Table of Content

1.Introduction **3**

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 3

1.5 References 3

2.Overall Description **4**

2.1Product Perspective 4

2.1.1.System Interface 4

2.1.2.User Interface 4

2.2 Product Functions 4

2.2.1.User Functions 4

2.2.2.Student Functions 4

2.3 User Classes and Characteristics 4

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 5

2.6 Assumptions and Dependencies 5

3.External Interface Requirements **5**

3.1 User Interfaces 5

3.2 Hardware Interfaces 6

3.3 Software Interfaces 6

3.4 Communications Interfaces 6

4.System Features **6**

4.1 Description and Priority 6

4.2 Stimulus/Response Sequences 6

4.3 Functional Requirements 7

4.3.1 Administrator Functions 7

4.3.1.1 Add User 7

4.3.1.2 View Applicants 7

4.3.1.3 View Students 7

4.3.1.4 Change Password 7

4.3.2 Student Functions 7

4.3.2.1 Profile 7

4.3.2.2. Apply Room 7

4.3.2.3. Status of Application 8

4.3.2.4. View History 8

4.3.2.5. Change Password 8

5.Other Nonfunctional Requirements **8**

5.1Performance Requirements 8

5.2Safety Requirements 8

5.3Security Requirements 8

5.4Software Quality Attributes 8
5.5Business Rules 9

6.Usage Scenario 9
6.1.Use case Diagram 9

7.System Design 10
7.1.Activity Diagram 11
7.1.1 For admin 11
7.1.2 For User 12
7.2 Class diagram 12
7.3 Sequence Diagram 13
7.3.1 Login for admin and user 13
7.3.2 Room Booking 13
7.4 ER Diagram 14

8.Implementation 15
8.1 Login Page 15
8.2 Dashboard 15
8.2.1 Admin 15
8.2.2 user 15

9.Testing 17
9.1 Performance Testing 17
9.2 Test Case documentation 19
9.2.1 Admin Dashboard 19
9.2.2 Admin Manage Course 19
9.2.3 View logs 20
9.2.4 Admin Add course 20
9.2.5 Admin manage Room 21
9.2.6 Admin Student Registration 21
9.2.7 Admin Add Rooms 22
9.2.8 View Booked Room 22
9.2.9 Update User Profile 23
9.2.10 Change Password 23
9.2.11 Admin Login 24
9.2.12 User Login 24
9.2.13 Book Room 25
9.2.14 Forgot Password 25

10.Tools Used 25
10.1 Version Control Tool-GIt Hub 26
10.2 process Tree- 26
10.3 Design Tool-Draw.io 26
10.4 Bug tracking- Pivotal Tracker 27
10.5 Performance Testing-Pingdom Tools 27
10.6 Automation Tool- Selenium 27

10.7 Webhook development tool and debugging- ngrok 28

1.Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Web Publishing System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

1.2 Document Conventions

This document uses the following conventions.

- DB Database
- DDB Distributed Database
- ER Entity Relationship
- JS javascript
- SRS – Software Requirements Specifications
- HMS Hostel Management System

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the hostel management system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the hostel management and as well as to the students

1.4 Product Scope

The software product “Hostel Management System” will be an application that will be used for maintaining records in an organised manner and to replace old paper work system. This project aims at automating the hostel management for smooth working of the hostel by automating almost all the activities. Updations and modifications will be easily achievable and all the calculations and accounting work would be more accurate.

1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2.Overall Description

2.1Product Perspective

2.1.1.System Interface

The HMS is a complete web enabled system which can be accessed through web browser.

2.1.2.User Interface

The user interface is as follows:
Screen Name Description
Login Login into system as student or admin
Student Module:
Profile Student can view and update personal details. Apply Room Student applies room by selecting the preferred room, semester. Status of Application Student can check their application status.View history Students can view history of past applications Change Password Student can change his/her password. Logout After the student is done using the system, he/she logs out.
Administrator Module:
Add user Admin can add a new user by assigning new user name and passwords respectively. View Applicants Admin can view the applicants by selecting the year and semester. View Students Admin can view the student'sdetails Change Password Admin can change his/her password.

2.2 Product Functions

2.2.1.User Functions

The administrator of HMS shall add new users to the system who is basically the student. After entering the information about the user, the system gives a unique username and password to the user.The administrator shall view applicants and students, and change password.

2.2.2.Student Functions

The student shall view and update their profiles.
The student shall apply a room.
The student shall view the status of application.
The student shall view their history and change their password.

2.3 User Classes and Characteristics

The Administrator
This user has to have at least Window 7/Linux OS and Internet browsing skills for administrating HMS user profiles.
The Student
This user has to have at least Window 7/Linux OS and Internet browsing skills to use the system.

2.4 Operating Environment

- Server Side

Apache Web server is installed and will enable HMS to interact with its users. PHP is a server-side scripting language, which will be used to code the HMS.

- Client Side

On the client side the required software product is Internet Explorer/Google Chrome/Mozilla Firefox supporting at least HTML version 3.2, java enabled, and any operating system that can run the browsers.

- Communication Interface

The default communication protocol for data transmission between server and the client is Transmission Control Protocol/ Internet Protocol (TCP/IP). At the upper level Hyper Text Transfer Protocol (HTTP) will be used for communication between the web server and client.

- The Administrator

This user has to have at least Window 7/Linux OS and Internet browsing skills for administrating HMS user profiles.

- The Student

This user has to have at least Window 7/Linux OS and Internet browsing skills to use the system.

2.5 Design and Implementation Constraints

Adhere to the institutional rules and regulation. It shoud'nt violate the DUGC set norms and principles

2.6 Assumptions and Dependencies

The system will not store any payment information; rather all payments will be handled by the finance department of the university.

Credit card payment or any other form of payment other than through the finance department is not allowed on the system

3.External Interface Requirements

3.1 User Interfaces

The user interface is as follows:

Login Login into system as student or admin

Student Module:

- Profile Student can view and update personal details.
- Apply Room Student applies room by selecting the preferred room, semester.
- Status of Application Student can check their application status.
- View history Students can view history of past applications
- Change Password Student can change his/her password.
- Logout After the student is done using the system, he/she logs out.

Administrator Module:

- Add user Admin can add a new user by assigning new user name and passwords respectively.
- View Applicants Admin can view the applicants by selecting the year and semester.
- View Students Admin can view the Student's Details
- Change Password Admin can change his/her password.
- Add user Admin can add a new user by assigning new user name and passwords respectively.
- View Applicants Admin can view the applicants by selecting the year and semester.

- View Students Admin can view the student’s Details.
- Change Password Admin can change his/her password.
- Add user Admin can add a new user by assigning new user name and passwords respectively.
- View Applicants Admin can view the applicants by selecting the year and semester.
- View Students Admin can view the student’s Details.
- Change Password Admin can change his/her password.

3.2 Hardware Interfaces

Client Side

Any Personal computer, which can support any 7-window or Windows environment with a mouse support, is acceptable.

Server Side

HMS will be run on a web server, which is installed into the school server.The school servers have requirements to operate PHP scripts (Apache Web server 1.3.2 with PHP 4.0
modules).

3.3 Software Interfaces

Server Side

Apache Web server is installed and will enable HMS to interact with its users. PHP is a server-side scripting language, which will be used to code the HMS.

Client Side

On the client side the required software product is Internet Explorer/Google Chrome/Mozilla Firefox supporting at least HTML version 3.2, java enabled, and any operating system that can run the browsers.

3.4 Communications Interfaces

The default communication protocol for data transmission between server and the client is Transmission Control Protocol/ Internet Protocol (TCP/IP). At the upper level Hyper Text Transfer Protocol (HTTP) will be used for communication between the web server and client.

4.System Features

4.1 Description and Priority

The HMS uses the standard input/output devices for a personal computer. This includes the following:
Keyboard, Mouse, Monitor and Printer.

4.2 Stimulus/Response Sequences

The HMS operations needed by the users are described below.
Administrator of the system creates and defines the status of users by (Add User). The user will be given a unique username and password. The Admin may change their passwords by (Change Password). The Admin can view applicants and also view the student’s details
The student accesses the system by logging in. They can view their profiles and update it (Profile), Apply room, View Status of Application, View history and change their passwords.

4.3 Functional Requirements

4.3.1 Administrator Functions

4.3.1.1 Add User

Introduction: HMS shall enable administrator to add new users to the system.
Input: username and password.
Process: The administrator activates the function and enters the username and password of the new user. The function will also check the database whether the user already exists or not. According to the results, the system adds the user to the all user list with a confirmation message, or the function displays an error message.
Output: error message or confirmation message.

4.3.1.2 View Applicants

Introduction: HMS shall display all the applicants archived in the system.
Input: none
Process: The administrator selects the semester and year. The function queries the database for the students who have applied for rooms.
Output: All applicants with their respective details (user id, preferred room, and assigned room id) will be displayed.

4.3.1.3 View Students

Introduction: HMS shall display all the students in the system.
Input: none
Process: When the administrator logon the system, automatically, all student list is displayed. The function queries the database for all the students.
Output: List of all students with their respective details (student id, first name, and last name, and gender, place of residence, phone number, and address) will be displayed.

4.3.1.4 Change Password

Introduction: HMS shall enable administrator to change the password.
Input: old password, new password, confirm password
Process: Administrator activates the function to change the password. The new password and confirm password fields are entered. If they match, the old password will be updated with the new one.
Output: Error or confirmation message will be displayed.

4.3.2 Student Functions

4.3.2.1 Profile

Introduction: HMS shall enable student to view and update their profile.
Input: none
Process: By this function, the database is queried for all the personal information of the student.
Output: All students’ personal information is displayed.

4.3.2.2. Apply Room

Introduction: HMS shall enable a student to apply a room.
Input: preferred room, semester, and year.

Process: By this function, the selected information is stored into the database.
Output: All students' application information is stored into the database

4.3.2.3. Status of Application

Introduction: HMS shall enable the student to view the status of their room application.
Input: student id
Process: By this function, the database is queried for all the room application information of the student.
Output: All application status is displayed.

4.3.2.4. View History

Introduction: HMS shall enable the student to view their previous application history.
Input: student id
Process: By this function, the database is queried for all the previous room application information of thestudent.
Output: All room application history is displayed.

4.3.2.5. Change Password

Introduction: HMS shall enable student to change the password.
Input: old password, new password, confirm password
Process: student activates the function to change the password. The new password and confirm password
fields are entered. If they match, the old password will be updated with the new one.
Output: Error or confirmation message will be displayed.

5.Other Nonfunctional Requirements

5.1Performance Requirements

Performance requirements define acceptable response times for system functionality. Depending on the user internet connection speed;
The load time for user interface screens should take no longer than two seconds.
The log in information shall be verified within five seconds.
Queries shall return results within five seconds.

5.2Safety Requirements

Access to the system is protected by username and password by using a user login screen. Maintaining backups ensures the database security. System restores in case of emergencies.

5.3Security Requirements

Nobody should be allowed to tamper with data; Enhanced Security for sensitive data. It should be made sure that only users who are given specific rights can access data and all actions are logged, thus providing an extensive role based authorization.

5.4Software Quality Attributes

Availability:The system shall be available for 24 hours.
Correctness : extent to which program satisfies specifications , fulfills user's mission and objectives

5.5Business Rules

Hostel management system is design so that our universities and colleges can easily manage the data of students and related things.

For the best understanding first we have to define the project scope or the scenario because different problem can be solve different design and more than one scenarios can be created for each problem. People design them according to their thinking.

We are also creating some type scenario so that our design can be bit specific for some kind of situation.Our project is defined as;As we can see that our university has the facility of hostel for boys and girls.

We will focus on the boy's hostel only as almost all the things will be same in both hostel we will manage only boys so that it will be simple and easy to understand for everyone.

Obviously many students will be living in the boy's hostel.

Boy's hostel has many rooms for the accommodation of the students in which more than one student accommodate their self.

6.Usage Scenario

This web application product the hostel management to improve their services for all the students of the hostel. This also reduce the manual work of the persons in admin panel and the bundle of registers that were search when to find the information of a previous student, because through this system you can store the data of those students who had left the hostel. Through this you can check the personal profile of all the current students within few minutes the data base of the system will help you to check a particular one. The system will help you to check the student's hostel receipt. The students of the hostel will be recognized from the ID number allocated at the room rental time. In the last this system will improve the management work in the hostel. This project is mainly focus on the solution regarding the hostel management online process to accommodate the issues that are done in manual existing offline systems.

6.1.Use case Diagram

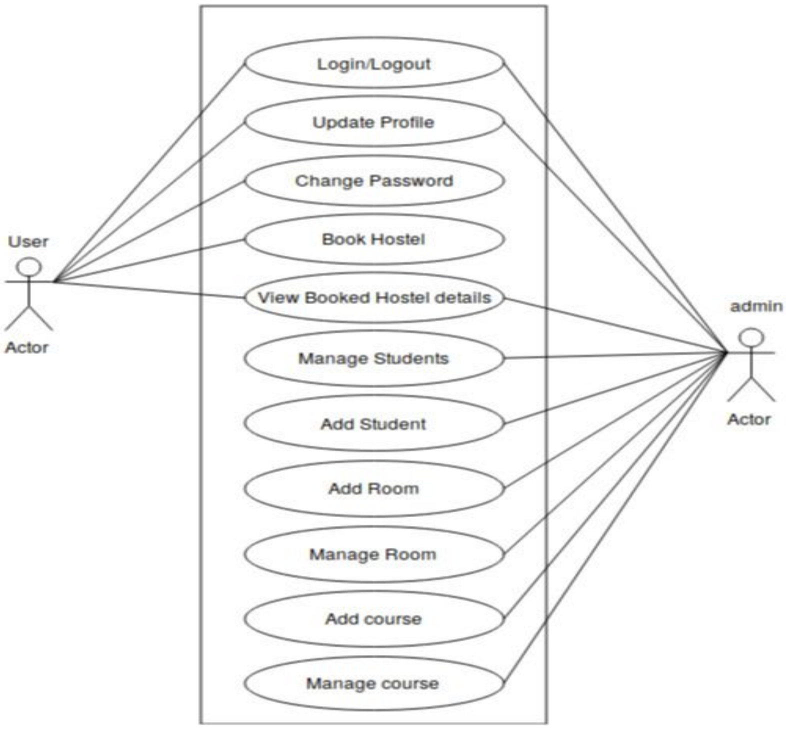


Fig 6.1 Use Case Diagram

7.System Design

The application comprises of many features and hence the system is divided into various components. The main objective of this section is to elaborate the system design and to give an overview of the various components of the application including their interfaces. It also provides information about the relationship between the various components and the different data elements used by each of the components. It also explains the overall system design. The application has a client-server architecture with the application running on the client side and the files residing on the server side with which the user interacts through the application. The following sections contains class diagram, sequence diagram and activity diagrams representing the various components and their interactions and also the detailed description of each of the components.

