on his mail | Idea author gets a notification on his mail | None |

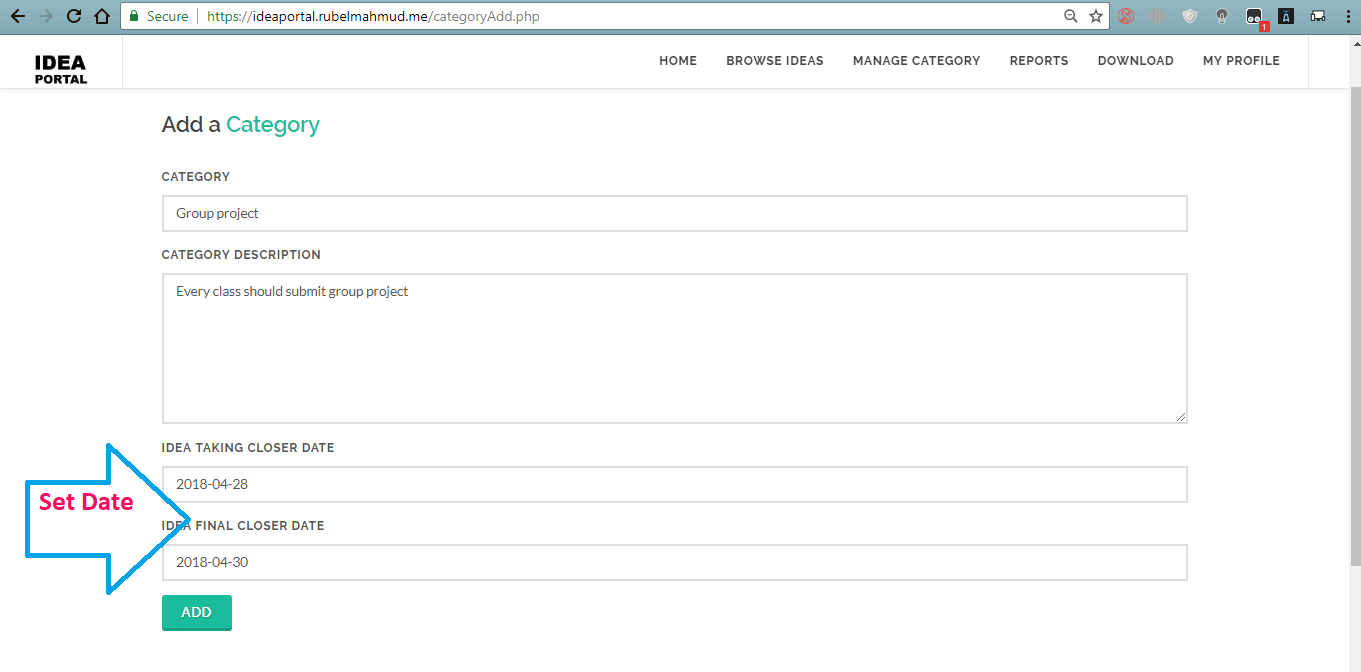
****

Figure: Set Category closure and final closure Date

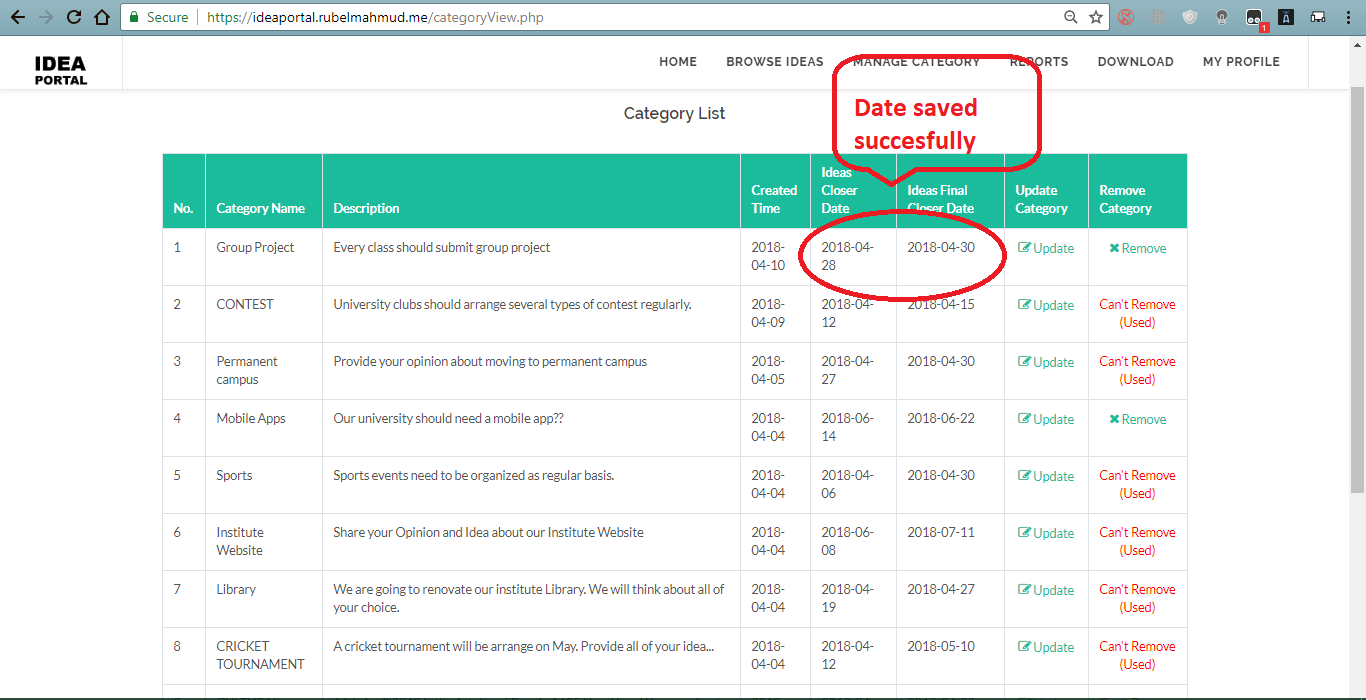
****

Figure: Category date saved successfully

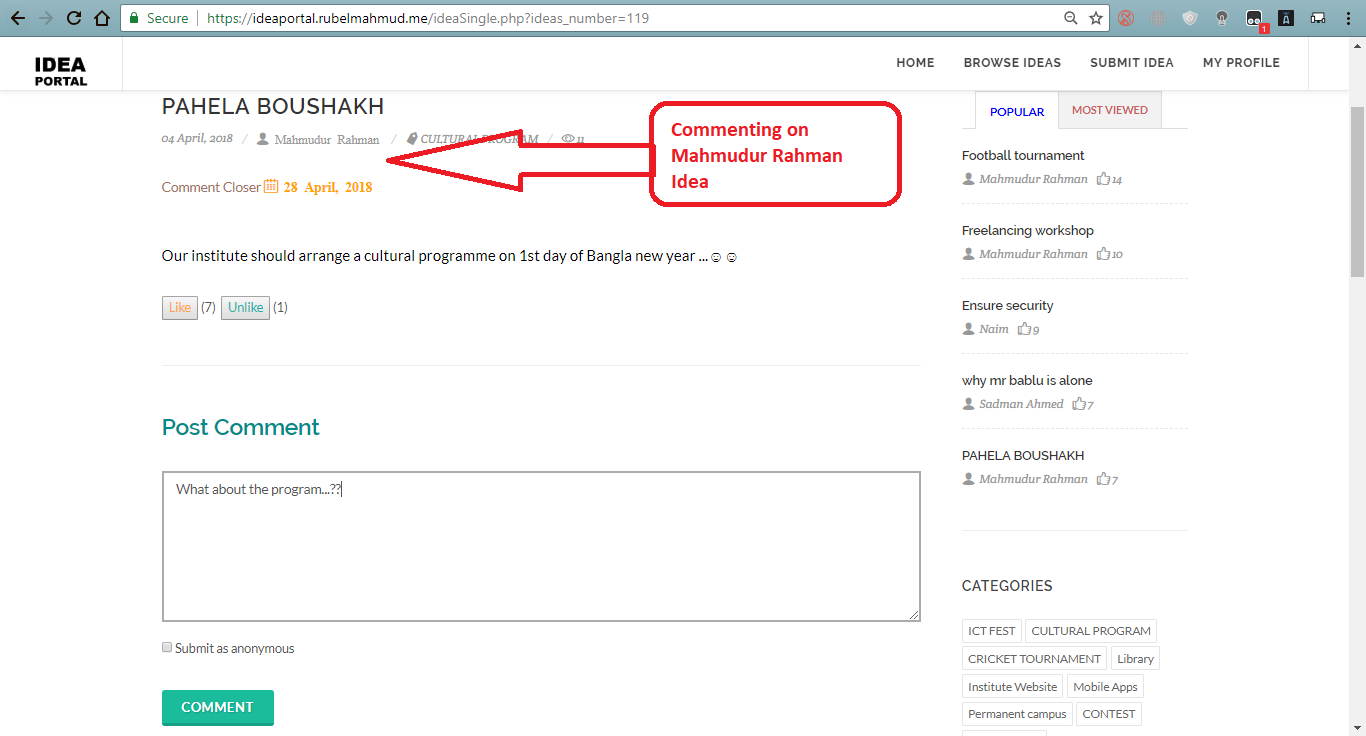
****

Figure: Commenting on a student Idea

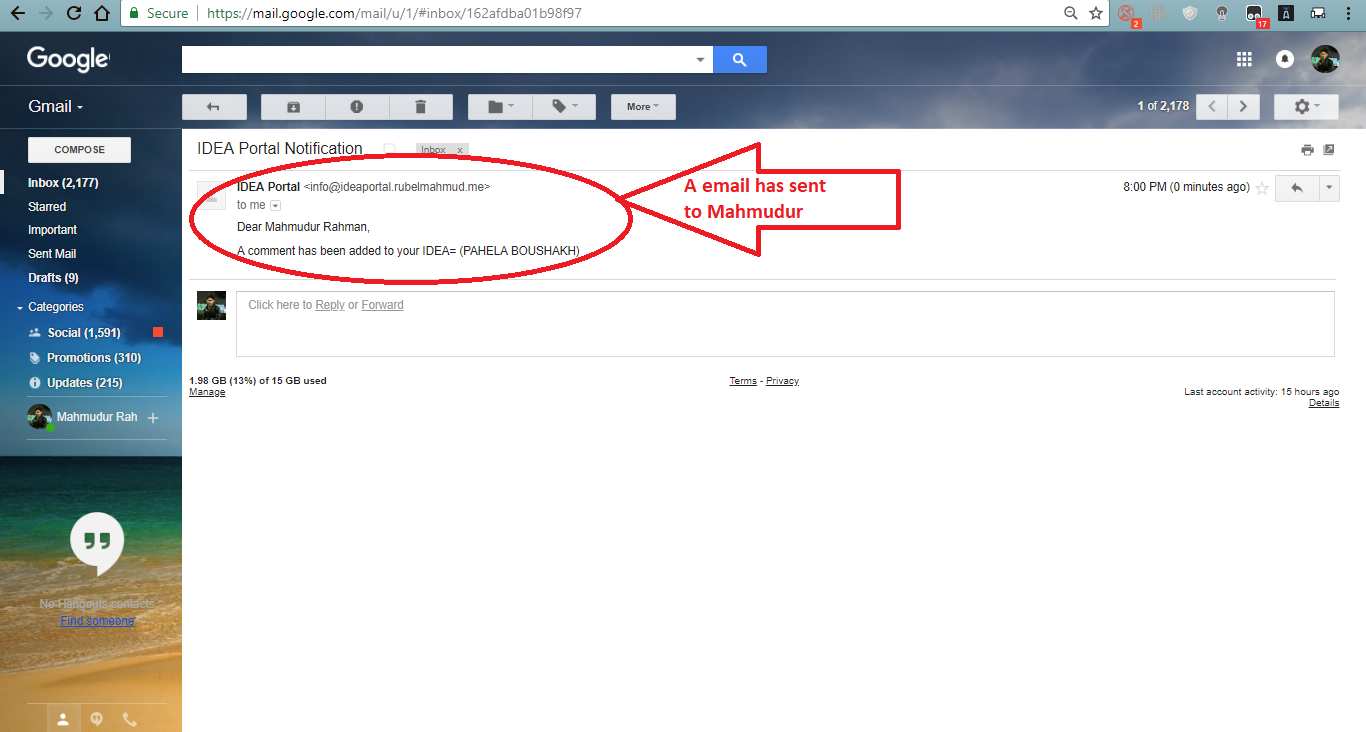
****

Figure: Idea author Received email notification

**Security Testing**

Security testing is most important for any enterprise web software. Security testing uncovers any flaws and bugs. Here we are going to test the following security testing-