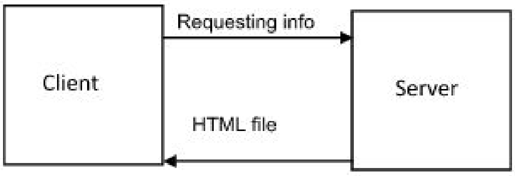


Fig7.1 Client server architecture

7.1.Activity Diagram

7.1.1 For admin

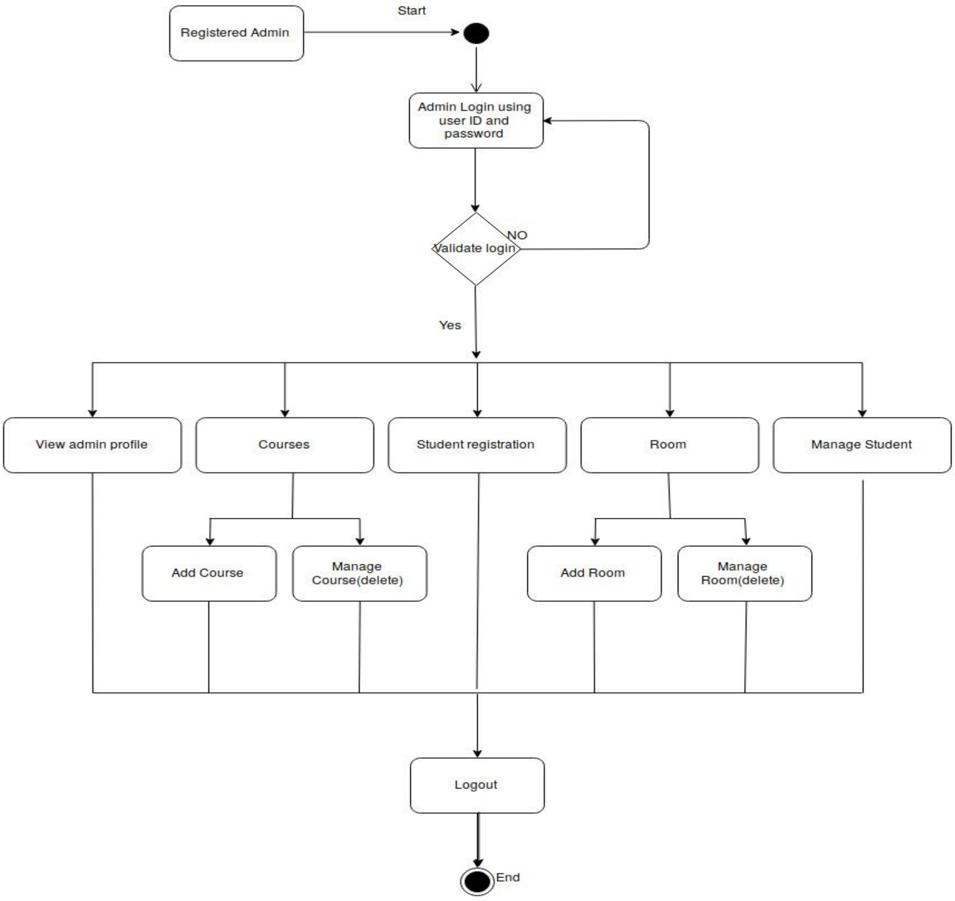


Fig 7.1.1 Activity Diagram for Admin

7.1.2 For User

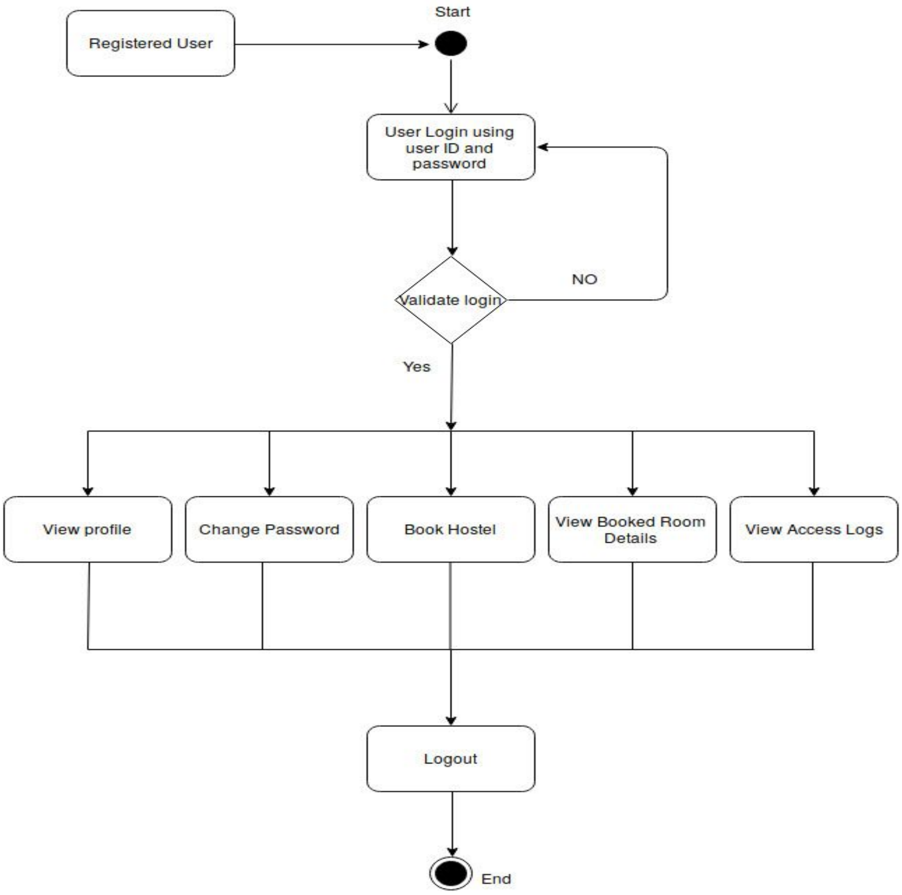


Fig 7.1.2 Activity Diagram for User

7.2 Class diagram

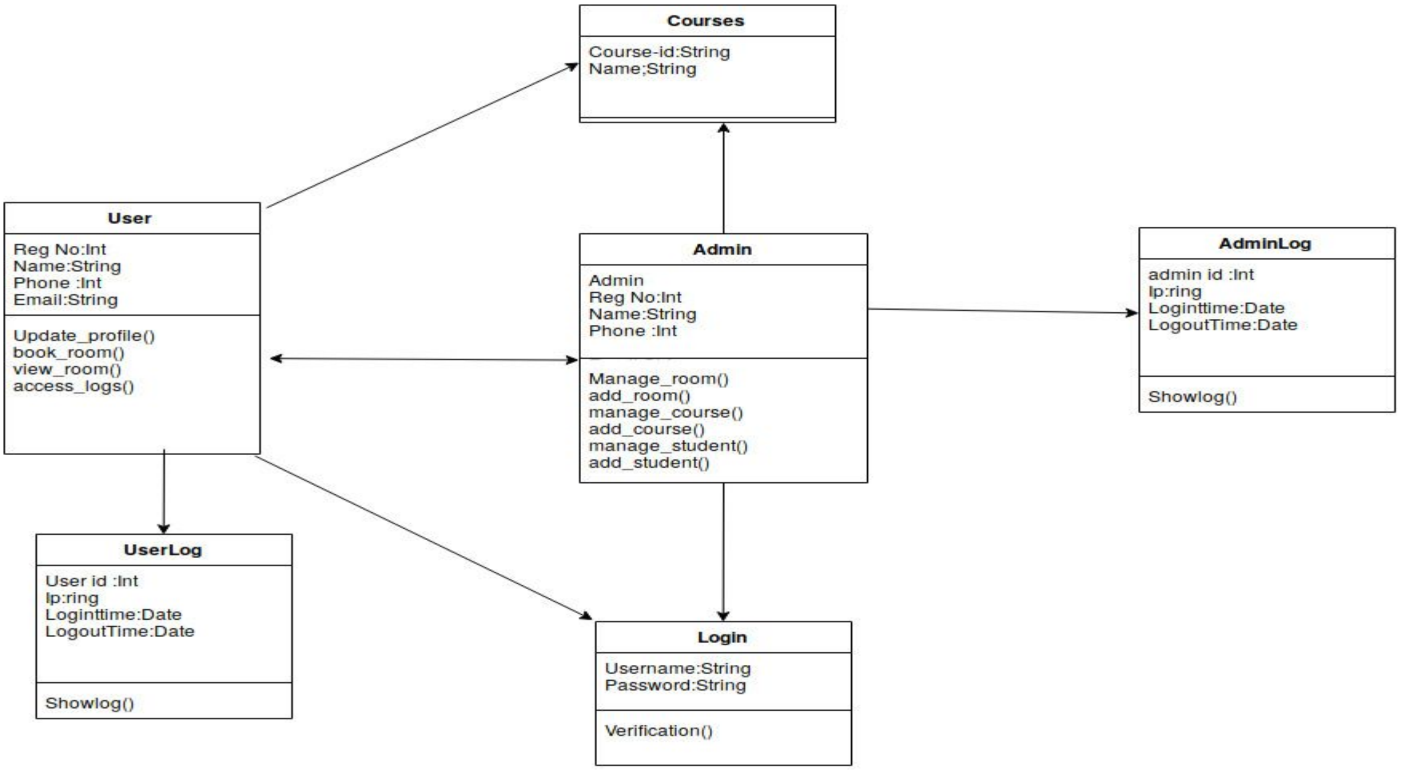


Fig 7.2.1 Class Diagram

7.3 Sequence Diagram

7.3.1 Login for admin and user

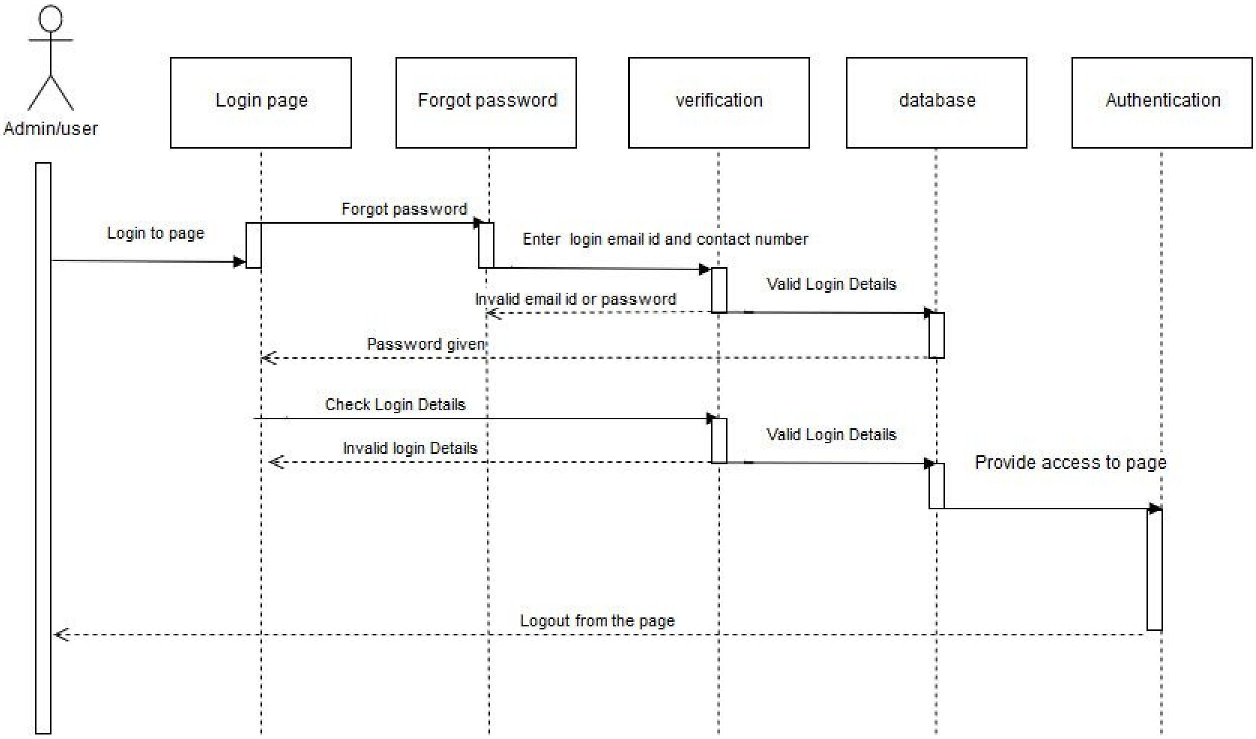


Fig 7.3.1 Sequence Diagram login for user and admin

7.3.2 Room Booking

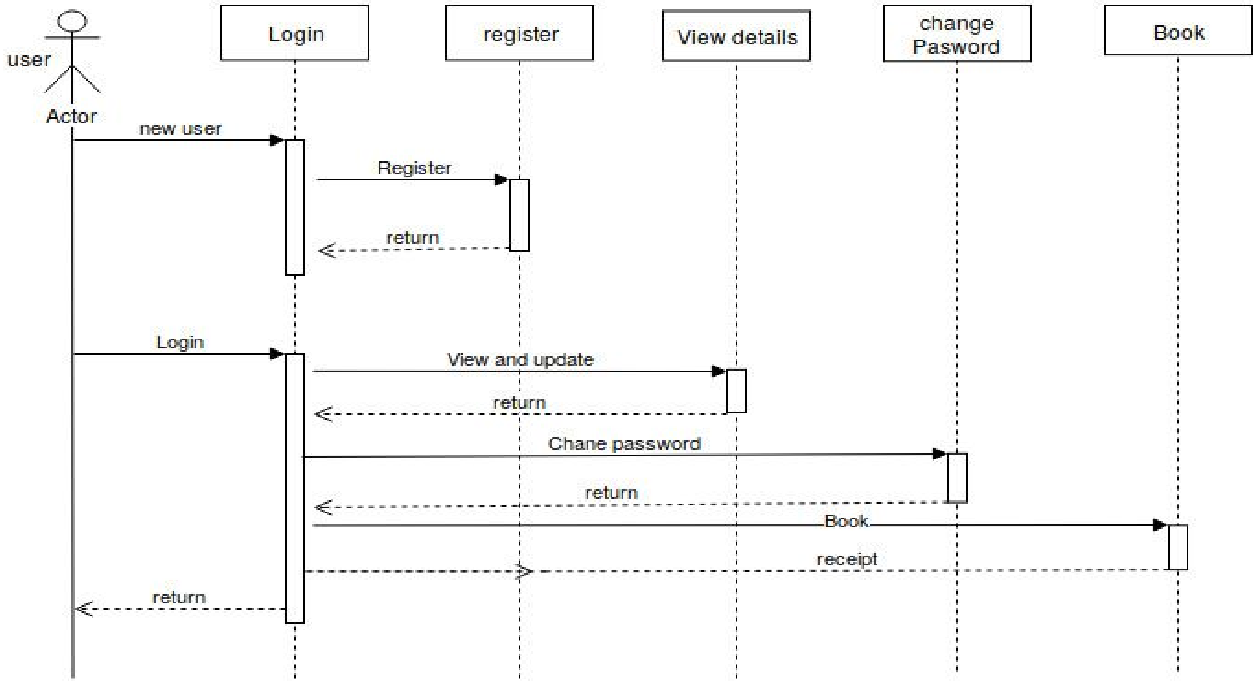


Fig 7.3.2 Sequence Diagram Room Booking

7.4 ER Diagram

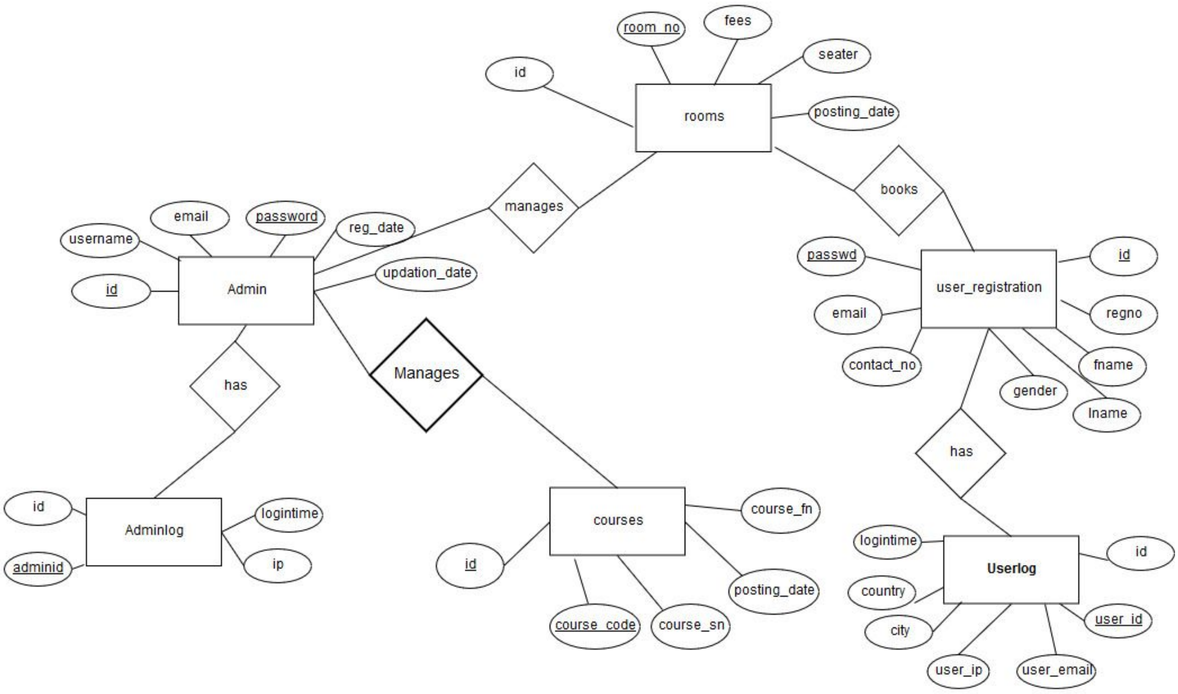
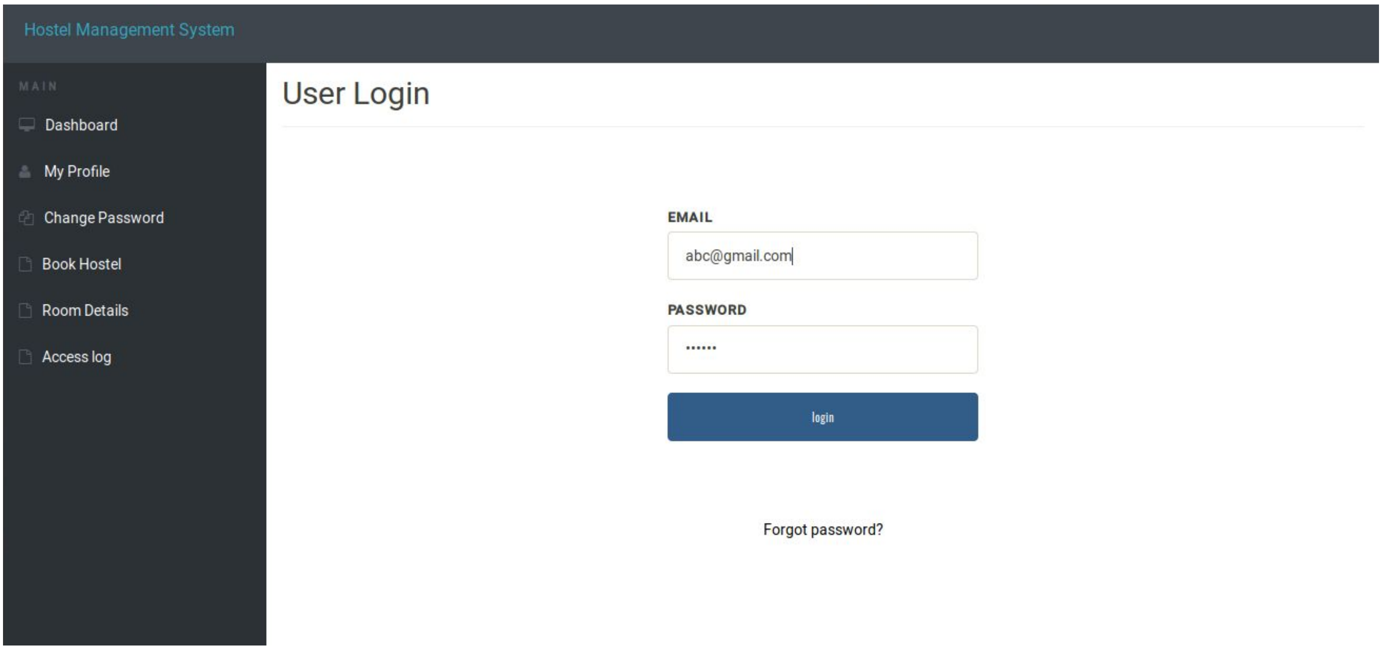