* Session hijacking testing
* SQL injection testing
* Brute Force Attack
* Authentication Testing

**Test case for security test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 2.1 | Security testing (Brute Force Attack) | Using hydra tools run brute force attack | Brute force attack would not be successful to break system security | Brute force attack was unsuccessful and output is nothing | None |
| 2.2 | Security testing  (SQL Injection) | Using havij tools run SQL injection attack | The tools will not get the database and table | The attack was unsuccessful to get the system database | None |
| 2.3 | Security testing (URL Hijacking) | Copy a url and paste on other browser | URL will not work and user need to login again | Url does not and user is asked to login to system | none |
| 2,4 | Security testing  (Authentication) | Try to login to the system as unauthorized user | Unauthorized user will not be able to access the system | Unauthorized login attempt was unsuccessful | None |

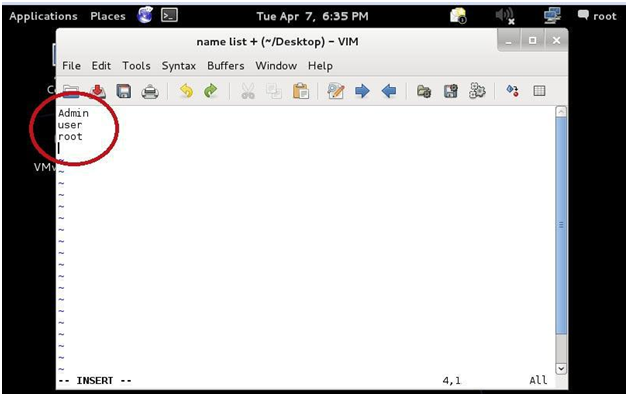


Figure: Brute force attack run

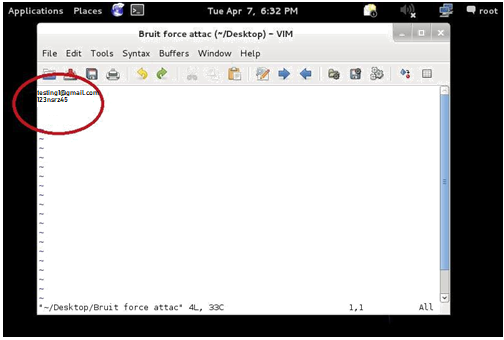


Figure: Provide Usermail and password

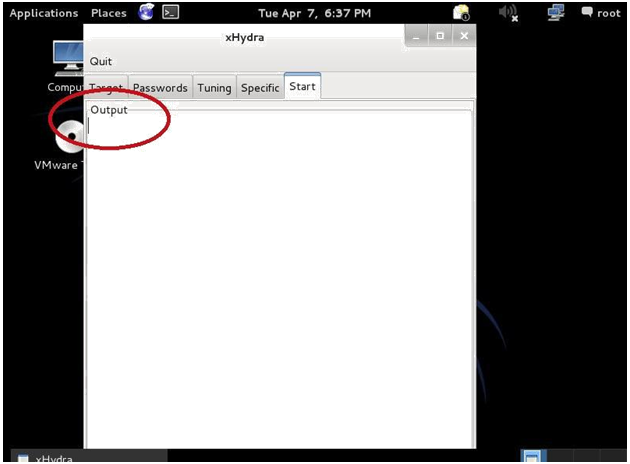


Figure: Brute force attack output

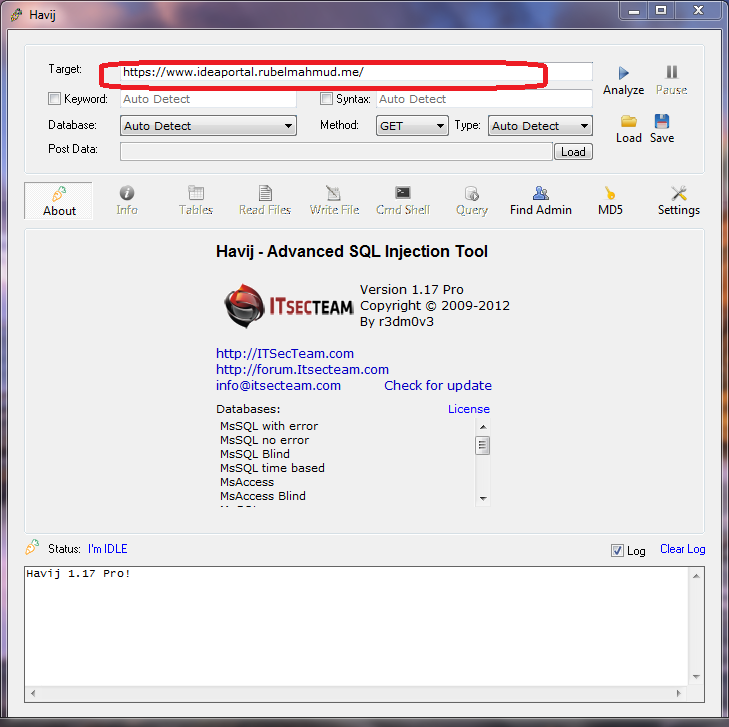


Figure: Running SQL injection tools



Figure: SQL injection failed

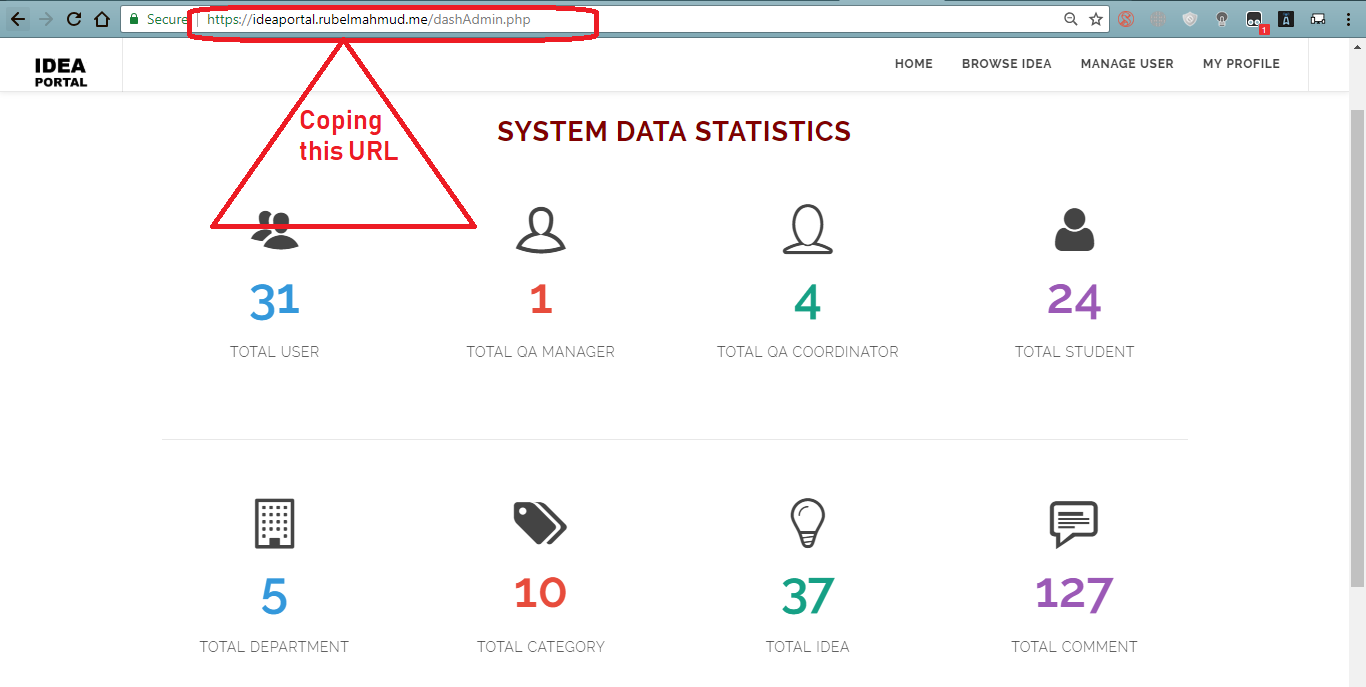


Figure: URL hijacking test

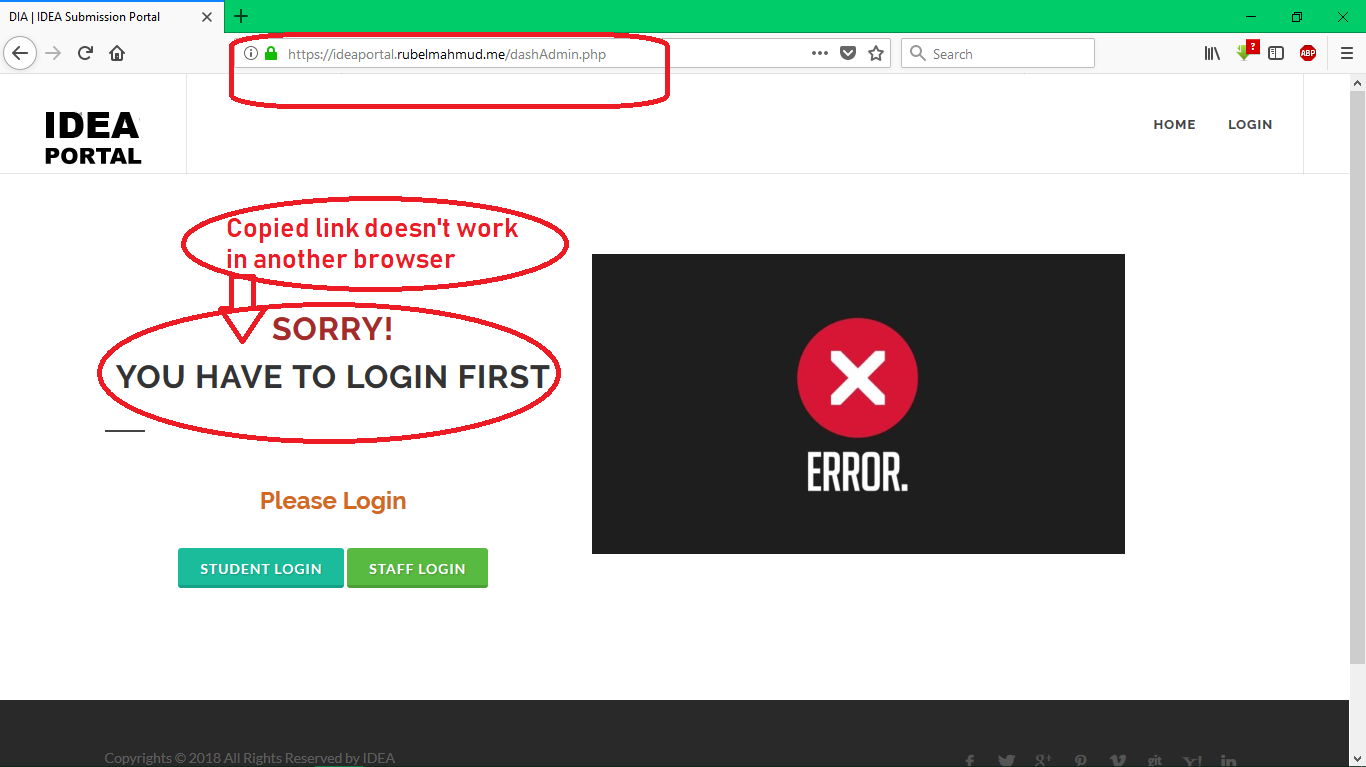


Figure: URL hijacking test result

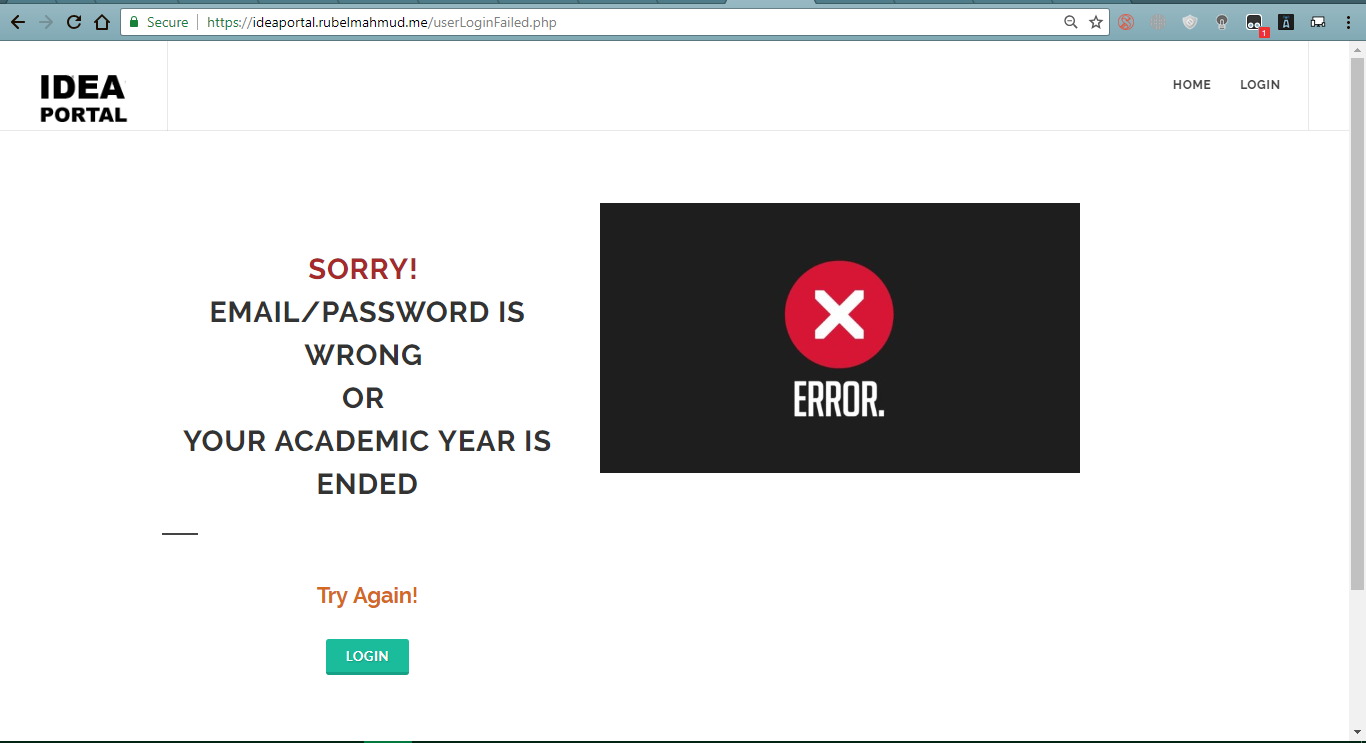
****

Figure: Unauthorized user attempt

**Performance Testing**

Performance testing is about testing software capabilities of performing under an amount of workload. The developed system is capable performing various functions.

* **Load testing**
* **Stress testing**
* **Scalability testing**

**Test case for performance test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 3.1 | Performance Testing  (Load Performance) | Run the system on online load performance tools | The system load performance will be successful | The system has passed the load performance test | None |
| 3,2 | Performance Testing (Stress Performance) | Run the system on a online stress performance tools | The system stress performance will be successful | The system has passed the stress performance test | None |

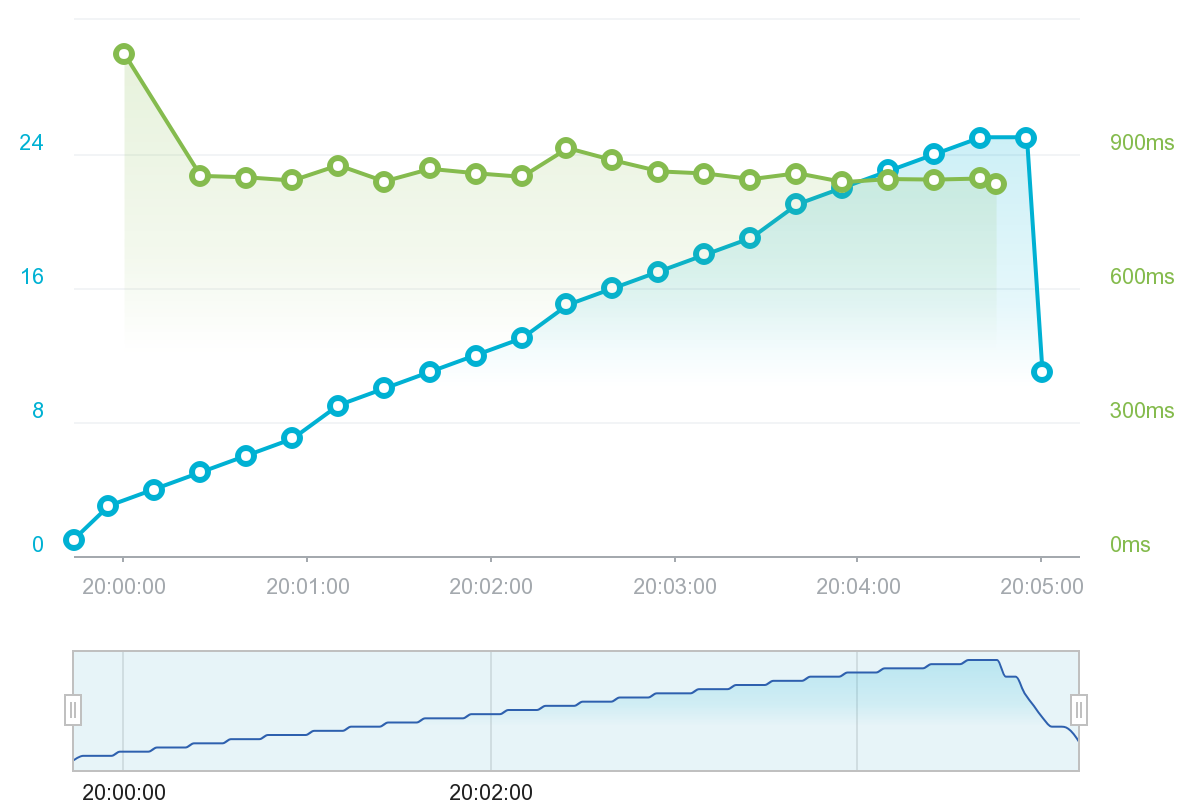
****

Figure: Load performance testing

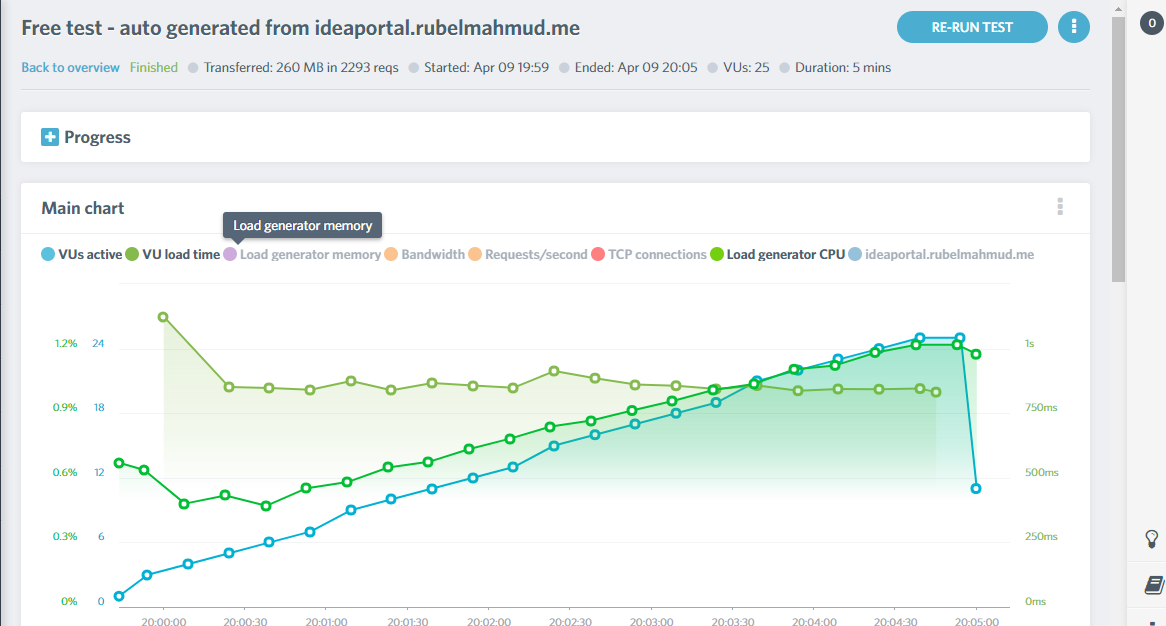
****

Figure: Load performance testing

****

Figure: Load performance testing

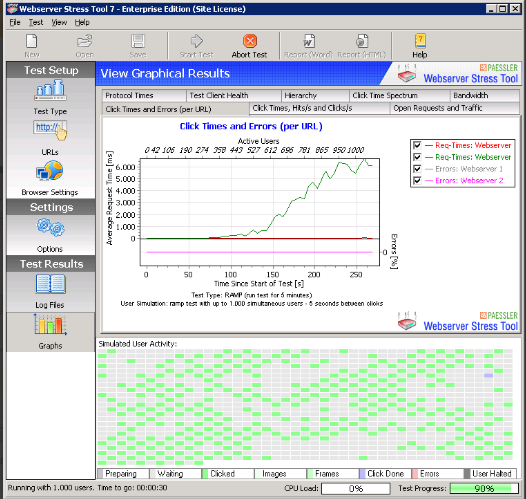
****

Figure: Stress performance testing

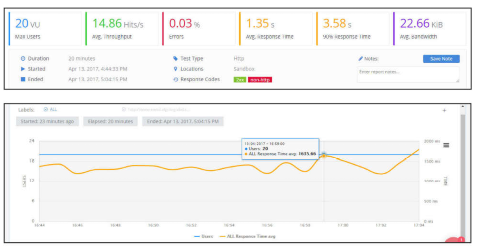
****

Figure: Stress performance testing

**Responsive Testing**

Responsive testing is applied to test a web software optimal viewing experience across many devices by resizing, scrolling, panning. The developed system is tested across different type of devices with different pixel. Responsive testing is executed on the following devices-