Fig 7.4.1 ER Diagram

8.Implementation

8.1 Login Page



User Login

EMAIL

abc@gmail.com

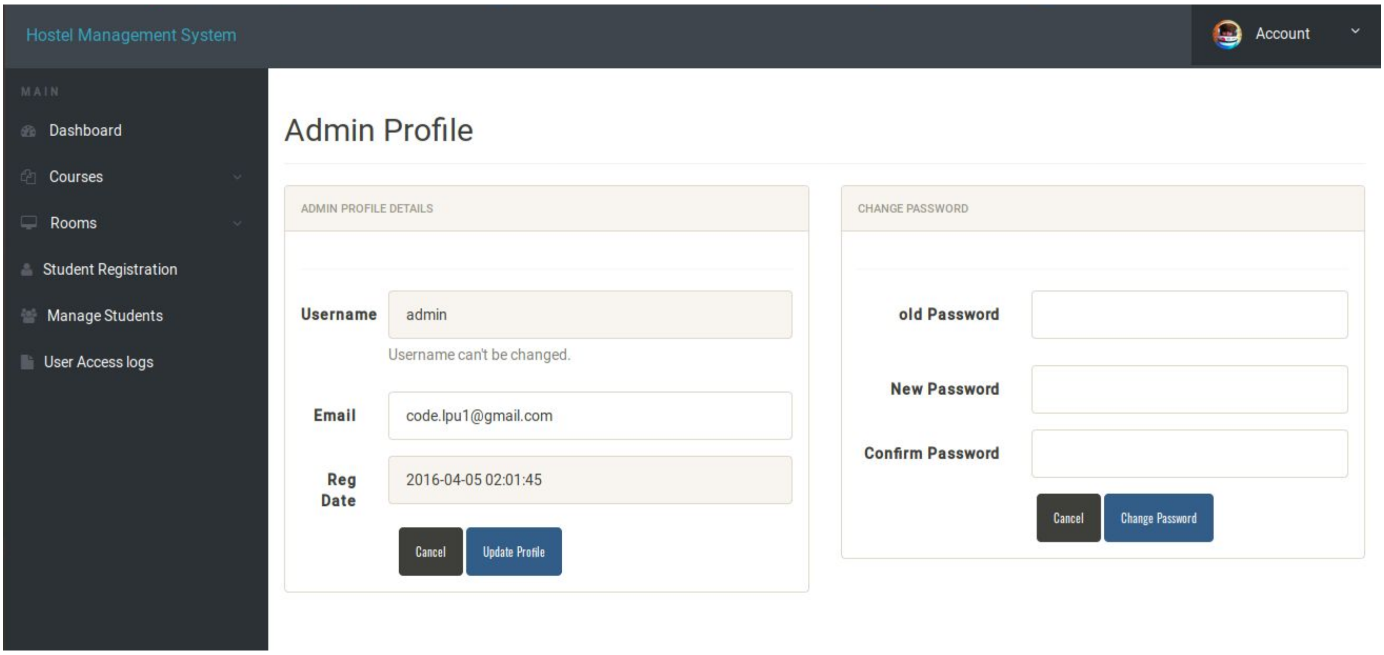
PASSWORD

••••••

Fig 8.1 ScreenShot of login

8.2 Dashboard

8.2.1 Admin



Admin Profile

ADMIN PROFILE DETAILS

Username: admin

Username can't be changed.

Email: code.lpu1@gmail.com

Reg Date: 2016-04-05 02:01:45

CHANGE PASSWORD

old Password

New Password

Confirm Password

Fig 8.2.1ScreenShots of Dashboard of admin

8.2.2 user

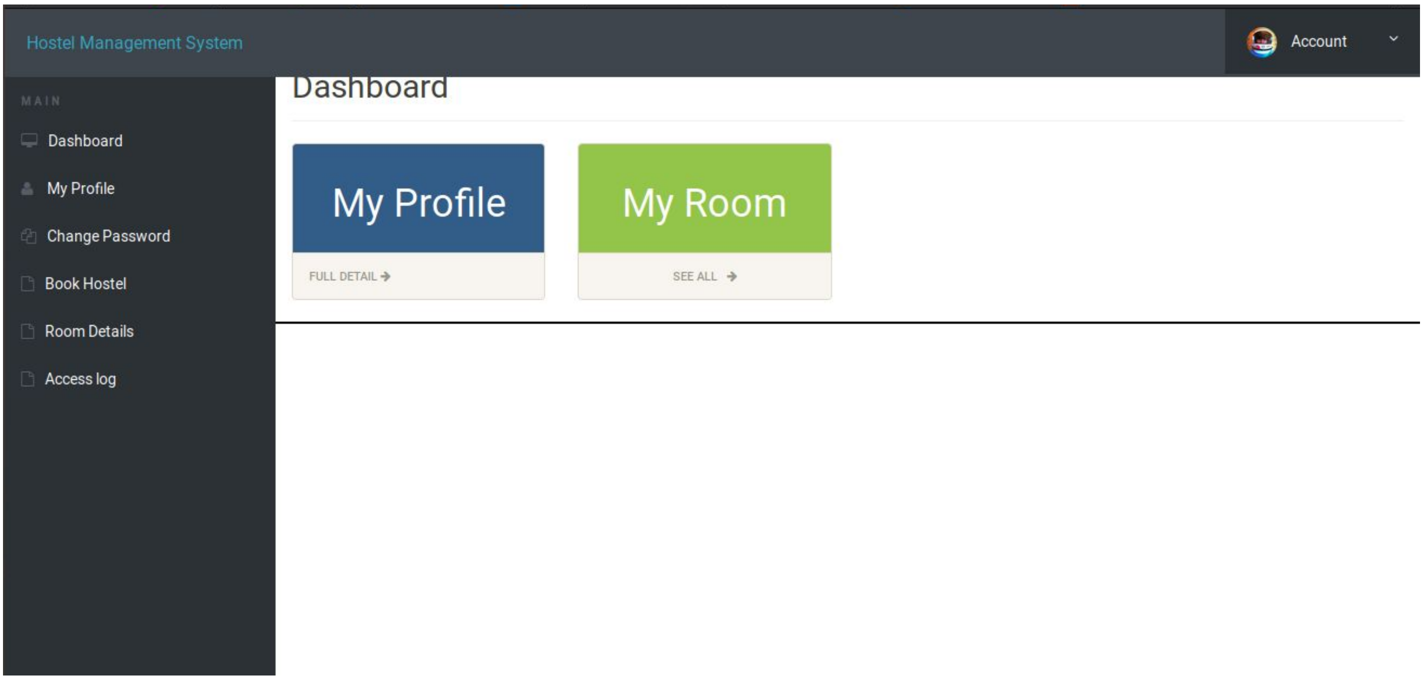
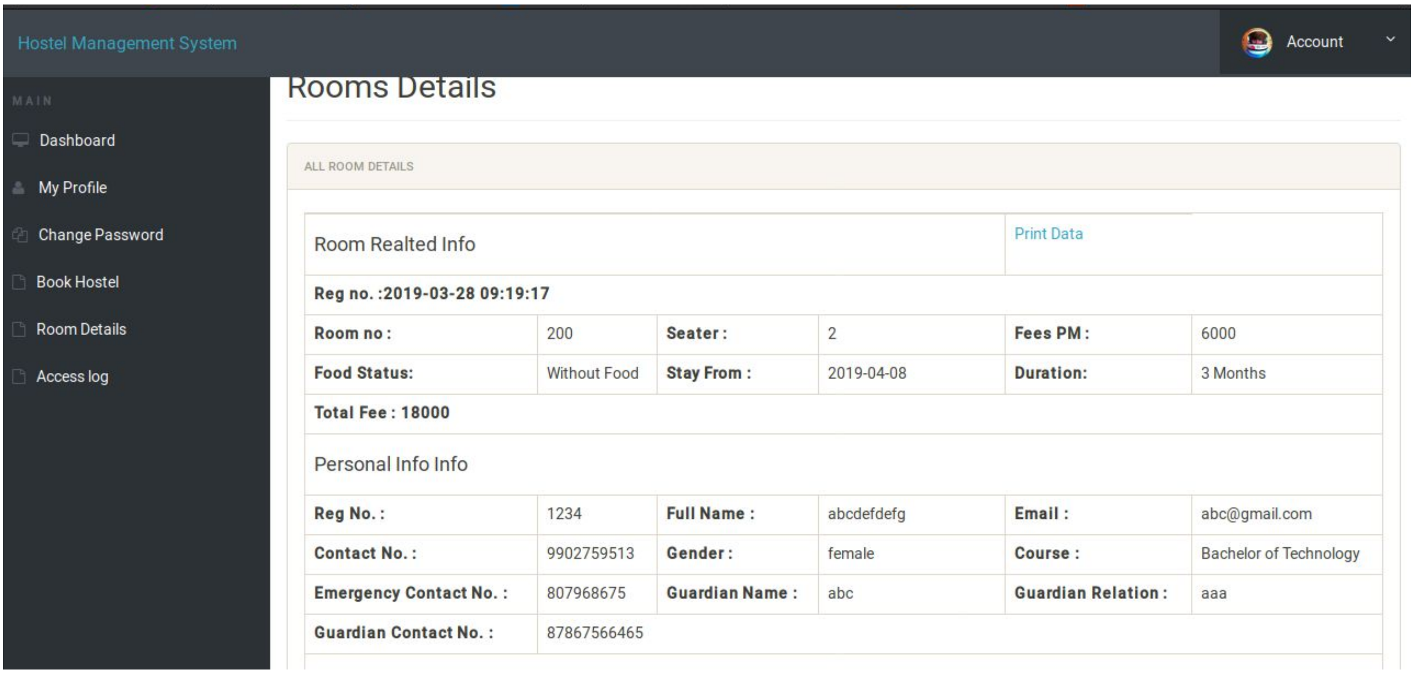


Fig 8.2.2 ScreenShots of Dashboard of admin

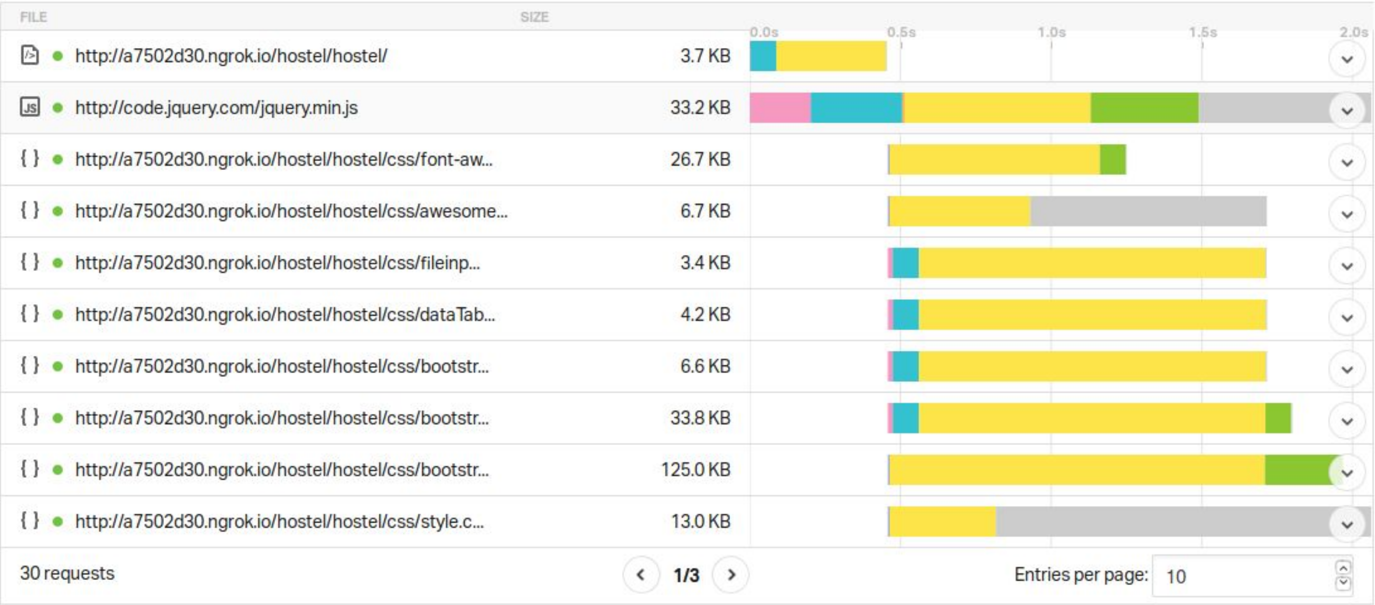


Room Realted Info				Print Data	
Reg no. :2019-03-28 09:19:17					
Room no :	200	**Seater :**	2	**Fees PM :**	6000
Food Status:	Without Food	**Stay From :**	2019-04-08	**Duration:**	3 Months
Total Fee : 18000					
Personal Info Info					
Reg No. :	1234	**Full Name :**	abcdefdefg	**Email :**	abc@gmail.com
Contact No. :	9902759513	**Gender :**	female	**Course :**	Bachelor of Technology
Emergency Contact No. :	807968675	**Guardian Name :**	abc	**Guardian Relation :**	aaa
Guardian Contact No. :	87867566465				

Fig 8.2.3 ScreenShots of Booked room detail

9.Testing

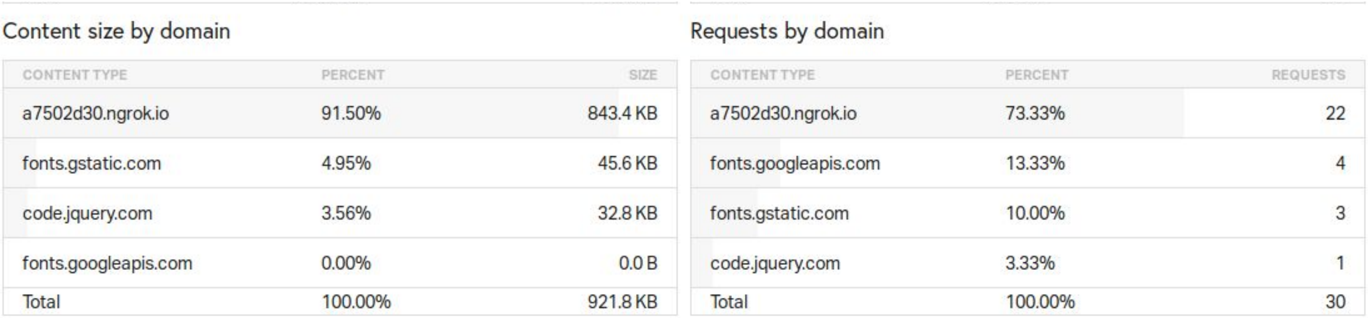
9.1 Performance Testing



FILE	SIZE
● http://a7502d30.ngrok.io/hostel/hostel/	3.7 KB
JS ● http://code.jquery.com/jquery.min.js	33.2 KB
{ } ● http://a7502d30.ngrok.io/hostel/hostel/css/font-aw...	26.7 KB
{ } ● http://a7502d30.ngrok.io/hostel/hostel/css/awesome...	6.7 KB
{ } ● http://a7502d30.ngrok.io/hostel/hostel/css/fileinp...	3.4 KB
{ } ● http://a7502d30.ngrok.io/hostel/hostel/css/dataTab...	4.2 KB
{ } ● http://a7502d30.ngrok.io/hostel/hostel/css/bootstr...	6.6 KB
{ } ● http://a7502d30.ngrok.io/hostel/hostel/css/bootstr...	33.8 KB
{ } ● http://a7502d30.ngrok.io/hostel/hostel/css/bootstr...	125.0 KB
{ } ● http://a7502d30.ngrok.io/hostel/hostel/css/style.c...	13.0 KB

30 requests 1/3 Entries per page: 10

Table 9.1.1 Request



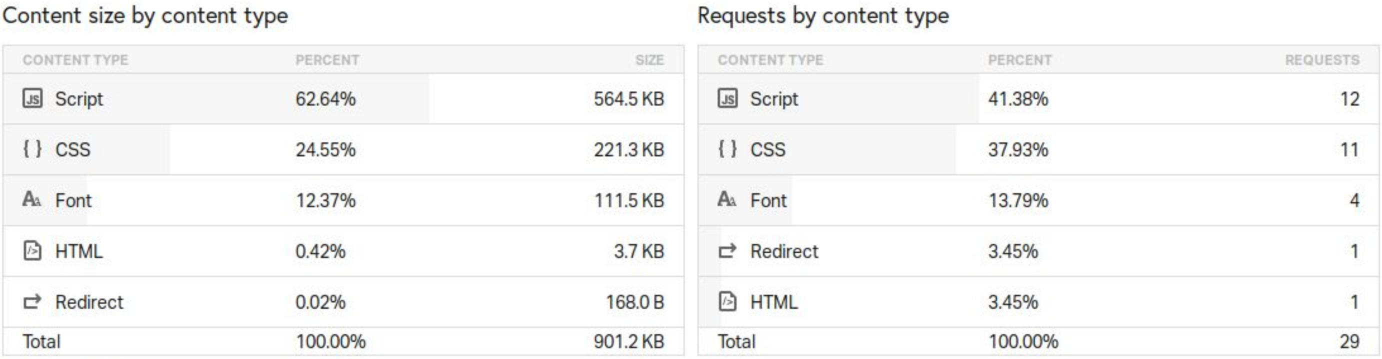
Content size by domain

CONTENT TYPE	PERCENT	SIZE
a7502d30.ngrok.io	91.50%	843.4 KB
fonts.gstatic.com	4.95%	45.6 KB
code.jquery.com	3.56%	32.8 KB
fonts.googleapis.com	0.00%	0.0 B
Total	100.00%	921.8 KB