* Mobile
* Notebook
* Tab

**Test case for responsive test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 4.1 | Responsive  Testing (Mobile) | Running the system in mobile | The system will be appear in a mobile view | The system has appeared in a mobile view without any problem | None |
| 4.2 | Responsive  Testing (Tab) | Running the system in a tab | The system will appear in a tab view screen | The system has appeared in a tab screen perfectly | None |
| 4.3 | Responsive  Testing (Notebook) | Running the system in a Notebook | The system will appear in a Notebook successfully | The system has appeared in notebook perfectly | None |



Figure: Mobile responsive testing

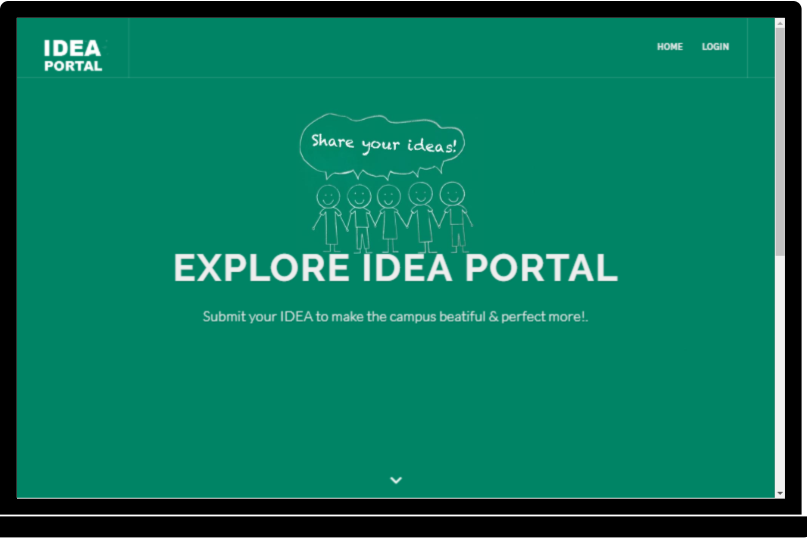


Figure: Notebook responsive testing

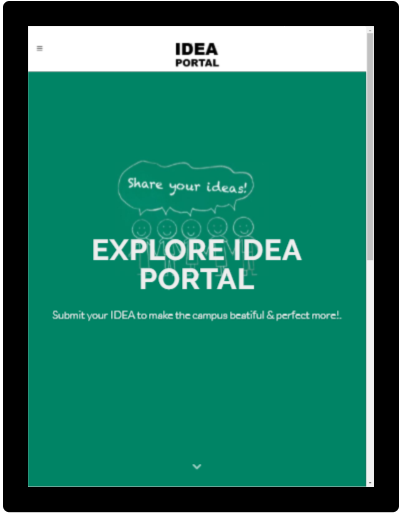


Figure: Tab responsive testing

**Platform Compatibility Testing**

The developed system is browse in different web browser of different Operating System. The following testing is conducted.

* Windows
* MAC
* Linux

**Test case for Platform Compatibility test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 6.1 | Platform Compatibility Testing (Windows) | The system is testing by running on  Windows operating system | The system will not face any problem to run in windows operating system | The is running on windows operating system smoothly | None |
| 6.2 | Platform Compatibility Testing (MAC) | The system is testing by running on  MAC operating system | The system will not face any problem to run on MAC | The is running on MAC without any problem | None |

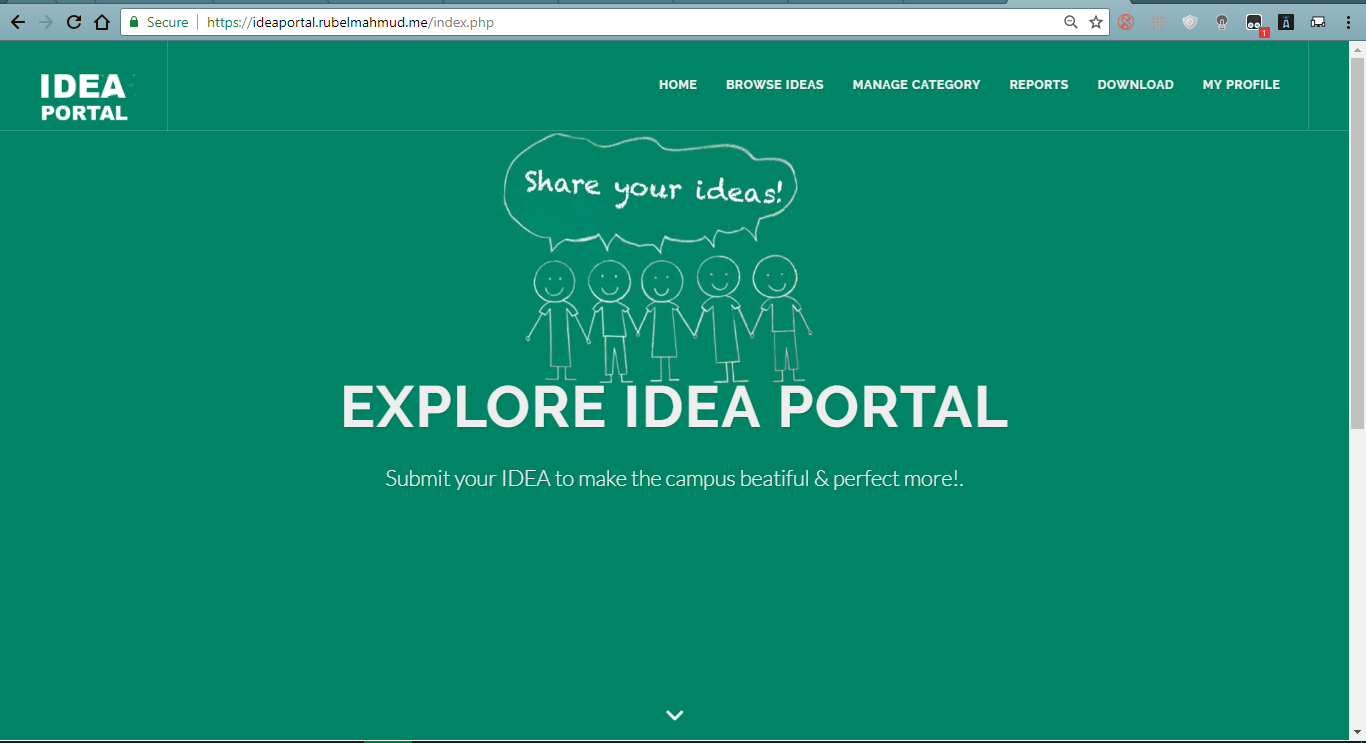


Figure: Windows Operating System Compatibility Testi

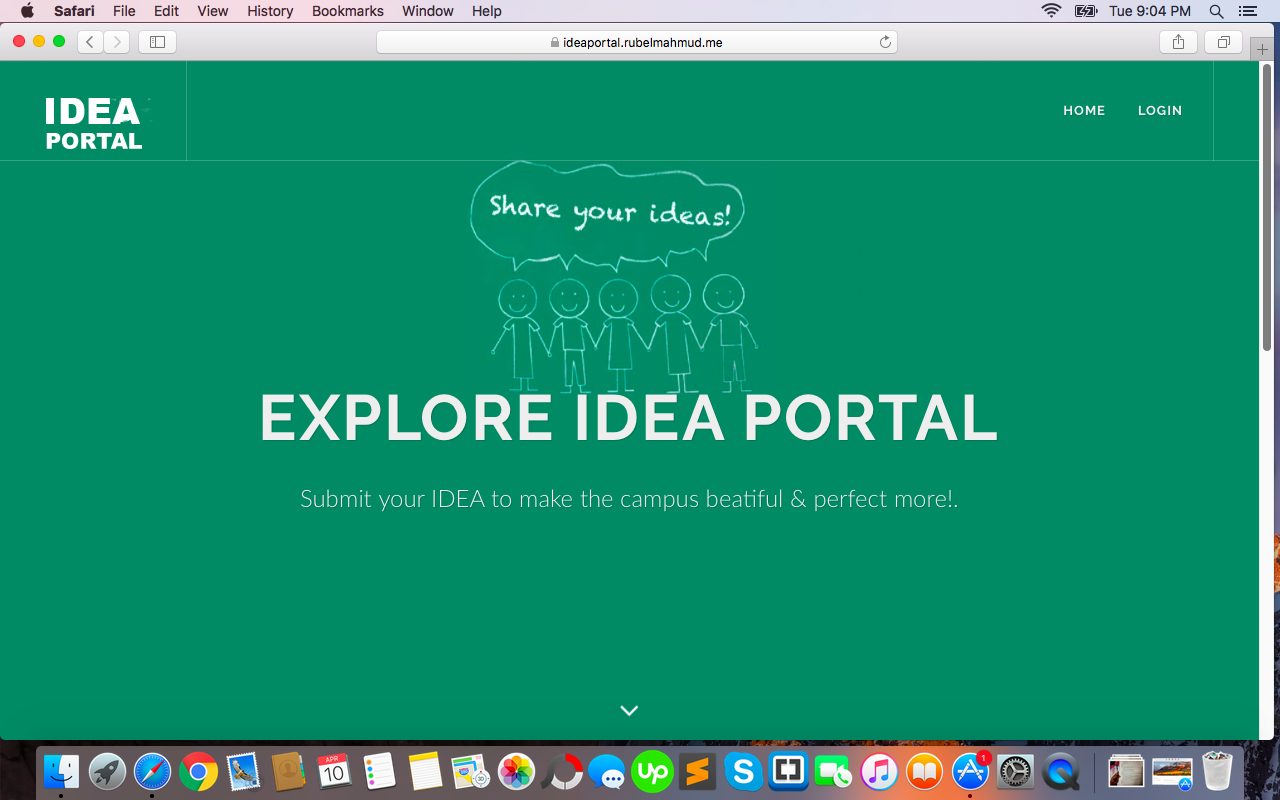


Figure: MAC Operating System Compatibility Testing

**Validation testing**

Validation ensures the data provided according to system requirements. Some condition is used to validate system data. The system will accept data if it matches with condition. The system was developed with confirming proper validation. There are some validation testing-

* Login validation
* Date validation
* Text input validation
* Idea submit validation
* Category delete validation

**Test case for Validation test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 7.1 | Validation  Testing  ( Login) | Providing Usermail without proper mail format | The system will ask for a mail address | The system has asked for proper mail format usermail | None |
| 7.2 | Validation  Testing  (Date) | Selecting Idea closure date and final closure date | User would not able to select previous as closure date and final closer date less than closure date | User failed to select previous any date for idea closure date and failed to select final closure date less than idea closure date | None |