Requests by domain

CONTENT TYPE	PERCENT	REQUESTS
a7502d30.ngrok.io	73.33%	22
fonts.googleapis.com	13.33%	4
fonts.gstatic.com	10.00%	3
code.jquery.com	3.33%	1
Total	100.00%	30

Table 9.1.2 Content and request by Domain



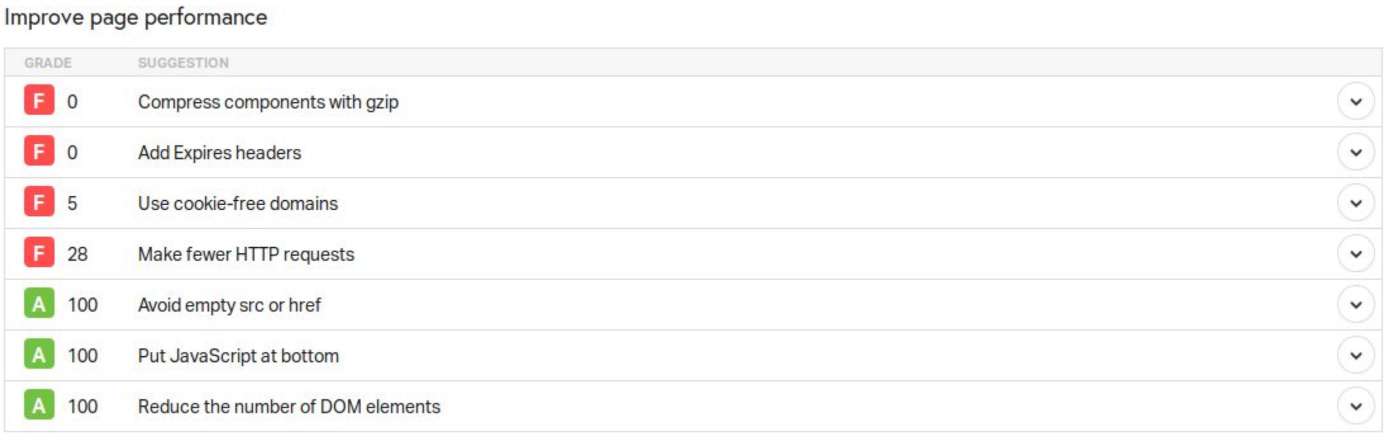
Content size by content type

CONTENT TYPE	PERCENT	SIZE
JS Script	62.64%	564.5 KB
{ } CSS	24.55%	221.3 KB
Aa Font	12.37%	111.5 KB
HTML	0.42%	3.7 KB
↪ Redirect	0.02%	168.0 B
Total	100.00%	901.2 KB

Requests by content type

CONTENT TYPE	PERCENT	REQUESTS
JS Script	41.38%	12
{ } CSS	37.93%	11
Aa Font	13.79%	4
↪ Redirect	3.45%	1
HTML	3.45%	1
Total	100.00%	29

Table 9.1.3 Content and request by Type

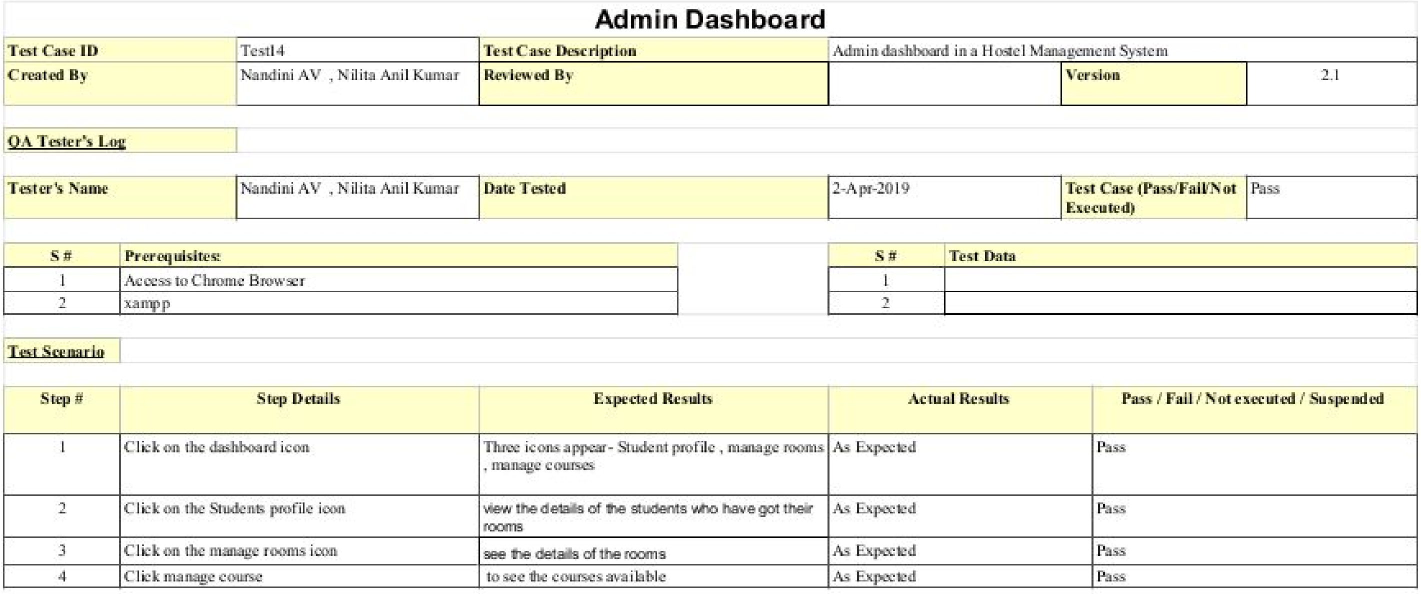


Improve page performance

GRADE		SUGGESTION
F	0	Compress components with gzip
F	0	Add Expires headers
F	5	Use cookie-free domains
F	28	Make fewer HTTP requests
A	100	Avoid empty src or href
A	100	Put JavaScript at bottom
A	100	Reduce the number of DOM elements

Table 9.1.4 performance Report

9.2 Test Case documentation



Admin Dashboard

Test Case ID	Test14	**Test Case Description**	Admin dashboard in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	**Date Tested**	2-Apr-2019	**Test Case (Pass/Fail/Not Executed)**	Pass

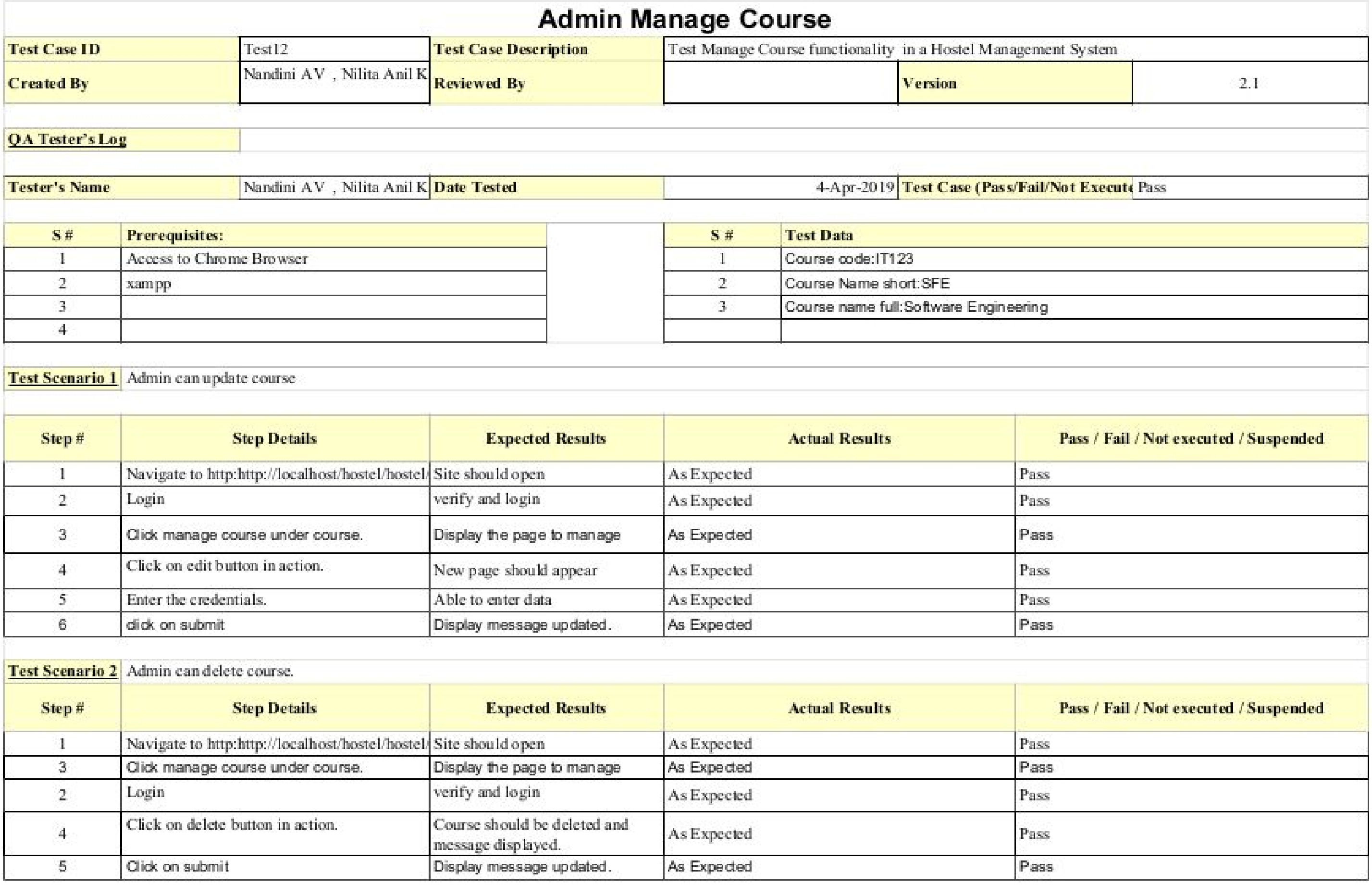
S #	Prerequisites:
1	Access to Chrome Browser
2	xampp

S #	Test Data
1	
2	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Click on the dashboard icon	Three icons appear- Student profile , manage rooms , manage courses	As Expected	Pass
2	Click on the Students profile icon	view the details of the students who have got their rooms	As Expected	Pass
3	Click on the manage rooms icon	see the details of the rooms	As Expected	Pass
4	Click manage course	to see the courses available	As Expected	Pass

9.2.1 Admin Dashboard



Admin Manage Course

Test Case ID	Test12	**Test Case Description**	Test Manage Course functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil K	**Reviewed By**		**Version**	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil K	**Date Tested**	4-Apr-2019	**Test Case (Pass/Fail/Not Executo**	Pass

S #	Prerequisites:
1	Access to Chrome Browser
2	xampp
3	
4	

S #	Test Data
1	Course code:IT123
2	Course Name short:SFE
3	Course name full:Software Engineering

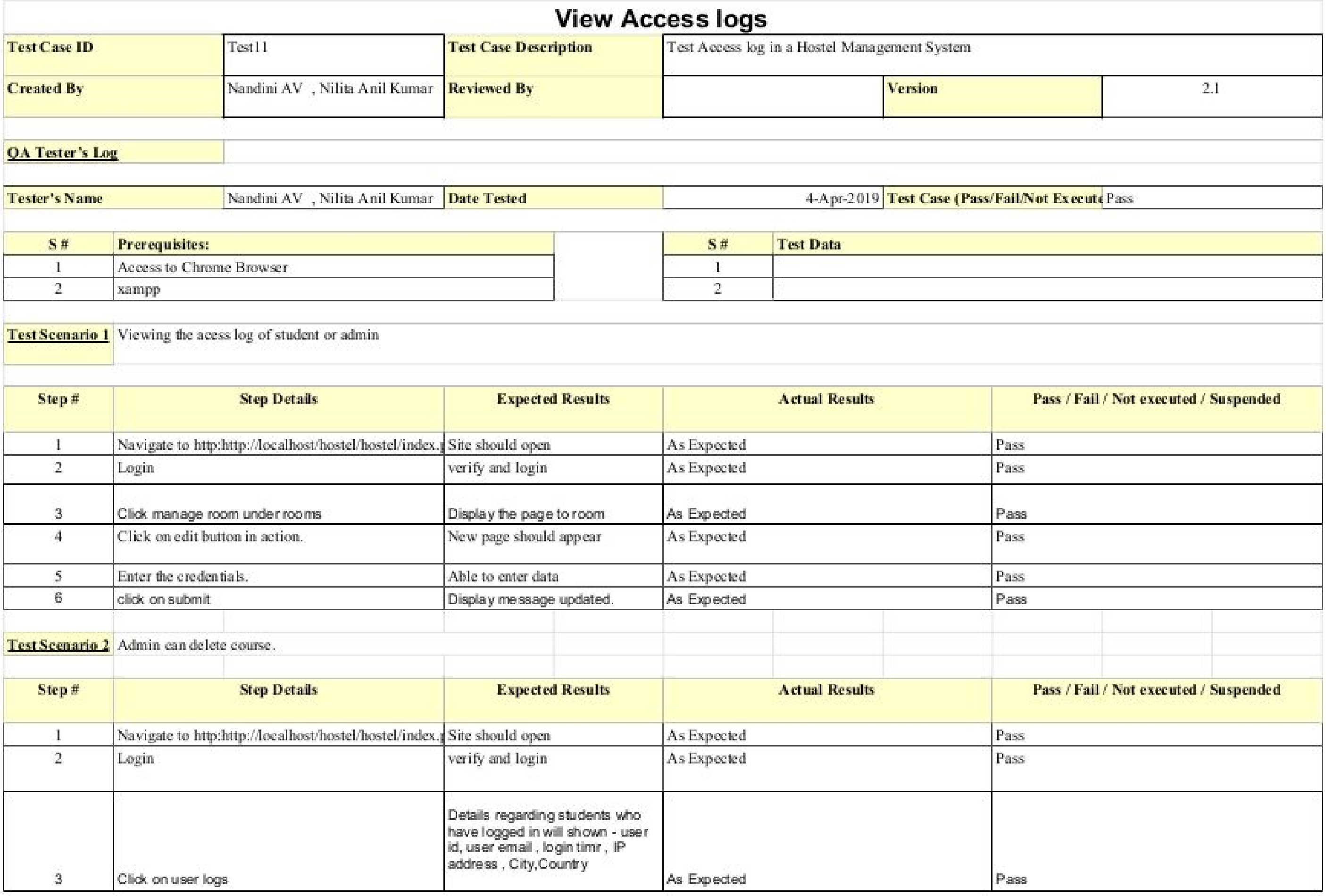
Test Scenario 1 Admin can update course

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/	Site should open	As Expected	Pass
2	Login	verify and login	As Expected	Pass
3	Click manage course under course.	Display the page to manage	As Expected	Pass
4	Click on edit button in action.	New page should appear	As Expected	Pass
5	Enter the credentials.	Able to enter data	As Expected	Pass
6	click on submit	Display message updated.	As Expected	Pass

Test Scenario 2 Admin can delete course.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/	Site should open	As Expected	Pass
3	Click manage course under course.	Display the page to manage	As Expected	Pass
2	Login	verify and login	As Expected	Pass
4	Click on delete button in action.	Course should be deleted and message displayed.	As Expected	Pass
5	Click on submit	Display message updated.	As Expected	Pass

9.2.2 Admin Manage Course



View Access logs

Test Case ID	Test11	Test Case Description	Test Access log in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	Reviewed By		Version	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	Date Tested	4-Apr-2019	Test Case (Pass/Fail/Not Execute	Pass

S #	Prerequisites:
1	Access to Chrome Browser
2	xampp

S #	Test Data
1	
2	

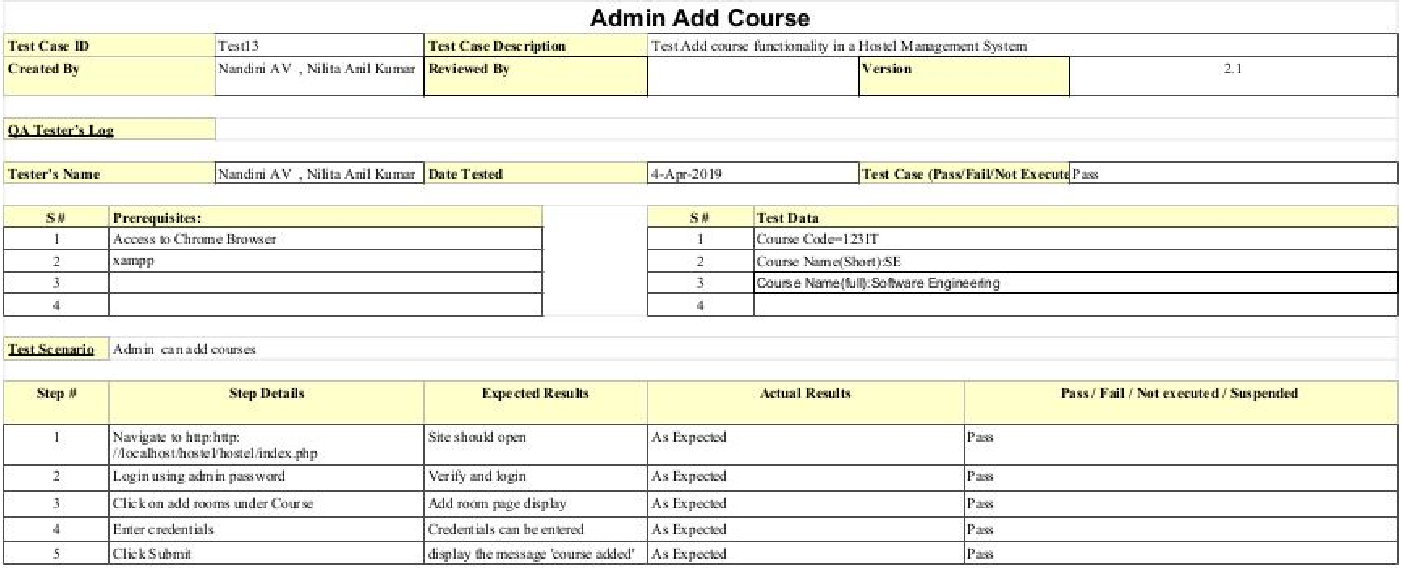
Test Scenario 1 Viewing the acess log of student or admin

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.p	Site should open	As Expected	Pass
2	Login	verify and login	As Expected	Pass
3	Click manage room under rooms	Display the page to room	As Expected	Pass
4	Click on edit button in action.	New page should appear	As Expected	Pass
5	Enter the credentials.	Able to enter data	As Expected	Pass
6	click on submit	Display message updated.	As Expected	Pass

Test Scenario 2 Admin can delete course.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.p	Site should open	As Expected	Pass
2	Login	verify and login	As Expected	Pass
3	Click on user logs	Details regarding students who have logged in will shown - user id, user email , login timr , IP address , City,Country	As Expected	Pass

9.2.3 View logs



Admin Add Course

Test Case ID	Test13	Test Case Description	Test Add course functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	Reviewed By		Version	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	Date Tested	4-Apr-2019	Test Case (Pass/Fail/Not Execute	Pass

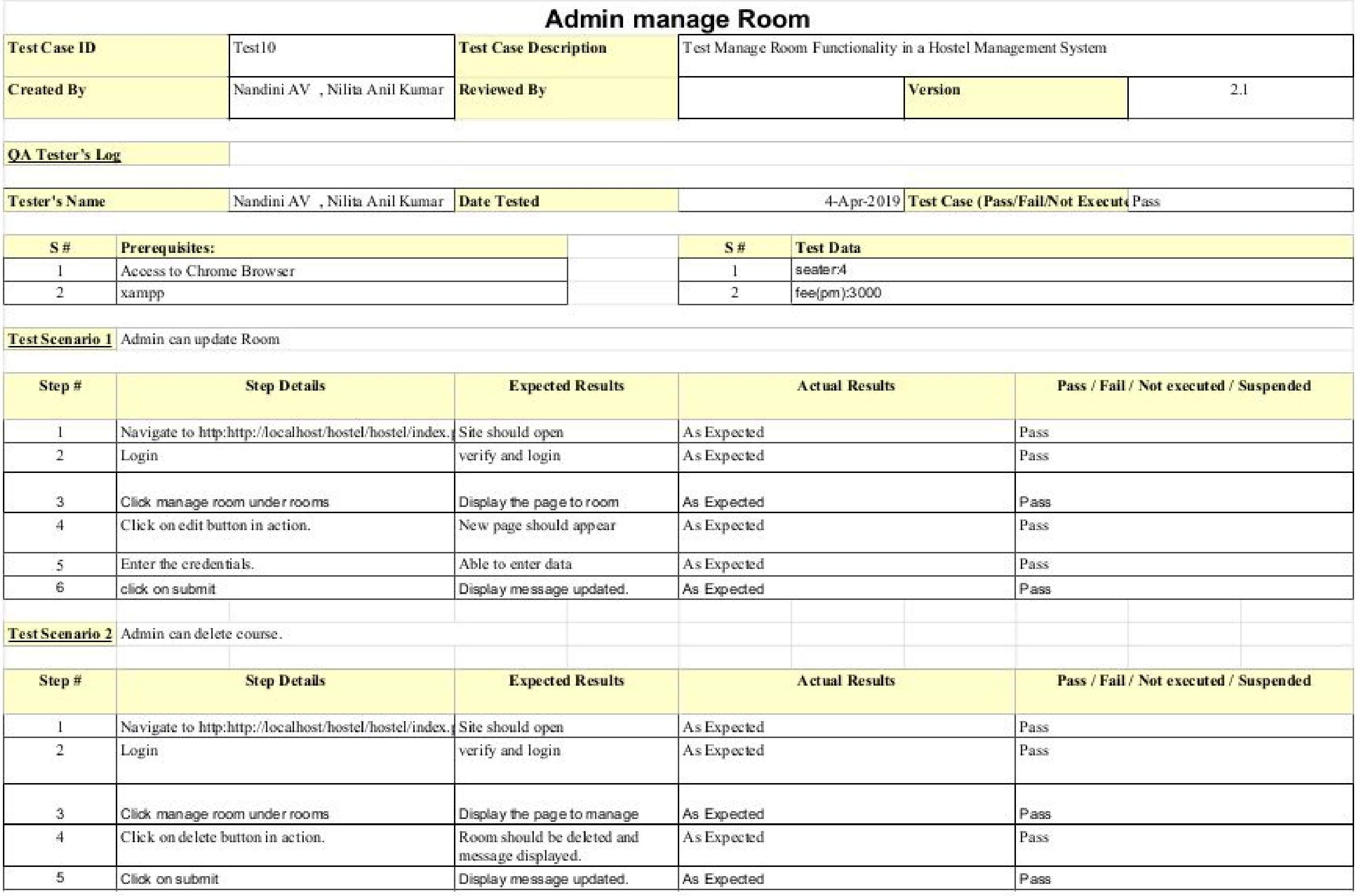
S #	Prerequisites:
1	Access to Chrome Browser
2	xampp
3	
4	

S #	Test Data
1	Course Code=123IT
2	Course Name(Short):SE
3	Course Name(full):Software Engineering
4	

Test Scenario Admin can add courses

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.php	Site should open	As Expected	Pass
2	Login using admin password	Verify and login	As Expected	Pass
3	Click on add rooms under Course	Add room page display	As Expected	Pass
4	Enter credentials	Credentials can be entered	As Expected	Pass
5	Click Submit	display the message 'course added'	As Expected	Pass

9.2.4 Admin Add course



Admin manage Room

Test Case ID	Test10	Test Case Description	Test Manage Room Functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	Reviewed By		Version	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	Date Tested	4-Apr-2019	Test Case (Pass/Fail/Not Execute	Pass

S #	Prerequisites:
1	Access to Chrome Browser
2	xampp

S #	Test Data
1	seater:4
2	fee(pm):3000

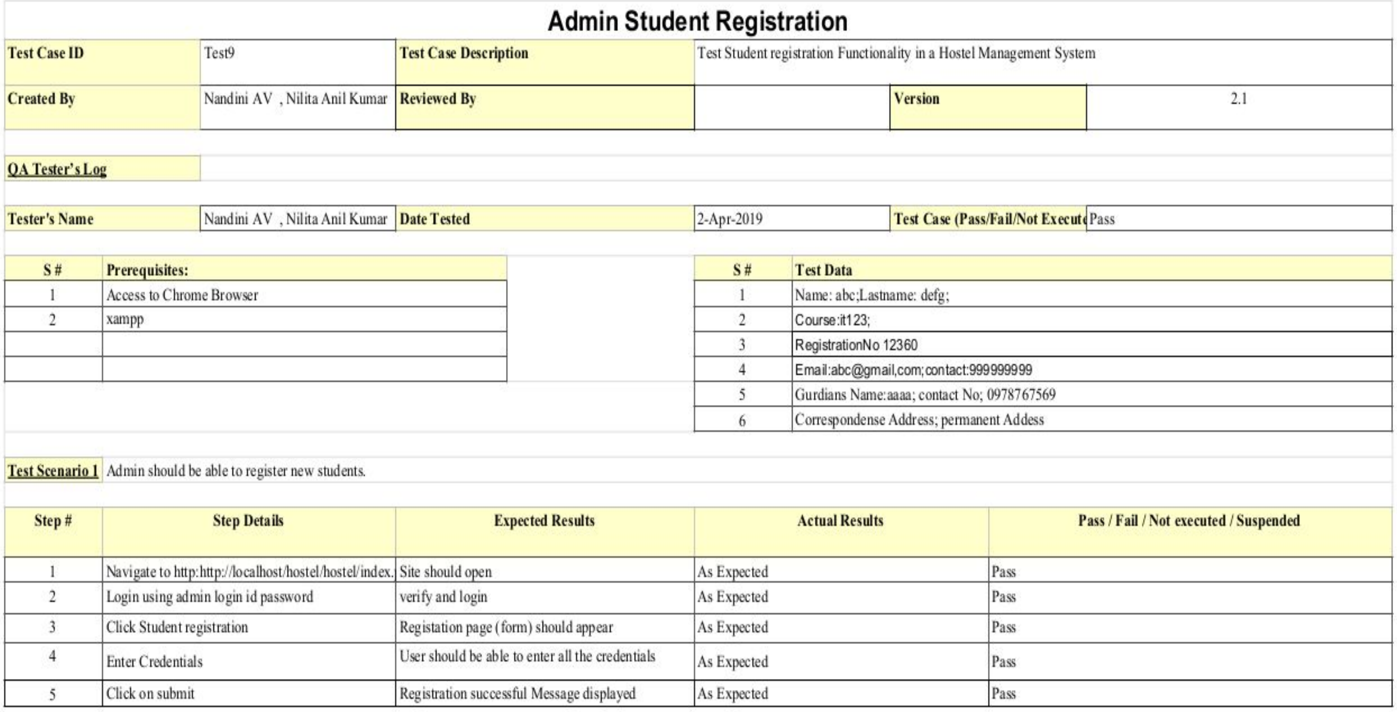
Test Scenario 1 Admin can update Room

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.j	Site should open	As Expected	Pass
2	Login	verify and login	As Expected	Pass
3	Click manage room under rooms	Display the page to room	As Expected	Pass
4	Click on edit button in action.	New page should appear	As Expected	Pass
5	Enter the credentials.	Able to enter data	As Expected	Pass
6	click on submit	Display message updated.	As Expected	Pass

Test Scenario 2 Admin can delete course.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.j	Site should open	As Expected	Pass
2	Login	verify and login	As Expected	Pass
3	Click manage room under rooms	Display the page to manage	As Expected	Pass
4	Click on delete button in action.	Room should be deleted and message displayed.	As Expected	Pass
5	Click on submit	Display message updated.	As Expected	Pass

9.2.5 Admin manage Room



Admin Student Registration

Test Case ID	Test9	Test Case Description	Test Student registration Functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	Reviewed By		Version	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	Date Tested	2-Apr-2019	Test Case (Pass/Fail/Not Execute	Pass

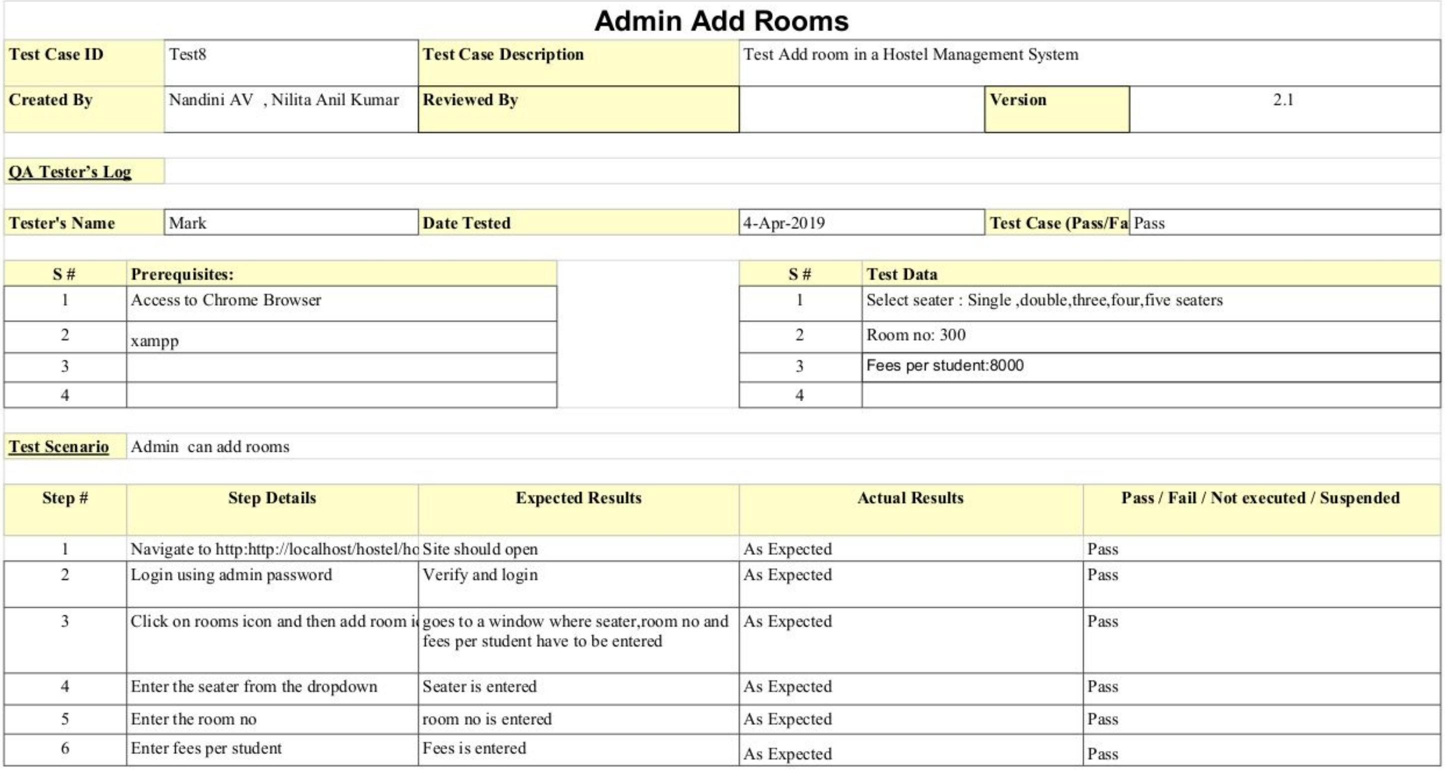
S #	Prerequisites:
1	Access to Chrome Browser
2	xampp

S #	Test Data
1	Name: abc;Lastname: defg;
2	Course:it123;
3	RegistrationNo 12360
4	Email:abc@gmail,com;contact:999999999
5	Gurdians Name:aaaa; contact No; 0978767569
6	Correspondense Address; permanent Addess

Test Scenario 1 Admin should be able to register new students.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should open	As Expected	Pass
2	Login using admin login id password	verify and login	As Expected	Pass
3	Click Student registration	Registation page (form) should appear	As Expected	Pass
4	Enter Credentials	User should be able to enter all the credentials	As Expected	Pass
5	Click on submit	Registration successful Message displayed	As Expected	Pass

9.2.6 Admin Student Registration



Admin Add Rooms

Test Case ID	Test8	**Test Case Description**	Test Add room in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

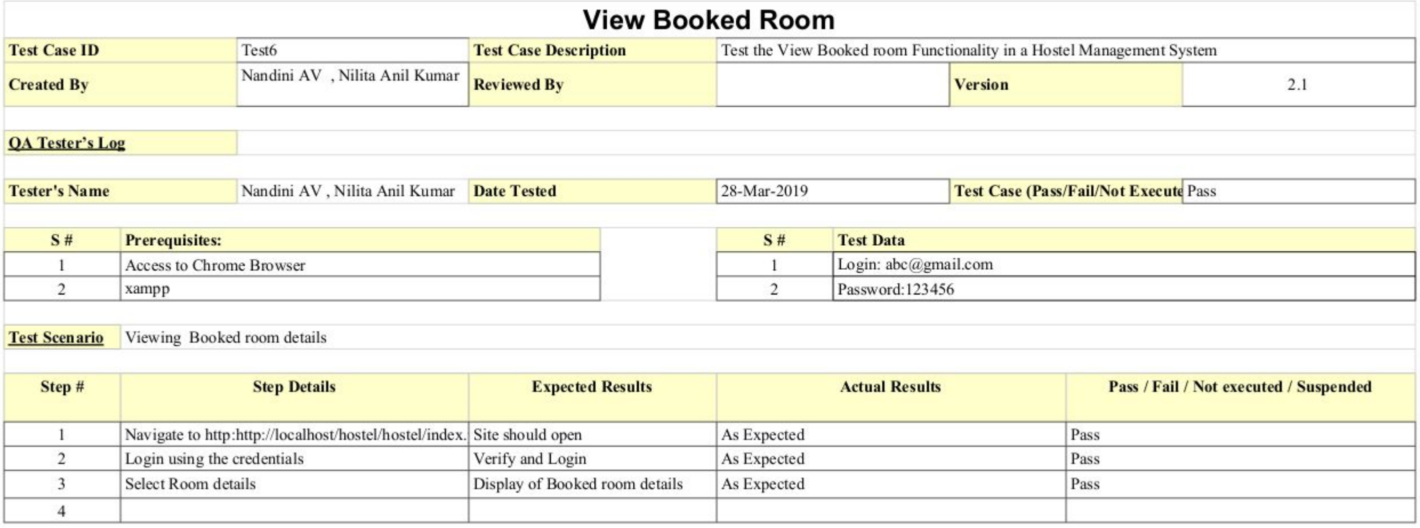
Tester's Name	Mark	**Date Tested**	4-Apr-2019	**Test Case (Pass/Fa**	Pass