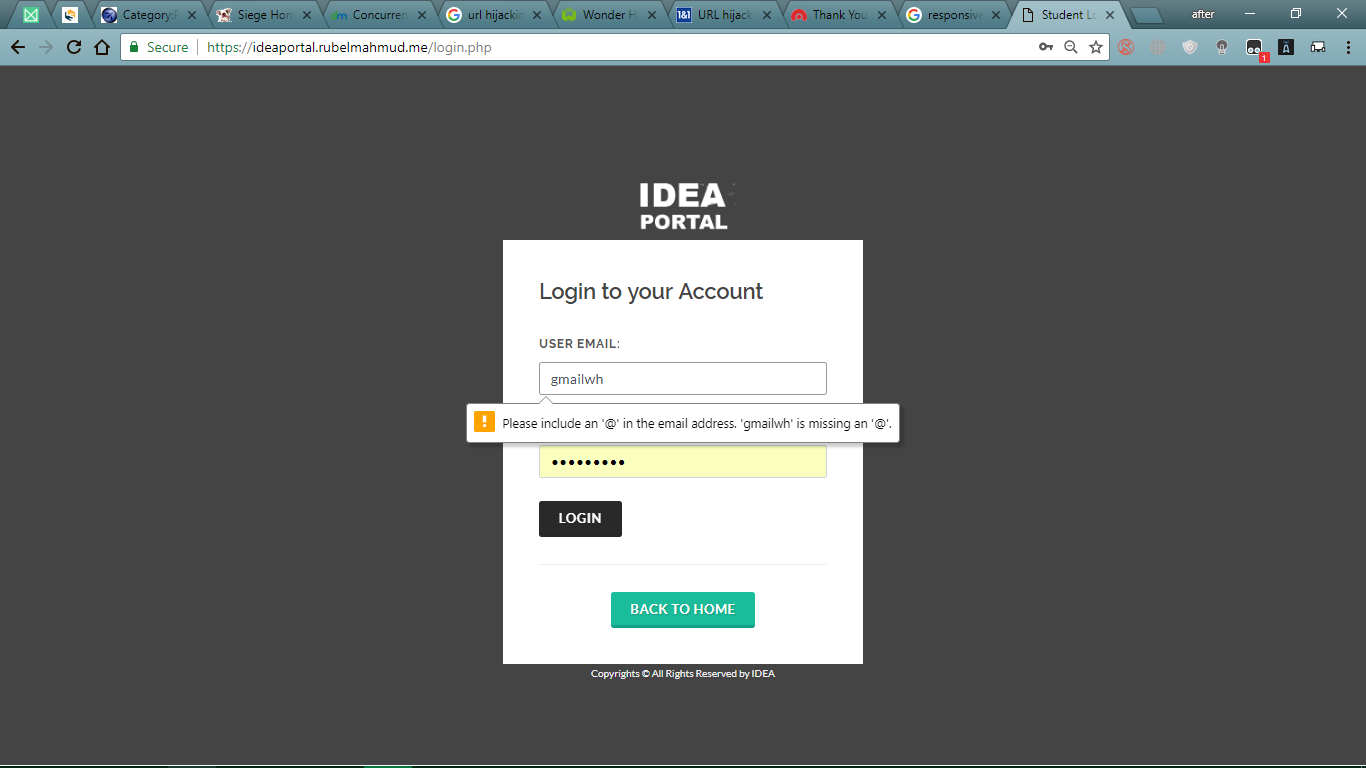


Figure: Login Usermail Validation

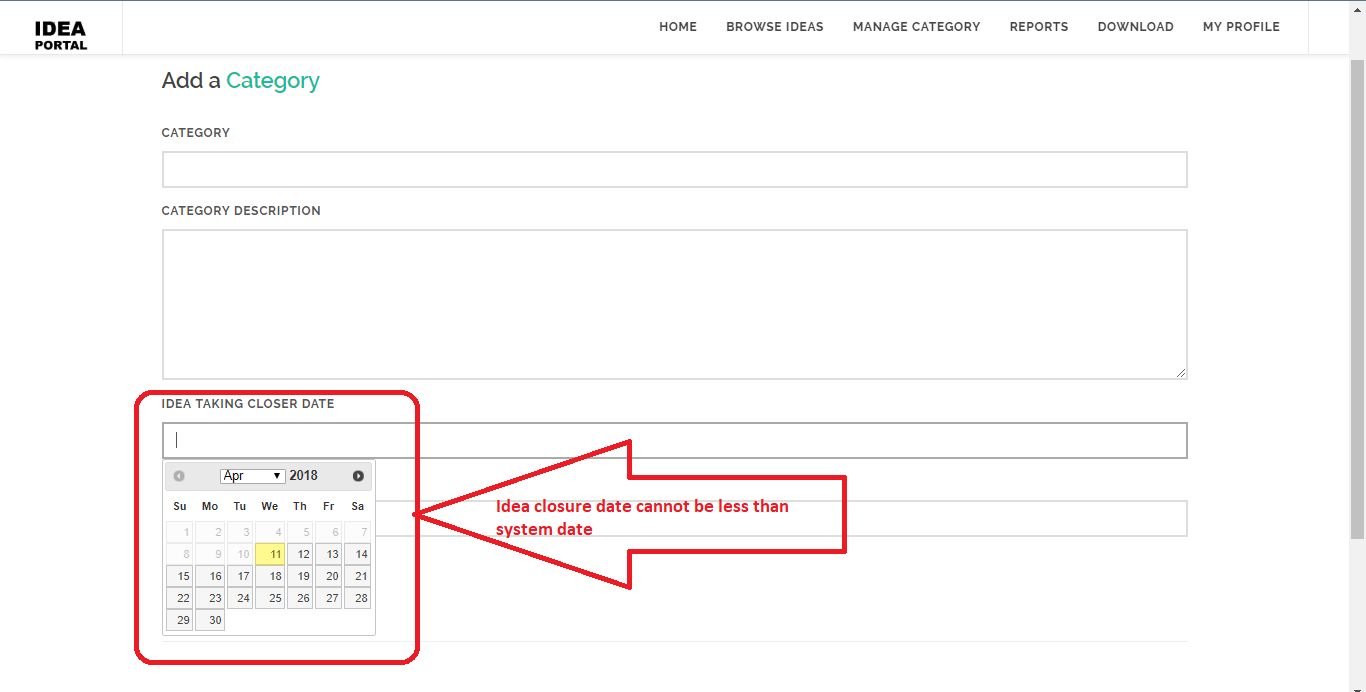


Figure: Closure date Validation

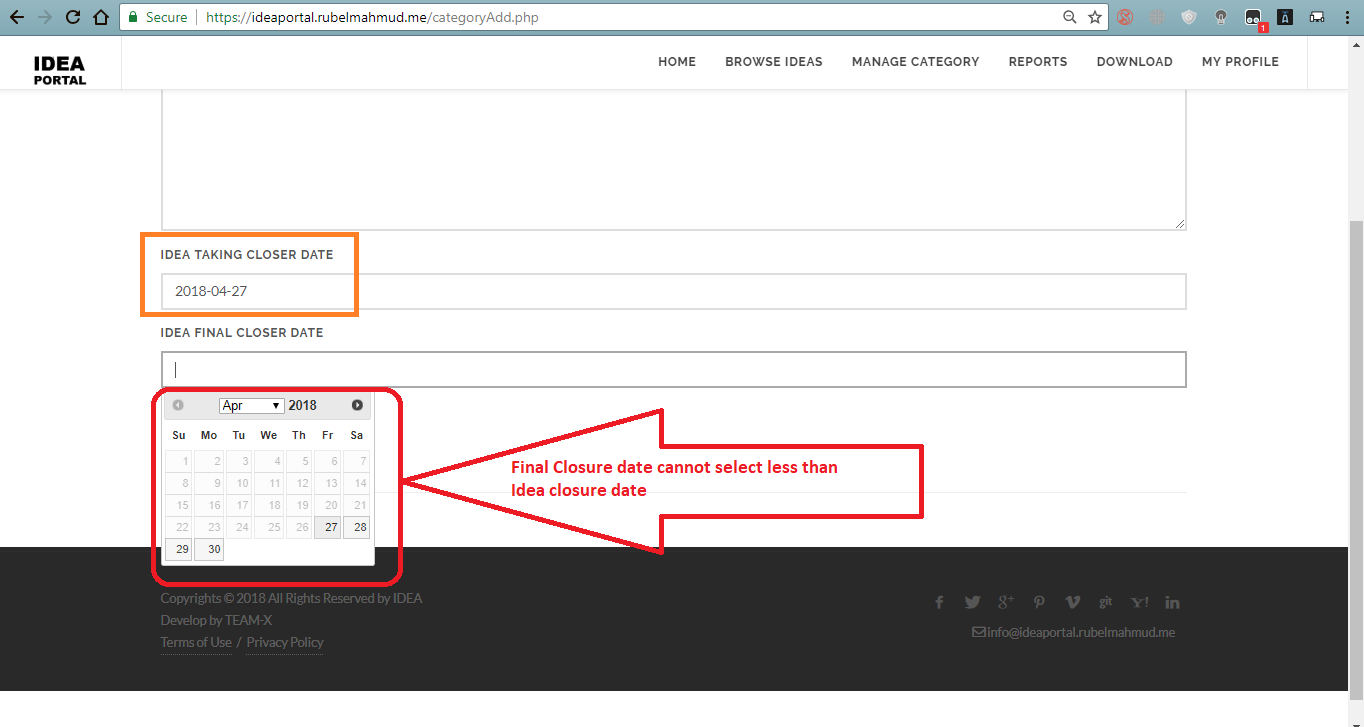


Figure: Final Closure date Validation

**Cross Browser testing**

Cross browser testing is about testing web system running on different browser to check all functionalities work properly. Here I have tested the developed on following browser**-**

* Google Chrome
* Mozilla Firefox
* Internet Explorer
* Safari

**Test case for Cross browser test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 8.1 | Cross Browser Testing (Google Chrome) | Running the system on Google Browser | The system will run perfectly | The system has run without facing any problem | None |
| 8.2 | Cross Browser Testing  (Mozilla Firefox) | Running the system on Mozilla Firefox Browser | The system will run perfectly | The system has run without facing any problem | None |
| 8.3 | Cross Browser Testing  (Safari) | Running the system on Safari Browser | The system will run perfectly | The system has run without facing any problem | None |

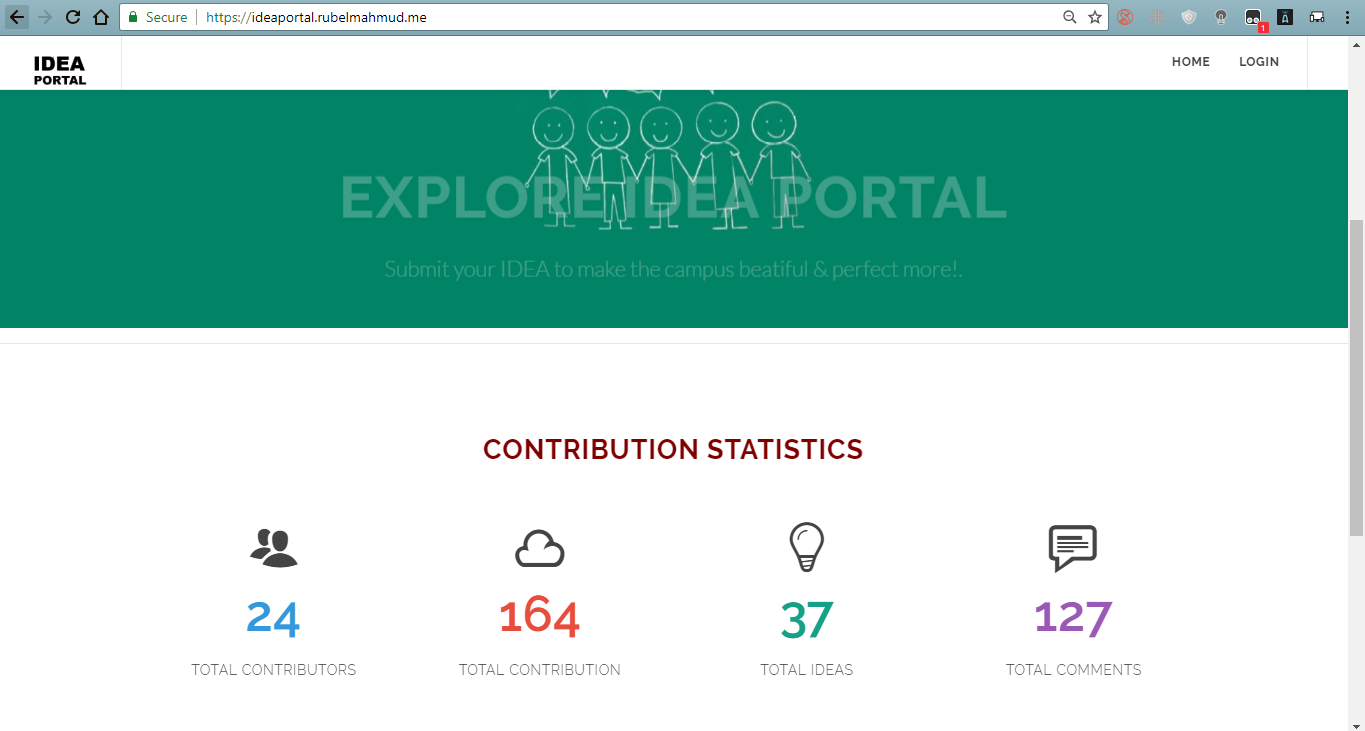
****

Figure: Google Chrome browser capability testing

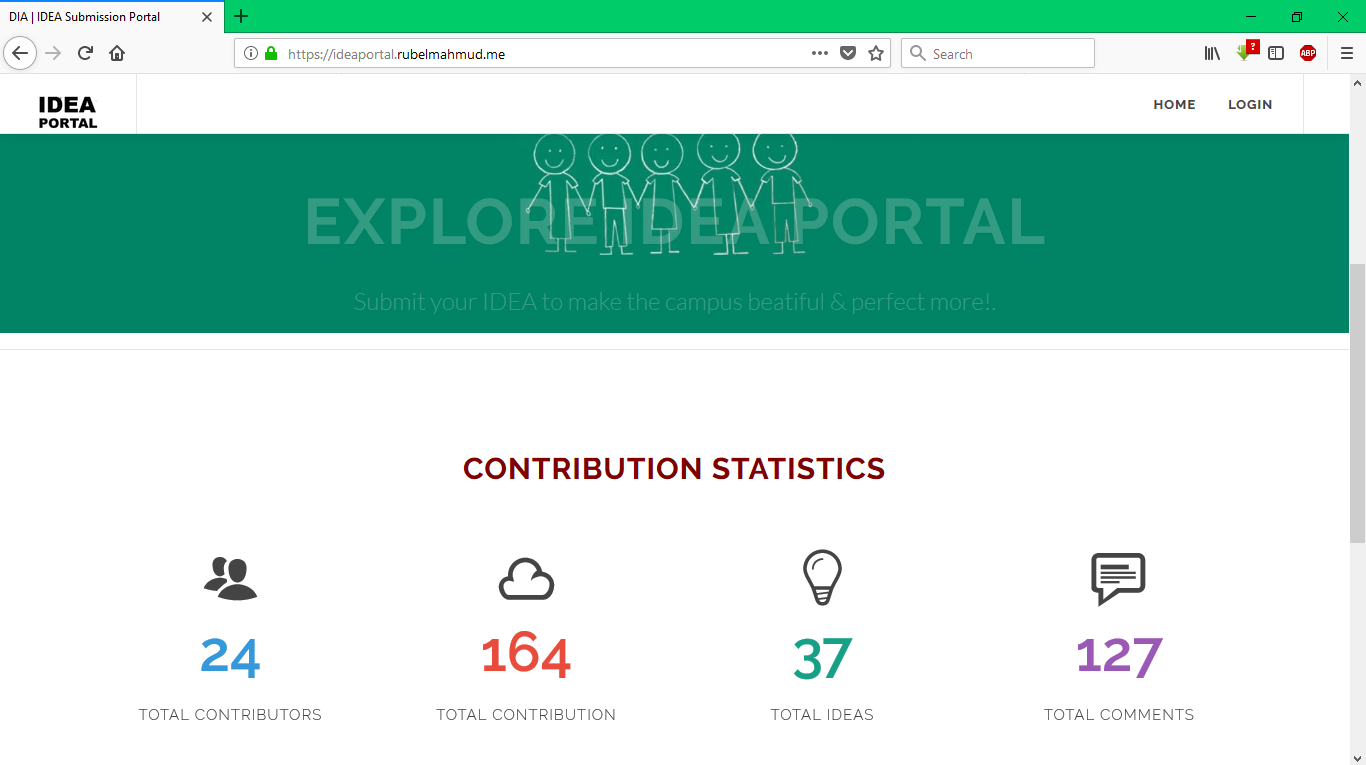
****

Figure: Mozilla Firefoxbrowser capability testing

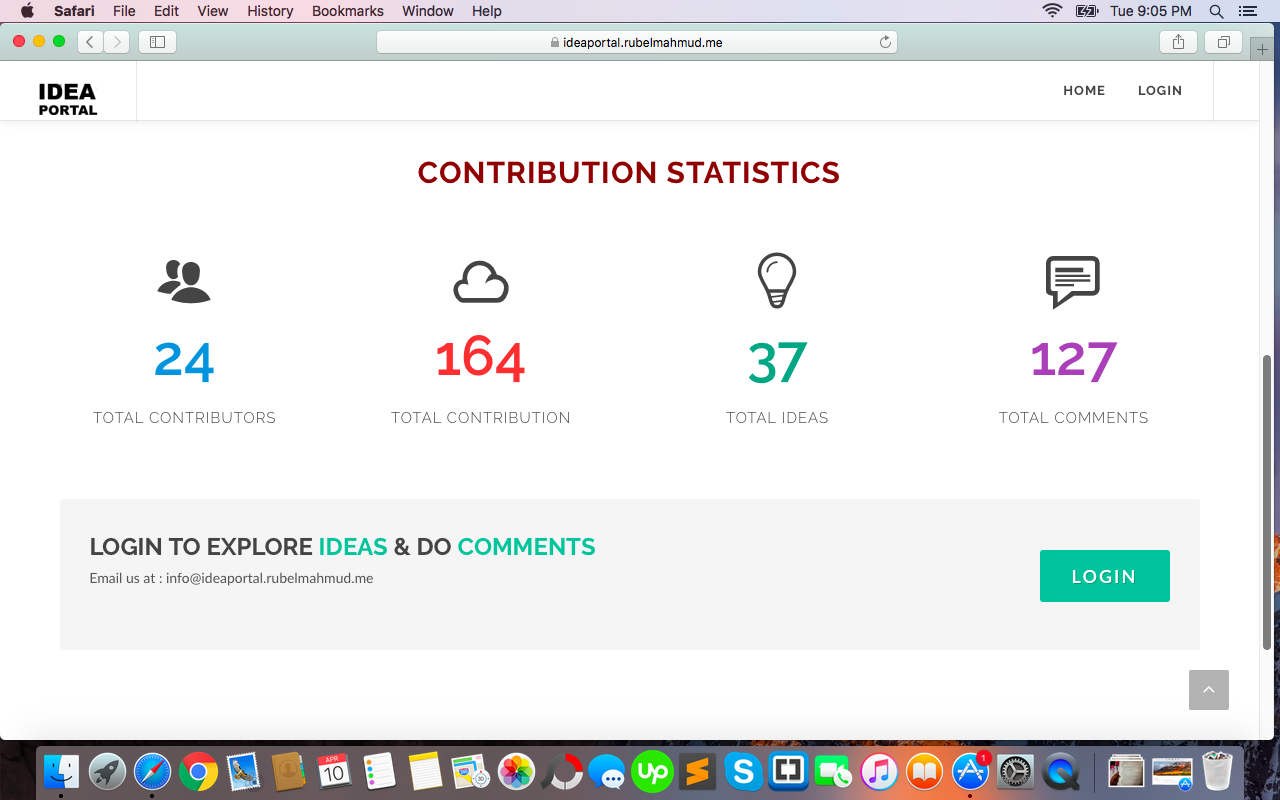
****

Figure: Safaribrowser capability testing

**Integration testing**

System integration confirmed according to its requirements. Integration testing test integration between various pages of a the system. The developed system integration are tested below-

**Test case for Integration test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 9.1 | Integration  Testing | User click on exception report | The system will redirect to exception report of the system | The system has successfully redirected to exception report page | None |
| 9.2 | Integration Testing | Clicking on My Profile Button | It will redirect to user profile information | The test was successful and redirected to user profile page | None |