S #	Prerequisites:	S #	Test Data
1	Access to Chrome Browser	1	Select seater : Single ,double,three,four,five seaters
2	xampp	2	Room no: 300
3		3	Fees per student:8000
4		4	

Test Scenario Admin can add rooms

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/ho	Site should open	As Expected	Pass
2	Login using admin password	Verify and login	As Expected	Pass
3	Click on rooms icon and then add room i	goes to a window where seater,room no and fees per student have to be entered	As Expected	Pass
4	Enter the seater from the dropdown	Seater is entered	As Expected	Pass
5	Enter the room no	room no is entered	As Expected	Pass
6	Enter fees per student	Fees is entered	As Expected	Pass

9.2.7 Admin Add Rooms



View Booked Room

Test Case ID	Test6	**Test Case Description**	Test the View Booked room Functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

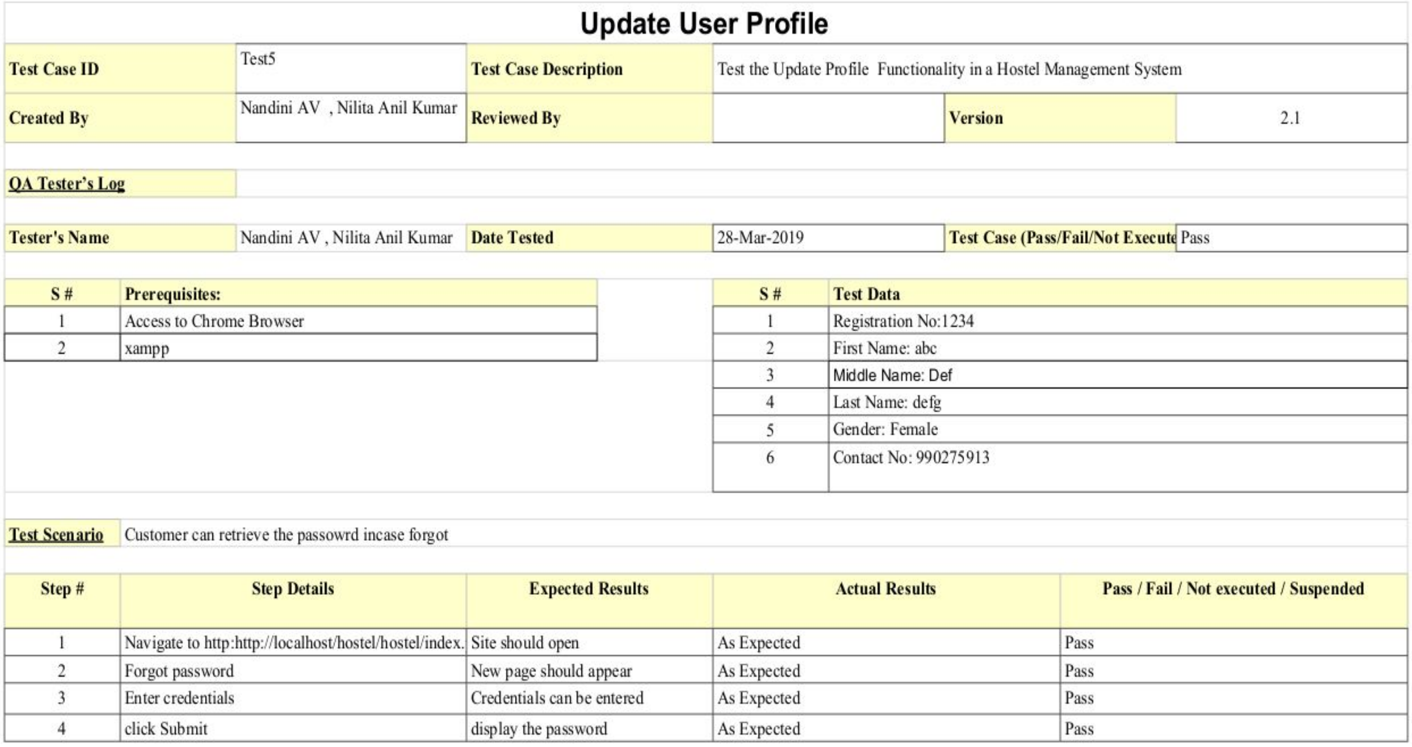
Tester's Name	Nandini AV , Nilita Anil Kumar	**Date Tested**	28-Mar-2019	**Test Case (Pass/Fail/Not Execute**	Pass

S #	Prerequisites:	S #	Test Data
1	Access to Chrome Browser	1	Login: abc@gmail.com
2	xampp	2	Password:123456

Test Scenario Viewing Booked room details

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should open	As Expected	Pass
2	Login using the credentials	Verify and Login	As Expected	Pass
3	Select Room details	Display of Booked room details	As Expected	Pass
4				

9.2.8 View Booked Room



Update User Profile

Test Case ID	Test5	**Test Case Description**	Test the Update Profile Functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	**Date Tested**	28-Mar-2019	**Test Case (Pass/Fail/Not Execute**	Pass

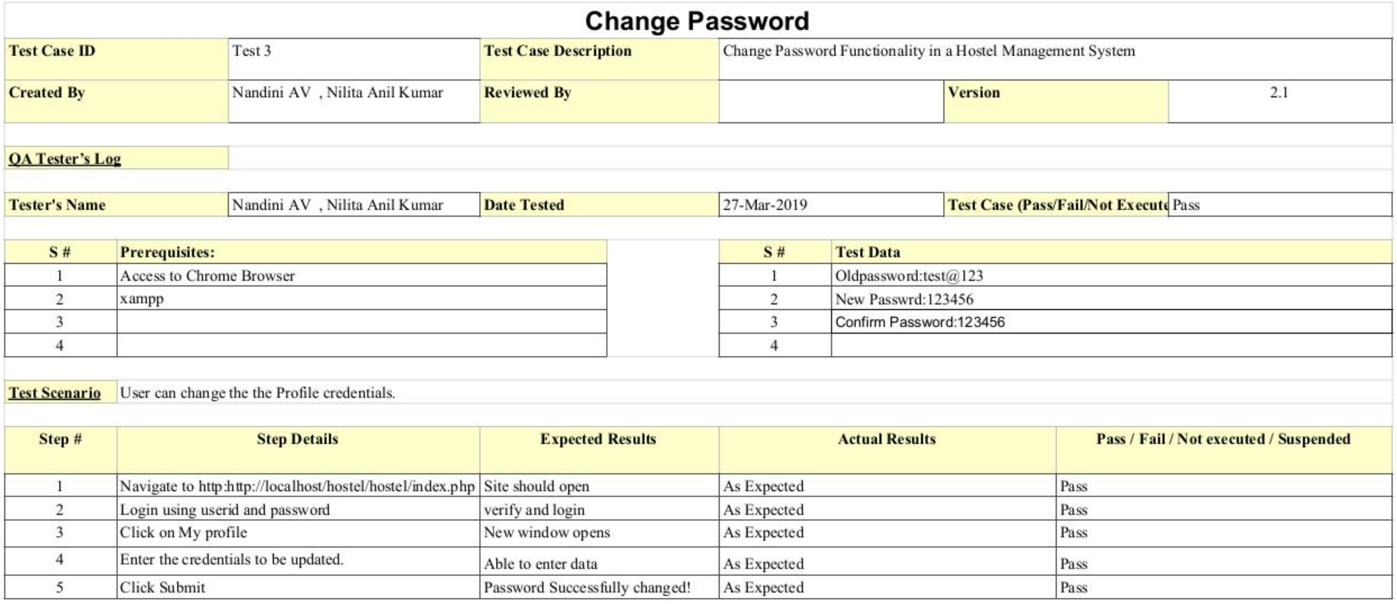
S #	Prerequisites:
1	Access to Chrome Browser
2	xampp

S #	Test Data
1	Registration No:1234
2	First Name: abc
3	Middle Name: Def
4	Last Name: defg
5	Gender: Female
6	Contact No: 990275913

Test Scenario Customer can retrieve the passowrd incase forgot

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should open	As Expected	Pass
2	Forgot password	New page should appear	As Expected	Pass
3	Enter credentials	Credentials can be entered	As Expected	Pass
4	click Submit	display the password	As Expected	Pass

9.2.9 Update User Profile



Change Password

Test Case ID	Test 3	**Test Case Description**	Change Password Functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	**Date Tested**	27-Mar-2019	**Test Case (Pass/Fail/Not Execute**	Pass

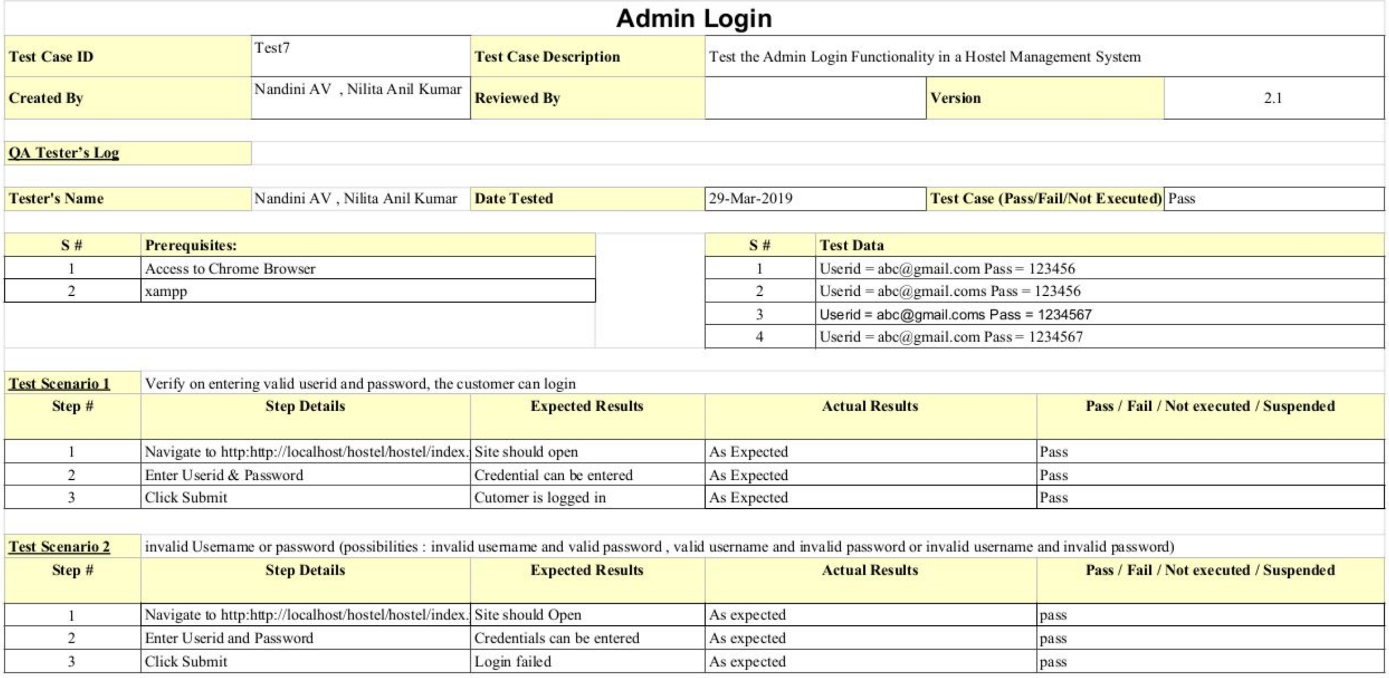
S #	Prerequisites:
1	Access to Chrome Browser
2	xampp
3	
4	

S #	Test Data
1	Oldpassword:test@123
2	New Passwrd:123456
3	Confirm Password:123456
4	

Test Scenario User can change the the Profile credentials.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.php	Site should open	As Expected	Pass
2	Login using userid and password	verify and login	As Expected	Pass
3	Click on My profile	New window opens	As Expected	Pass
4	Enter the credentials to be updated.	Able to enter data	As Expected	Pass
5	Click Submit	Password Successfully changed!	As Expected	Pass

9.2.10 Change Password



Admin Login

Test Case ID	Test7	**Test Case Description**	Test the Admin Login Functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	**Date Tested**	29-Mar-2019	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites:	S #	Test Data
1	Access to Chrome Browser	1	Userid = abc@gmail.com Pass = 123456
2	xampp	2	Userid = abc@gmail.coms Pass = 123456
		3	Userid = abc@gmail.coms Pass = 1234567
		4	Userid = abc@gmail.com Pass = 1234567

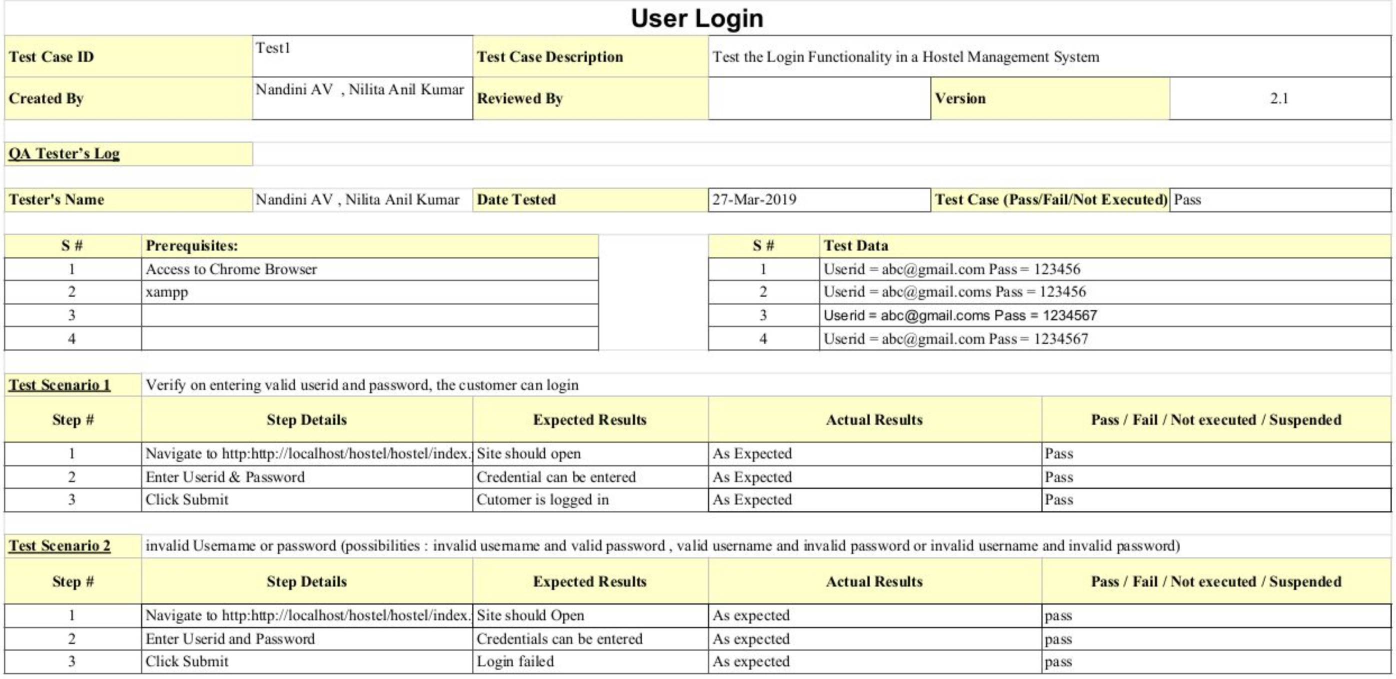
Test Scenario 1 Verify on entering valid userid and password, the customer can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should open	As Expected	Pass
2	Enter Userid & Password	Credential can be entered	As Expected	Pass
3	Click Submit	Cutomer is logged in	As Expected	Pass

Test Scenario 2 invalid Username or password (possibilities : invalid username and valid password , valid username and invalid password or invalid username and invalid password)

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should Open	As expected	pass
2	Enter Userid and Password	Credentials can be entered	As expected	pass
3	Click Submit	Login failed	As expected	pass

9.2.11 Admin Login



User Login

Test Case ID	Test1	**Test Case Description**	Test the Login Functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	**Date Tested**	27-Mar-2019	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites:	S #	Test Data
1	Access to Chrome Browser	1	Userid = abc@gmail.com Pass = 123456
2	xampp	2	Userid = abc@gmail.coms Pass = 123456
3		3	Userid = abc@gmail.coms Pass = 1234567
4		4	Userid = abc@gmail.com Pass = 1234567