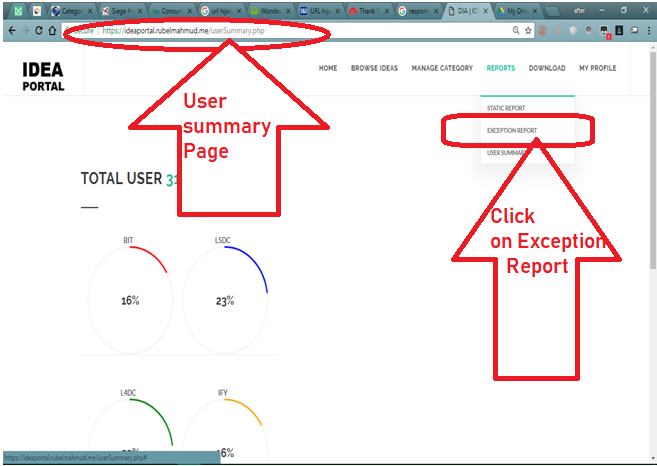
****

Figure: Integration between pages

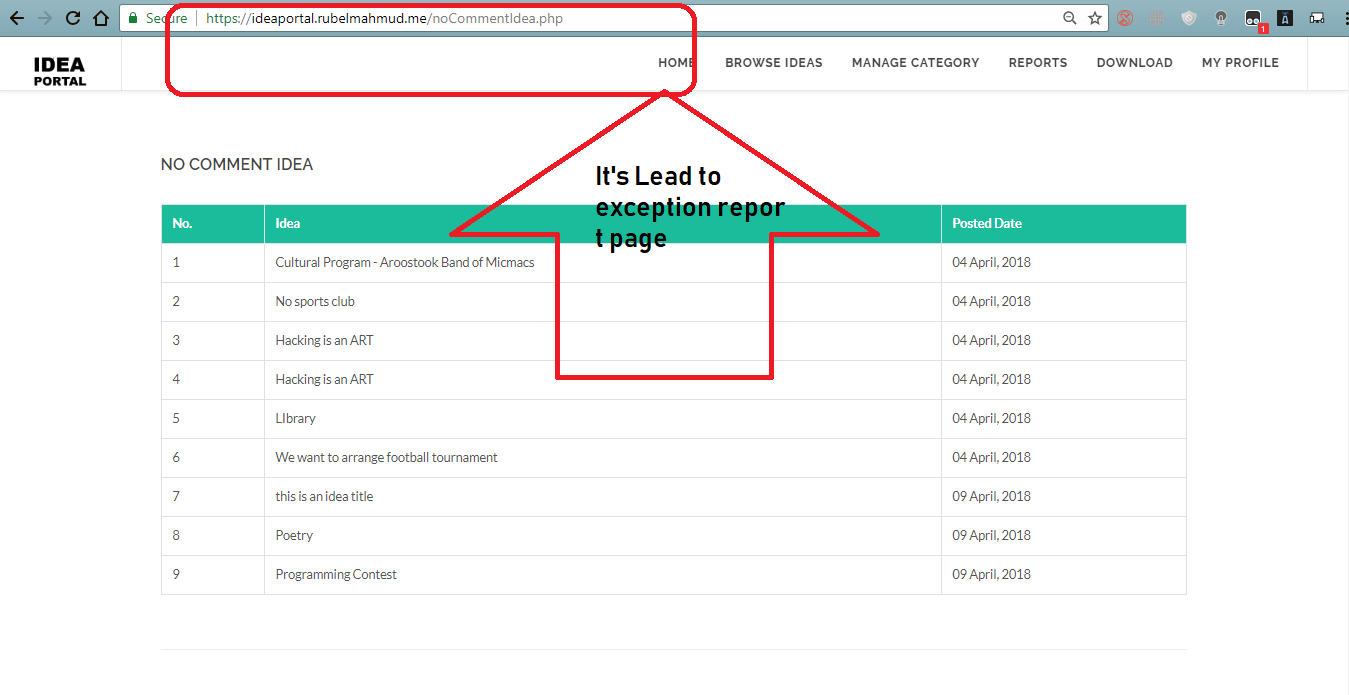
****

Figure: Integration between pages

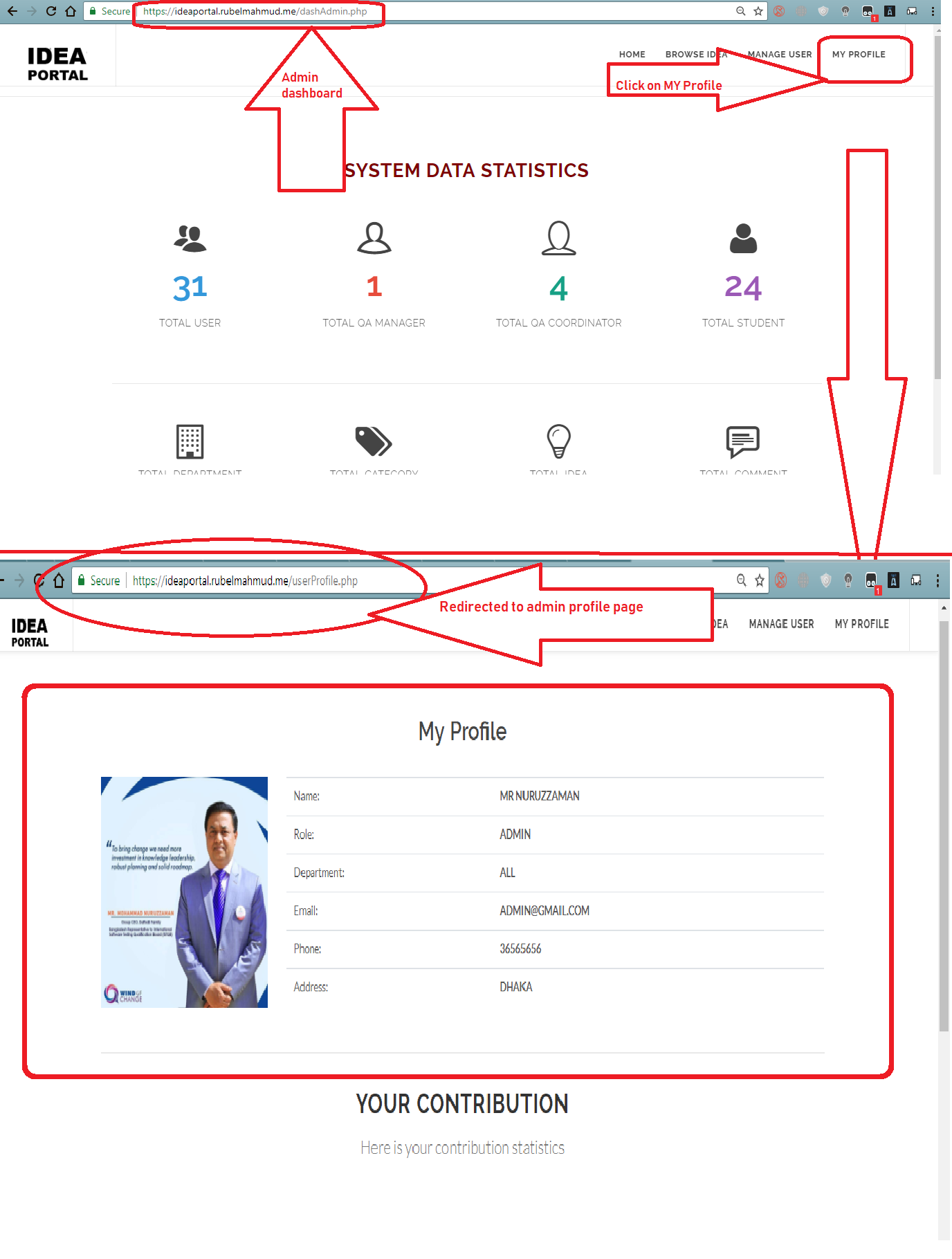
****

Figure: Integration between pages

**Usability testing**

**Usability testing is about testing the functions how easy to use. It is a measure evaluates a system how easily it can be properly used by the end user. The following testing are performed-**

**Test case for Usability test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 10.1 | Usability testing | View logged in user name | It will be easy to see logged in username by user himself | User can see his user name and logout button easily | None |
| 10.2 | Usability Testing | Idea list usability | Idea list will provide information about idea and other information about idea | Idea list contain ideas with their submitted date, author name, closure date and final closure date. | None |
| 10.3 | Usability Testing | Login usability | When user try to login to the system, they will find to login with their role | The system shows in which role user want to login | None |
| 10.4 | Usability Testing | View an idea and see it’s usability | The idea information page will contain idea information and option for other ideas. | The idea information page appear with idea name, submit date, total view, category name, author name, final closure date, most viewed idea and popular idea. | None |