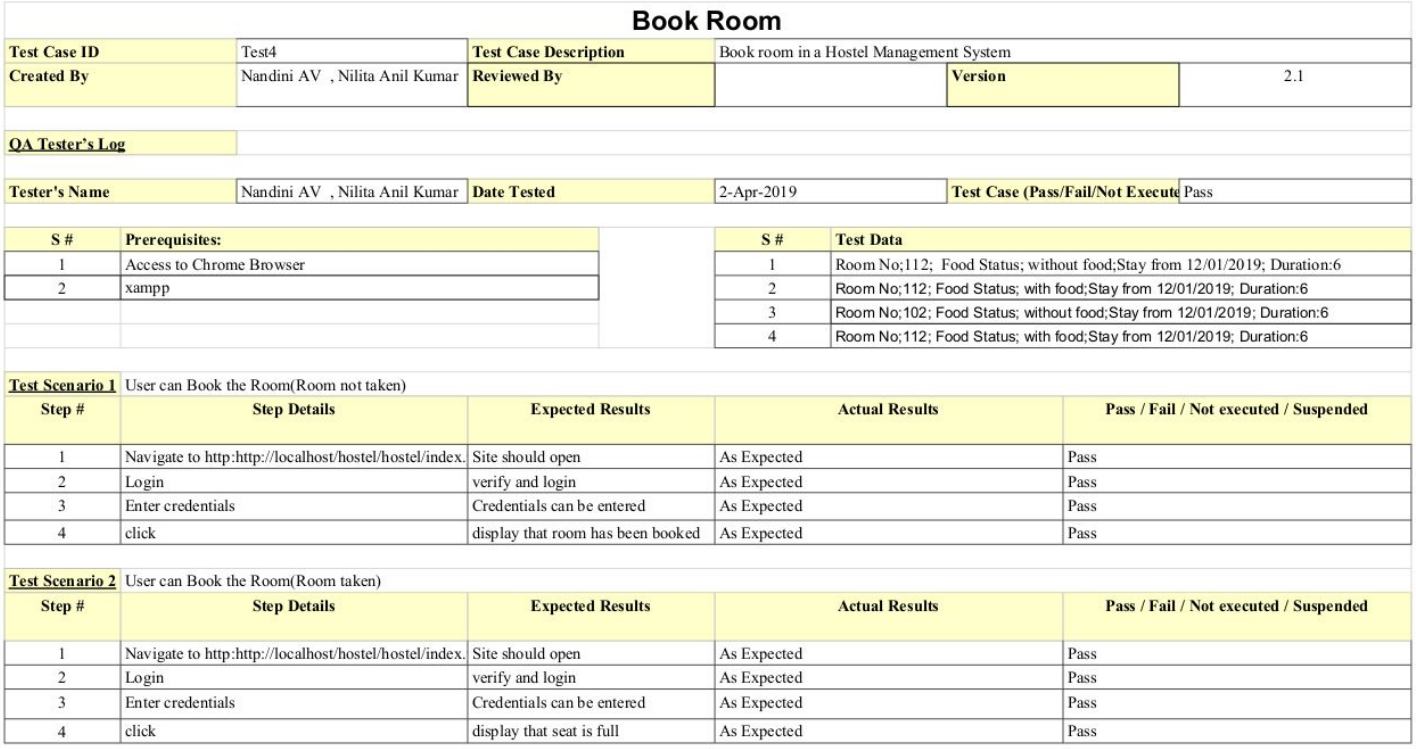
Test Scenario 1 Verify on entering valid userid and password, the customer can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should open	As Expected	Pass
2	Enter Userid & Password	Credential can be entered	As Expected	Pass
3	Click Submit	Cutomer is logged in	As Expected	Pass

Test Scenario 2 invalid Username or password (possibilities : invalid username and valid password , valid username and invalid password or invalid username and invalid password)

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should Open	As expected	pass
2	Enter Userid and Password	Credentials can be entered	As expected	pass
3	Click Submit	Login failed	As expected	pass

9.2.12 User Login



Book Room

Test Case ID	Test4	**Test Case Description**	Book room in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	**Date Tested**	2-Apr-2019	**Test Case (Pass/Fail/Not Execute**	Pass

S #	Prerequisites:
1	Access to Chrome Browser
2	xampp

S #	Test Data
1	Room No;112; Food Status; without food;Stay from 12/01/2019; Duration:6
2	Room No;112; Food Status; with food;Stay from 12/01/2019; Duration:6
3	Room No;102; Food Status; without food;Stay from 12/01/2019; Duration:6
4	Room No;112; Food Status; with food;Stay from 12/01/2019; Duration:6

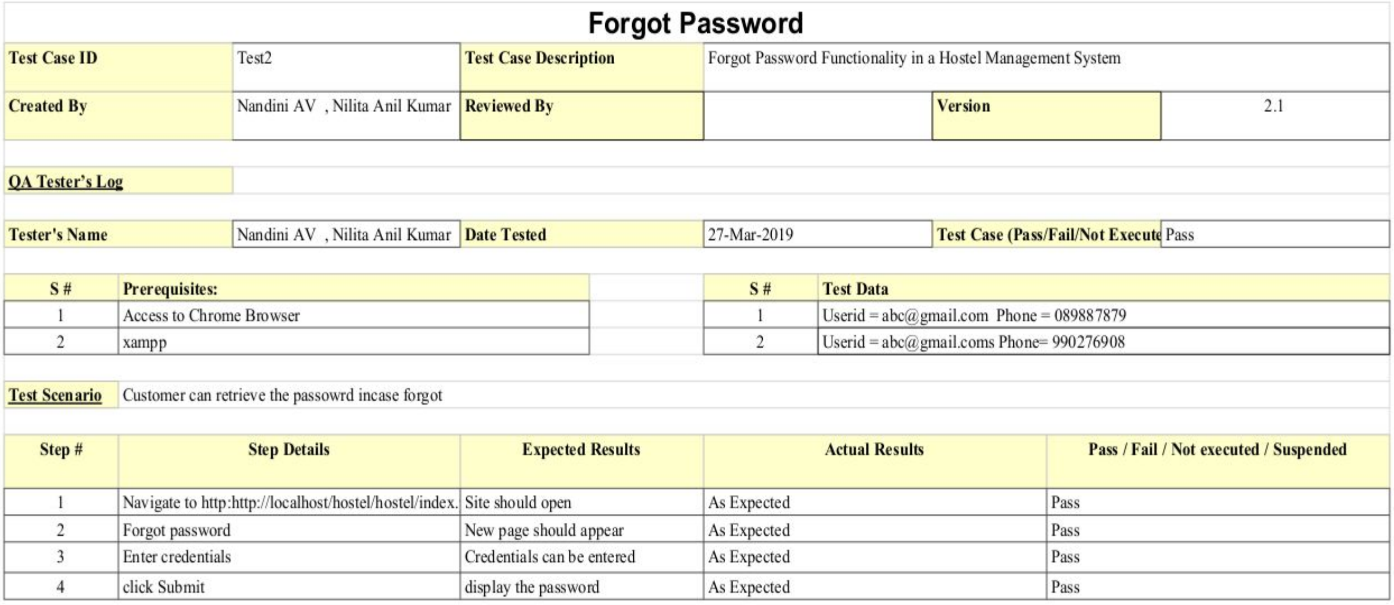
Test Scenario 1 User can Book the Room(Room not taken)

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should open	As Expected	Pass
2	Login	verify and login	As Expected	Pass
3	Enter credentials	Credentials can be entered	As Expected	Pass
4	click	display that room has been booked	As Expected	Pass

Test Scenario 2 User can Book the Room(Room taken)

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should open	As Expected	Pass
2	Login	verify and login	As Expected	Pass
3	Enter credentials	Credentials can be entered	As Expected	Pass
4	click	display that seat is full	As Expected	Pass

9.2.13 Book Room



Forgot Password

Test Case ID	Test2	**Test Case Description**	Forgot Password Functionality in a Hostel Management System		
Created By	Nandini AV , Nilita Anil Kumar	**Reviewed By**		**Version**	2.1

QA Tester's Log

Tester's Name	Nandini AV , Nilita Anil Kumar	**Date Tested**	27-Mar-2019	**Test Case (Pass/Fail/Not Execute**	Pass

S #	Prerequisites:
1	Access to Chrome Browser
2	xampp

S #	Test Data
1	Userid = abc@gmail.com Phone = 089887879
2	Userid = abc@gmail.coms Phone= 990276908

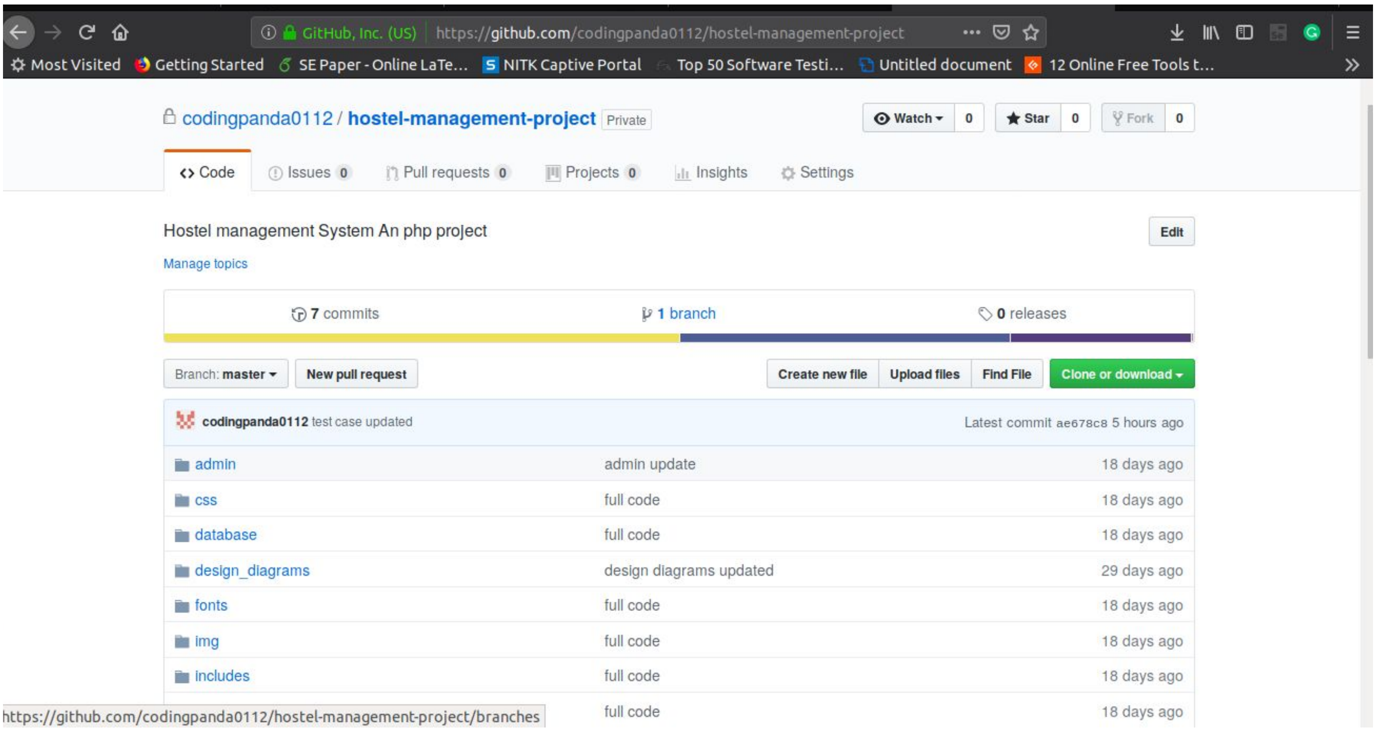
Test Scenario Customer can retrieve the passowrd incase forgot

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http:http://localhost/hostel/hostel/index.	Site should open	As Expected	Pass
2	Forgot password	New page should appear	As Expected	Pass
3	Enter credentials	Credentials can be entered	As Expected	Pass
4	click Submit	display the password	As Expected	Pass

9.2.14 Forgot Password

10.Tools Used

10.1 Version Control Tool-Git Hub



codingpanda0112 / **hostel-management-project** Private

Hostel management System An php project

Manage topics

7 commits | 1 branch | 0 releases

Branch: master

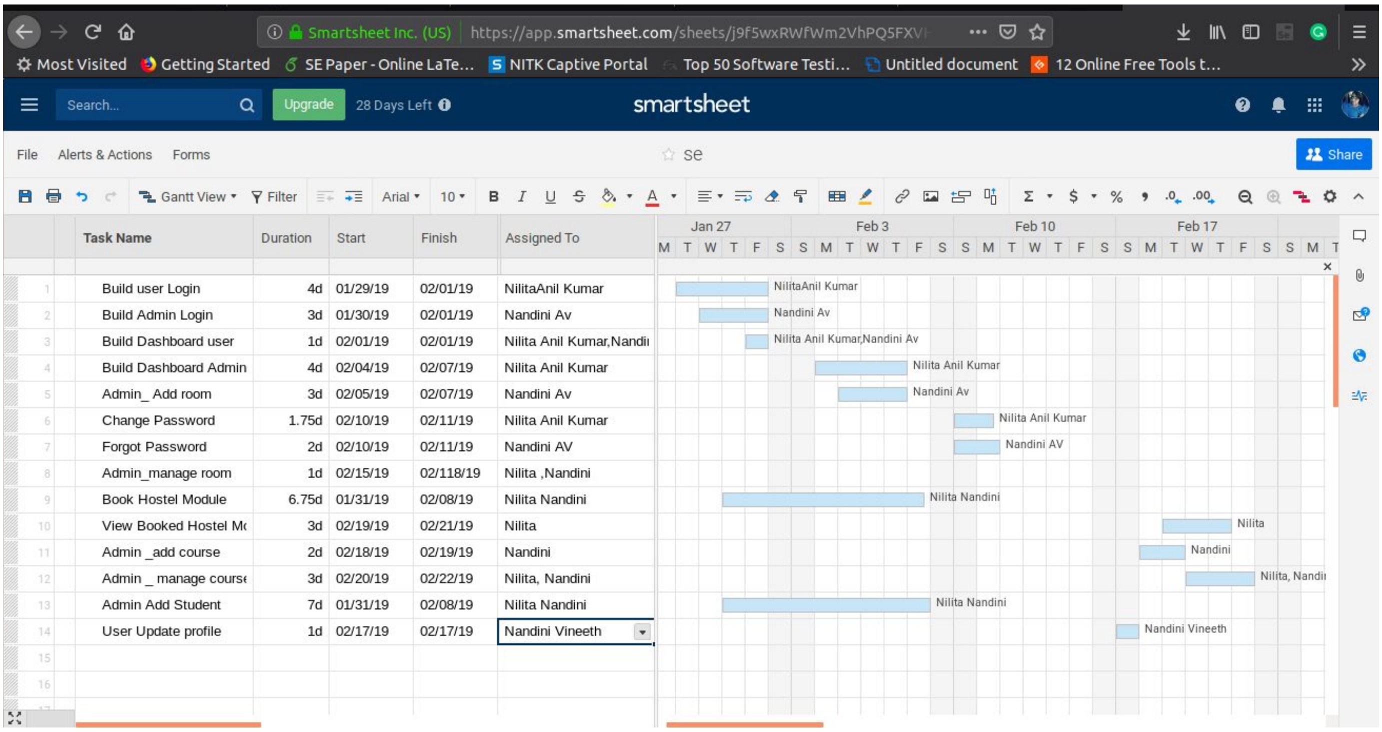
codingpanda0112 test case updated — Latest commit ae678c8 5 hours ago

admin	admin update	18 days ago
css	full code	18 days ago
database	full code	18 days ago
design_diagrams	design diagrams updated	29 days ago
fonts	full code	18 days ago
img	full code	18 days ago
includes	full code	18 days ago
	full code	18 days ago

https://github.com/codingpanda0112/hostel-management-project/branches

Fig 10.1 Git Hub Repository

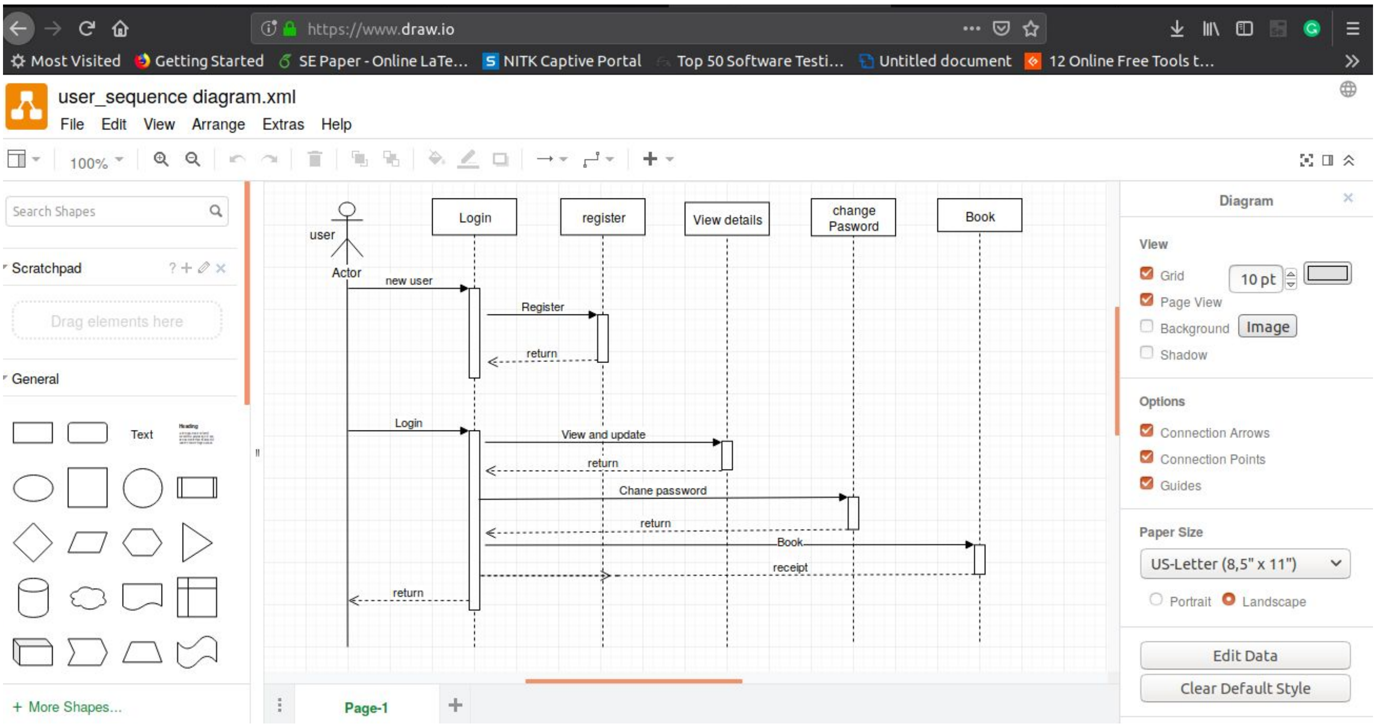
10.2 Process Tree-SmartSheet(gantt Chart)