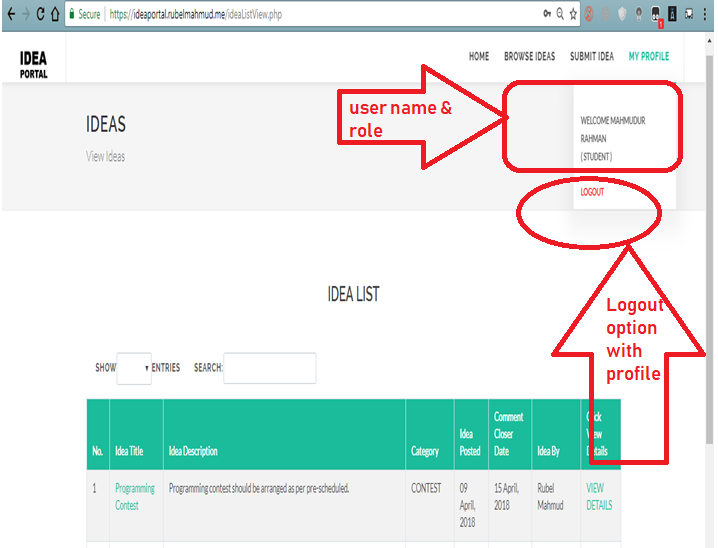
****

Figure: Usability of the system

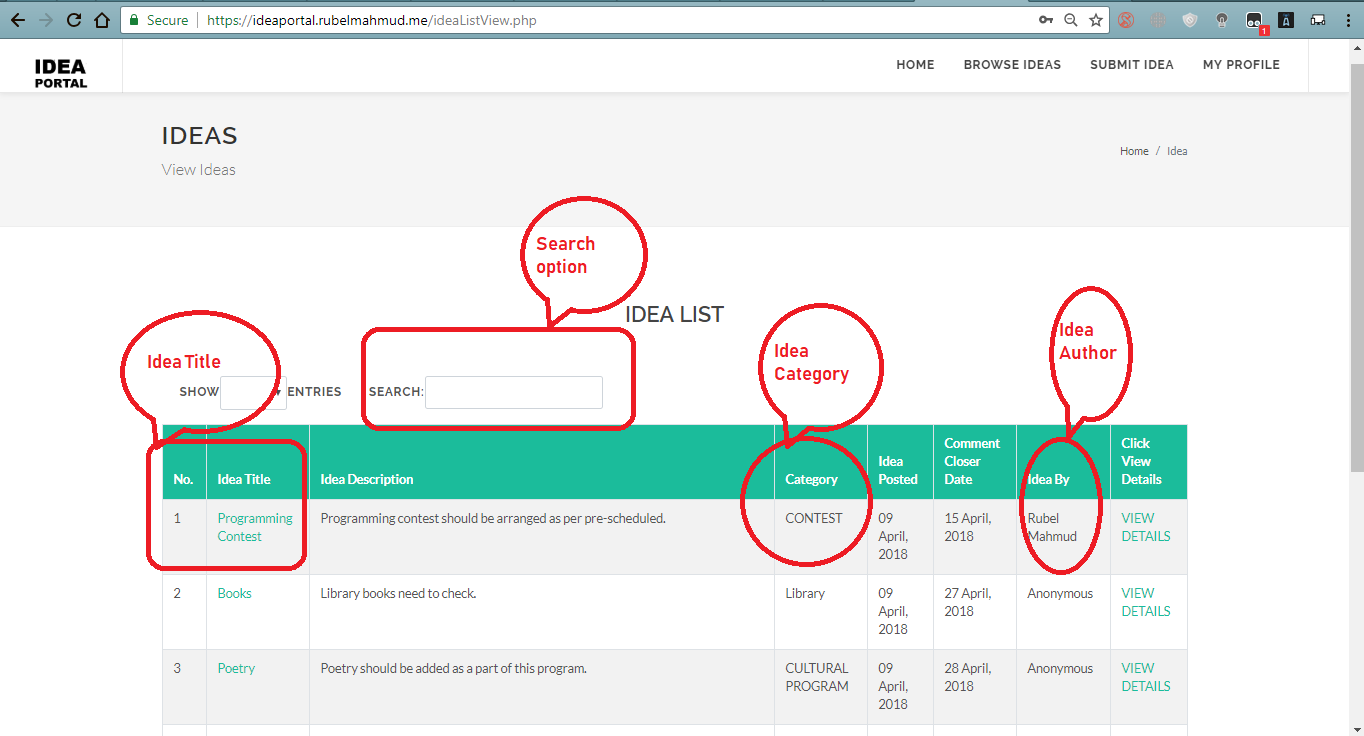
****

Figure: Usability of the system

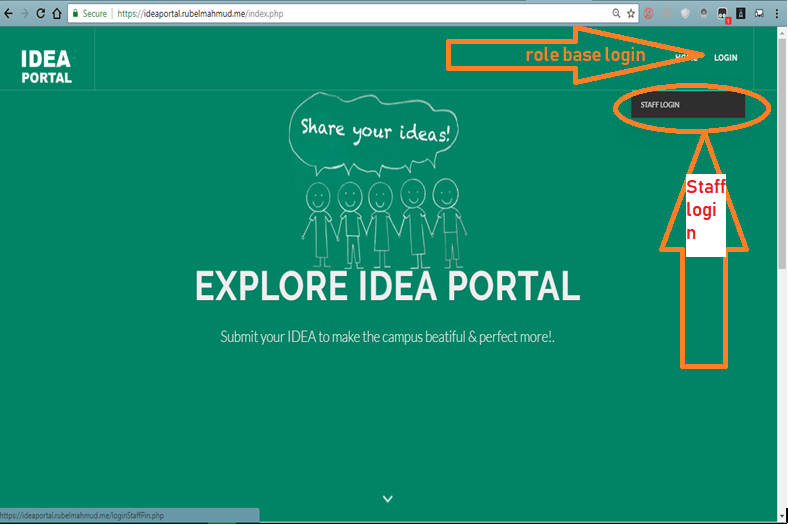
****

Figure: Usability of the system

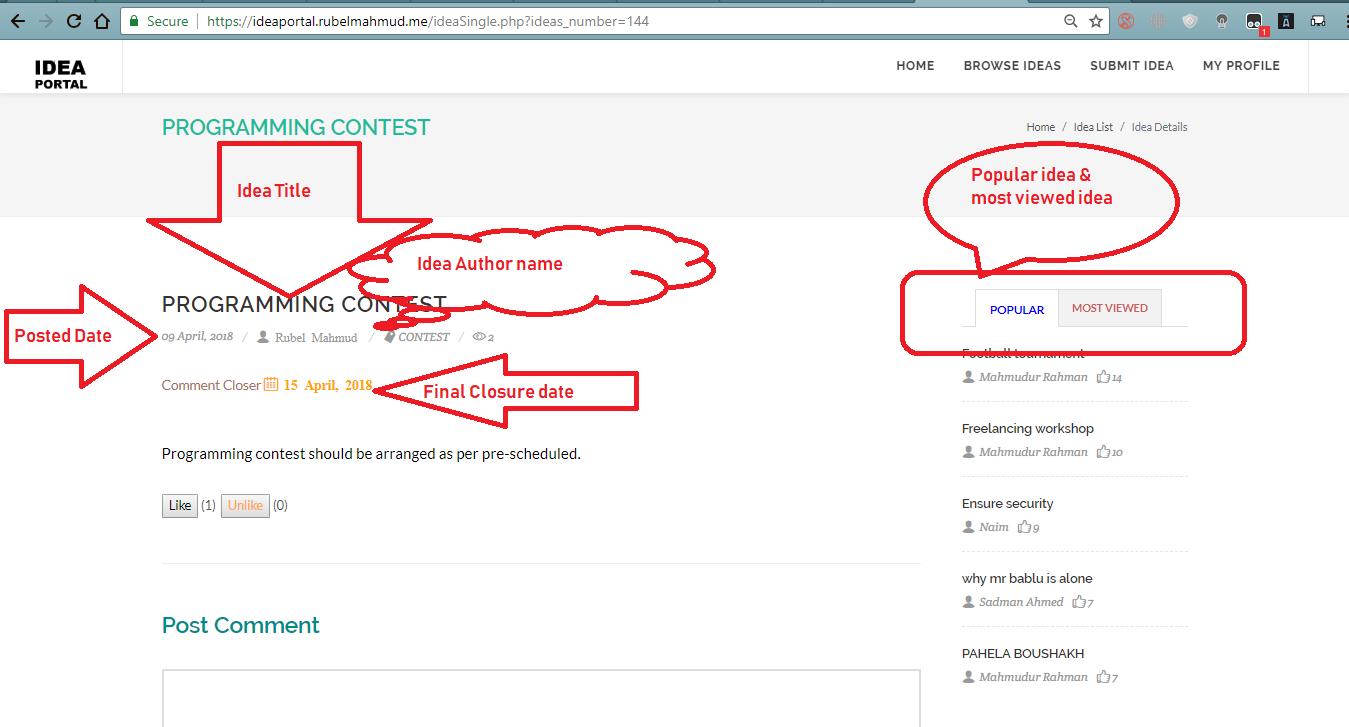
****

Figure: Usability of the system

**Role based access Testing**

The system is role based system. Every user doesn’t get access to all data of the system. Few users can perform limited number of task that is assigned for them. In this user role based access is tested-

* Admin access
* QAM access
* QAC access
* Student access
* Supervisor access

**Test case for Role Based test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test Title | Test process | Expected Result | Actual result | Action Taken |
| 11.1 | Role based access Testing | QAM login to the system selecting staff login interface and providing pin he gets access to the system providing matching Usermail and password | The system will lead to QAM dashboard. | QAM logged into th system and got access the QAM role based information | None |

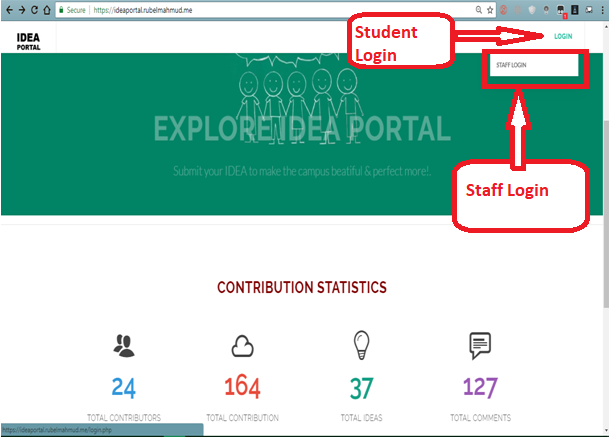


Figure: QAM login access

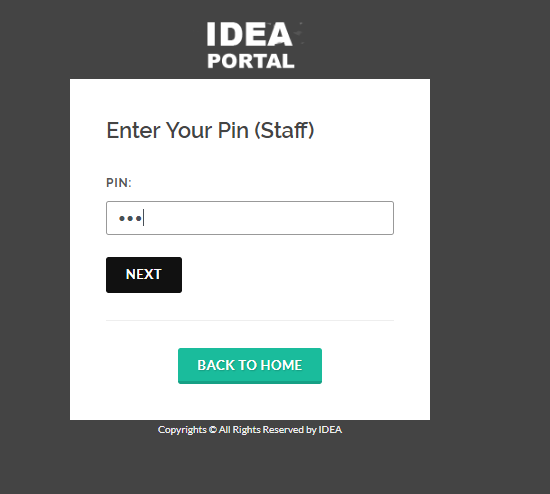


Figure: QAM provide pin

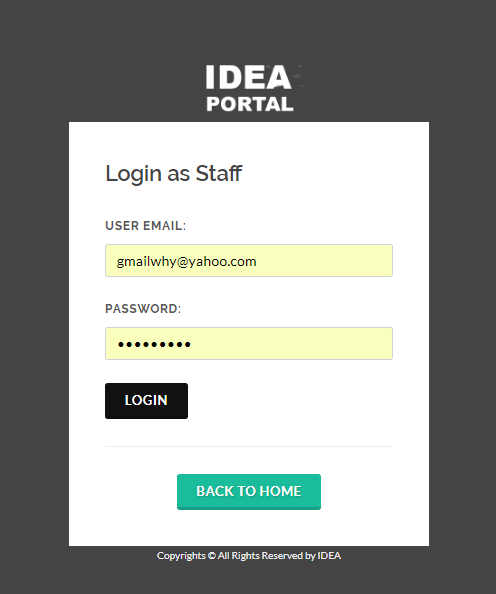


Figure: QAM login information

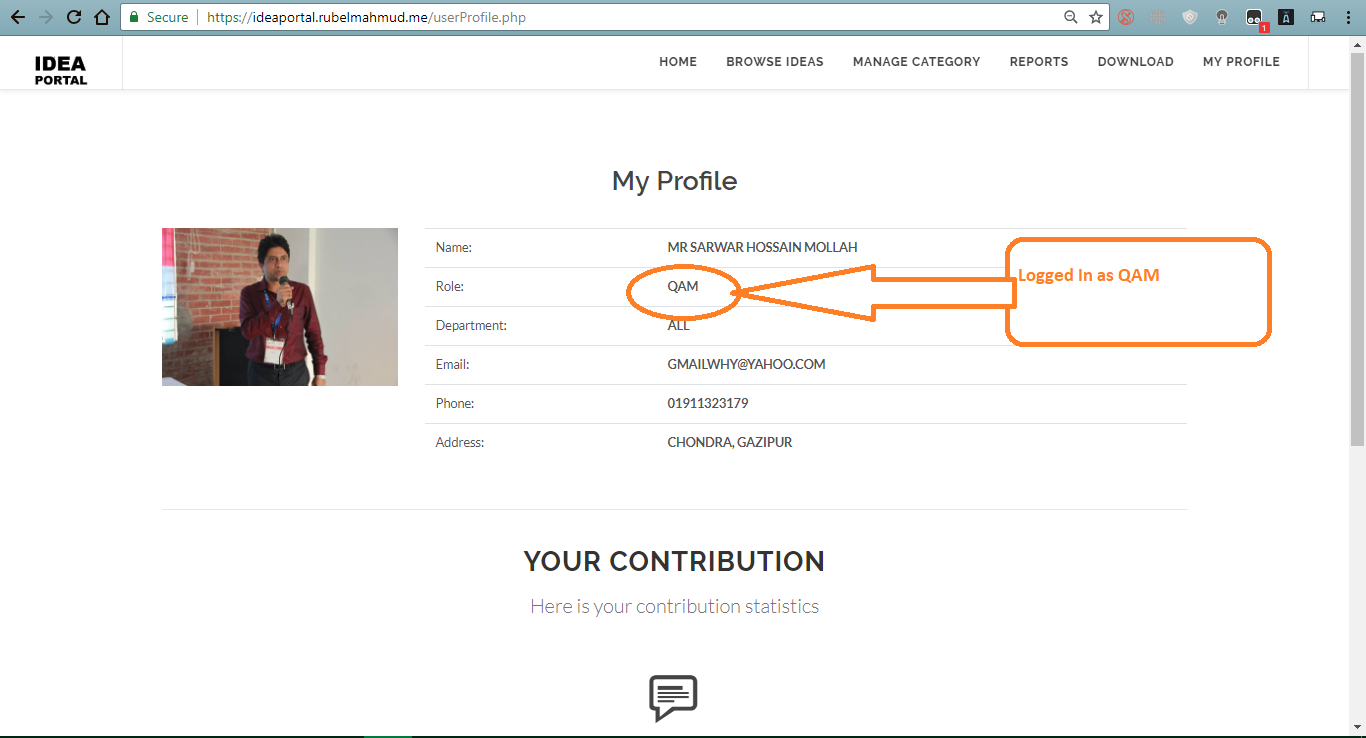


Figure: QAM get access to separate information

**Crowd Testing**

Crowd testing means testing software by a large number of people. People use the software and share their experience.

**Sprint Backlog testing list**

Sprint backlog is the list of tasks that was identified by the scrum master. Sprint backlog describe which tasks need to be completed. Sprint backlog testing tests scrum sprint task list. It confirms that development of the software was done with the sprint backlog and meeting minutes. Sprint backlog testing list is there-

# Test Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Test Name** | **Description** | **Outcome** | **Result** |
| 1 | **Functional Testing** | Testing the function according to the requirements of the system | The functions works perfectly according the requirements | Passed |
| 2 | **Security Testing** | Testing the security threats of the system. | The system is well secured from any kind of threat | Passed |
| 3 | **Performance Testing** | Testing various performance against different situation | The system performance is as expected | Passed |
| 4 | Responsive testing | Testing the responsiveness of the system in different sizes of devices | The system is compatible in different sizes of devices | Passed |
| 5 | **Integration Testing** | Testing integration between different pages | The system integration is perfect | Passed |
| 6 | Role based access to system testing | Testing different role based access to the system | The particular user get access to the system based on their role | Passed |
| 7 | **Usability testing** | Testing how the system to complete any kind of system related task | The system interface is easy to find and accomplish any task | Passed |
| 8 | Validation testing | Checking validation  In different part of the system | The system validation works perfectly | Passed |
| 9 | Platform Compatibility testing | System capability of running in different operating system | The system run smoothly in all operating system | Passed |
| 10 | **Database testing** | The system database was tested by providing data, updating data and deleting data | The system database works properly with the system | Passed |
| 11 | Cross Browser Testing | Running the system in different browser | The system run perfectly in different browser | Passed |
| 12 | Crowd Testing | The system tested by a group of people | The user could use and understand the system properly | Passed |
| 13 | Product Backlog Testing | Identify the task was completed and tested according to product backlog | The system functionalities was completed and tested according to product backlog | Passed |