	Task Name	Duration	Start	Finish	Assigned To
1	Build user Login	4d	01/29/19	02/01/19	NilitaAnil Kumar
2	Build Admin Login	3d	01/30/19	02/01/19	Nandini Av
3	Build Dashboard user	1d	02/01/19	02/01/19	Nilita Anil Kumar,Nandii
4	Build Dashboard Admin	4d	02/04/19	02/07/19	Nilita Anil Kumar
5	Admin_ Add room	3d	02/05/19	02/07/19	Nandini Av
6	Change Password	1.75d	02/10/19	02/11/19	Nilita Anil Kumar
7	Forgot Password	2d	02/10/19	02/11/19	Nandini AV
8	Admin_manage room	1d	02/15/19	02/118/19	Nilita ,Nandini
9	Book Hostel Module	6.75d	01/31/19	02/08/19	Nilita Nandini
10	View Booked Hostel Mc	3d	02/19/19	02/21/19	Nilita
11	Admin _add course	2d	02/18/19	02/19/19	Nandini
12	Admin _ manage course	3d	02/20/19	02/22/19	Nilita, Nandini
13	Admin Add Student	7d	01/31/19	02/08/19	Nilita Nandini
14	User Update profile	1d	02/17/19	02/17/19	Nandini Vineeth
15					
16					

Fig 10.2 Gantt Chart

10.3 Design Tool-Draw.io



https://www.draw.io

user_sequence diagram.xml

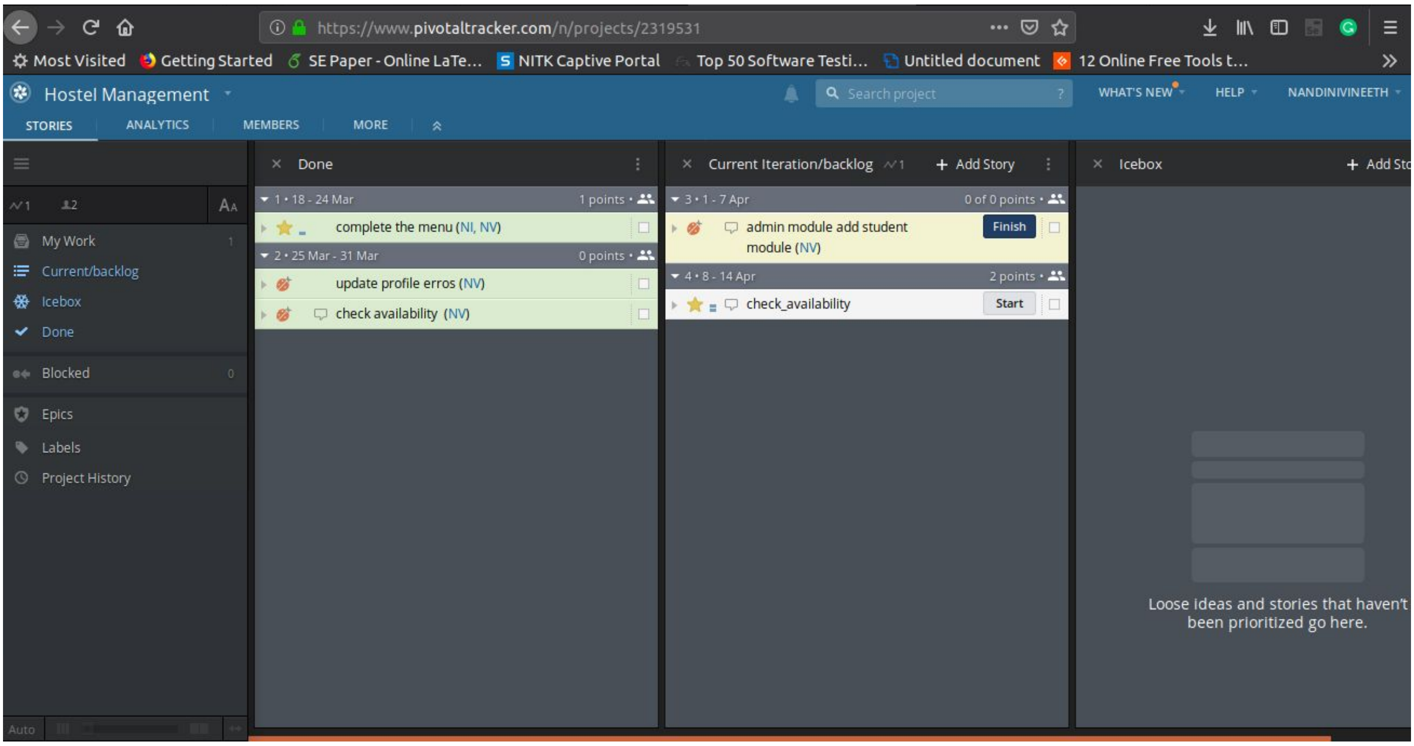
View
☑ Grid 10 pt
☑ Page View
□ Background Image
□ Shadow

Options
☑ Connection Arrows
☑ Connection Points
☑ Guides

Paper Size
US-Letter (8,5" x 11")
○ Portrait ● Landscape

Fig 10.3 Design Tool

10.4 Bug tracking- Pivotal Tracker



https://www.pivotaltracker.com/n/projects/2319531

Hostel Management

Done
1 • 18 - 24 Mar — 1 points
- complete the menu (NI, NV)

2 • 25 Mar - 31 Mar — 0 points
- update profile erros (NV)
- check availability (NV)

Current Iteration/backlog
3 • 1 - 7 Apr — 0 of 0 points
- admin module add student module (NV) — Finish

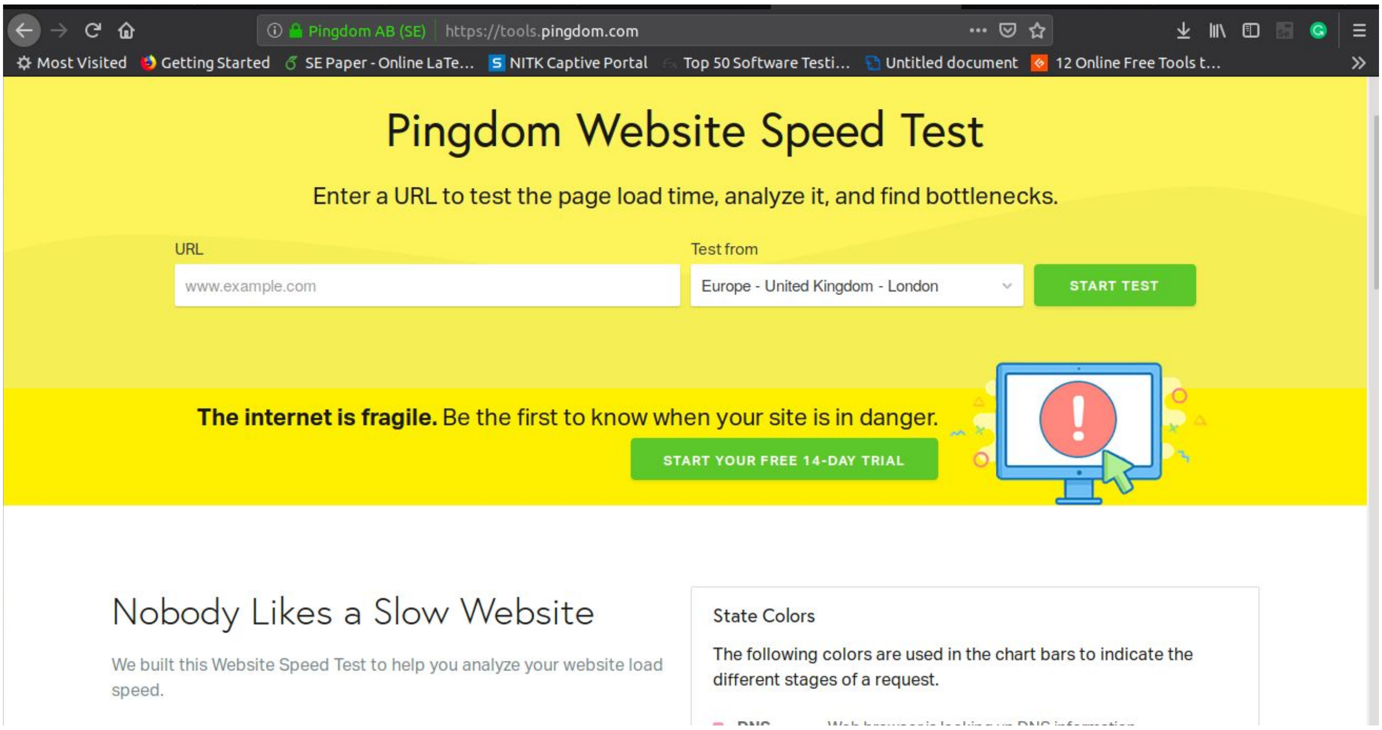
4 • 8 - 14 Apr — 2 points
- check_availability — Start

Icebox

Loose ideas and stories that haven't been prioritized go here.

Fig 10.4 Bug Tracking

10.5 Performance Testing-Pingdom Tools



Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.

URL: www.example.com

Test from: Europe - United Kingdom - London

START TEST

The internet is fragile. Be the first to know when your site is in danger.

START YOUR FREE 14-DAY TRIAL

Nobody Likes a Slow Website

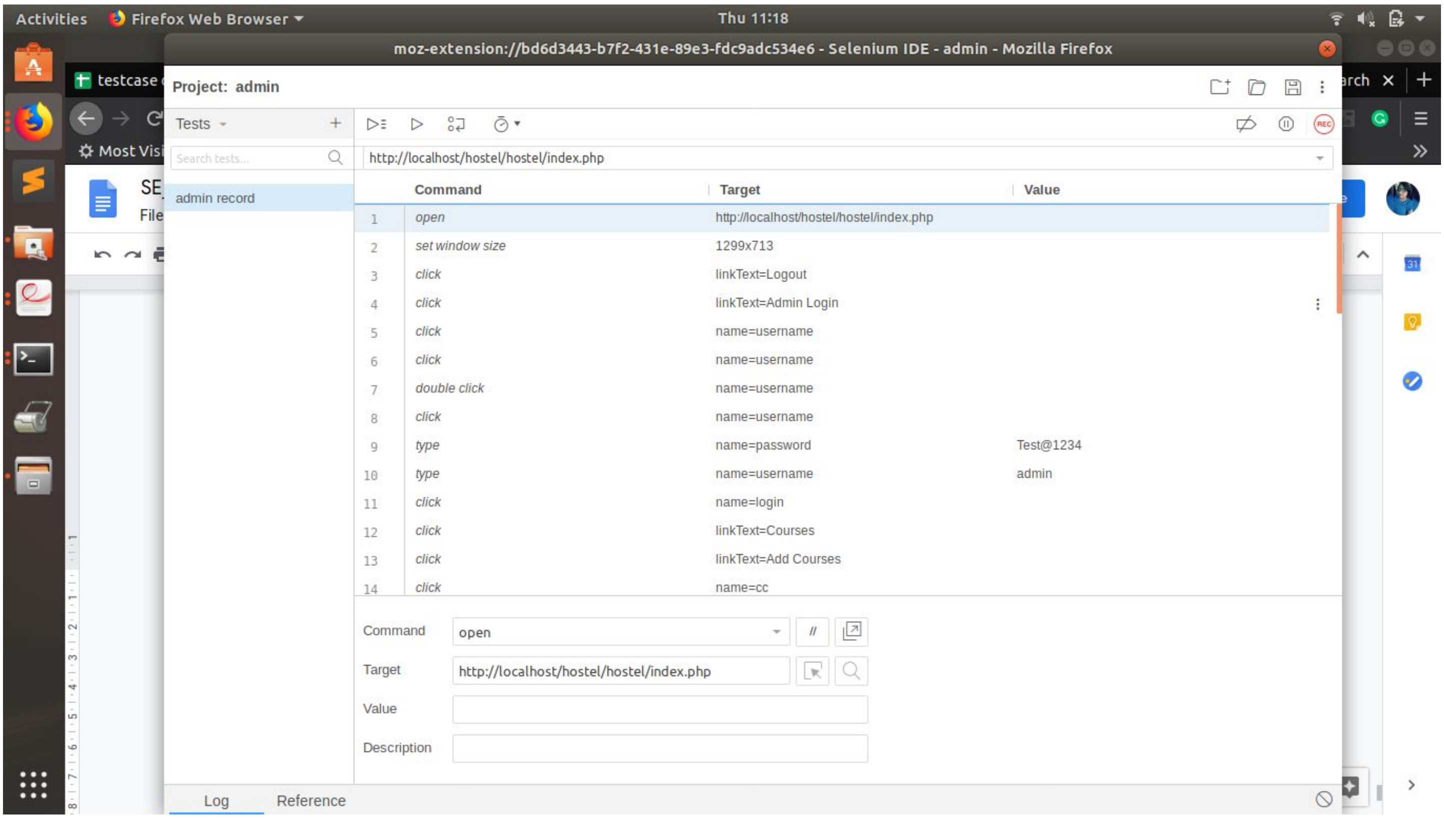
We built this Website Speed Test to help you analyze your website load speed.

State Colors

The following colors are used in the chart bars to indicate the different stages of a request.

Fig 10.5 Load Testing

10.6 Automation Tool- Selenium



moz-extension://bd6d3443-b7f2-431e-89e3-fdc9adc534e6 - Selenium IDE - admin - Mozilla Firefox

Project: admin

http://localhost/hostel/hostel/index.php

	Command	Target	Value
1	open	http://localhost/hostel/hostel/index.php	
2	set window size	1299x713	
3	click	linkText=Logout	
4	click	linkText=Admin Login	
5	click	name=username	
6	click	name=username	
7	double click	name=username	
8	click	name=username	
9	type	name=password	Test@1234
10	type	name=username	admin
11	click	name=login	
12	click	linkText=Courses	
13	click	linkText=Add Courses	
14	click	name=cc	

Command: open

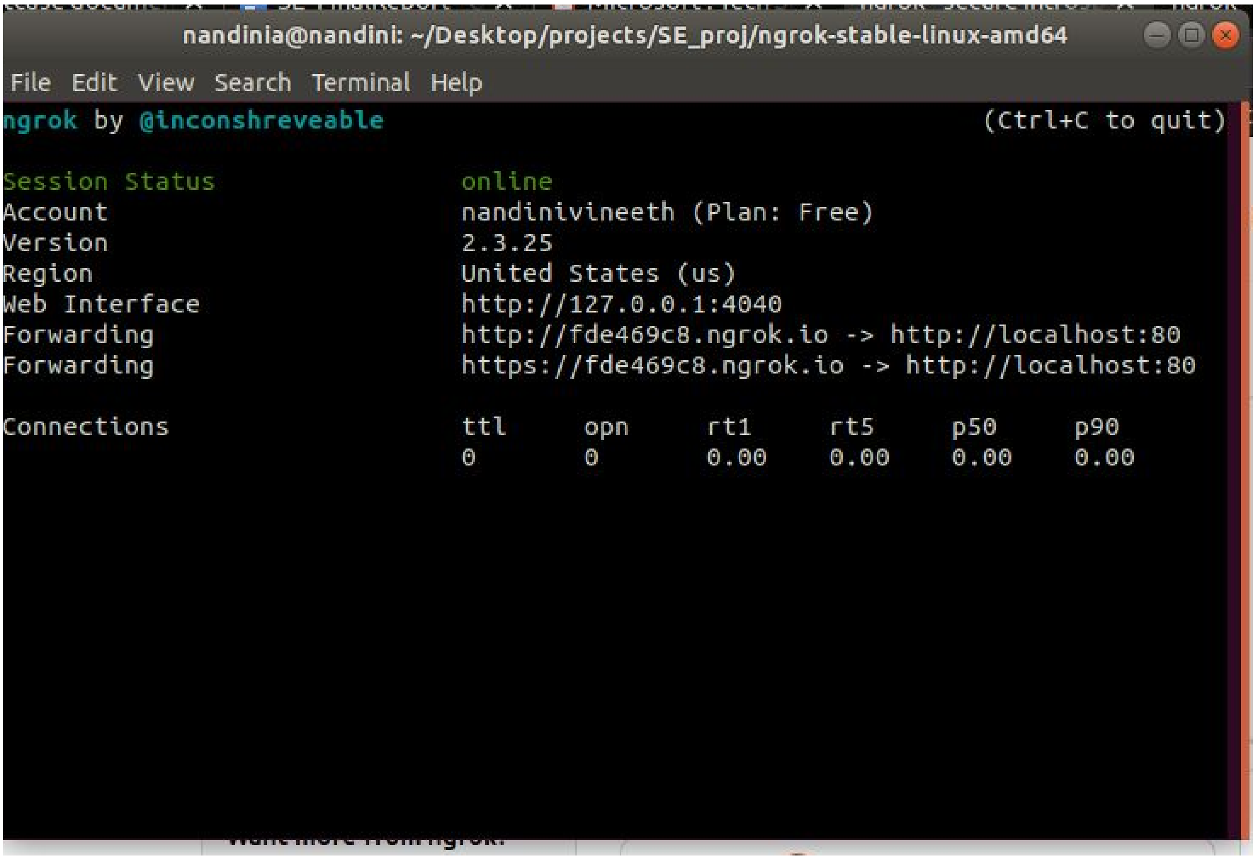
Target: http://localhost/hostel/hostel/index.php

Value:

Description:

Fig 10.6 Automation Tool

10.7 Webhook development tool and debugging- ngrok

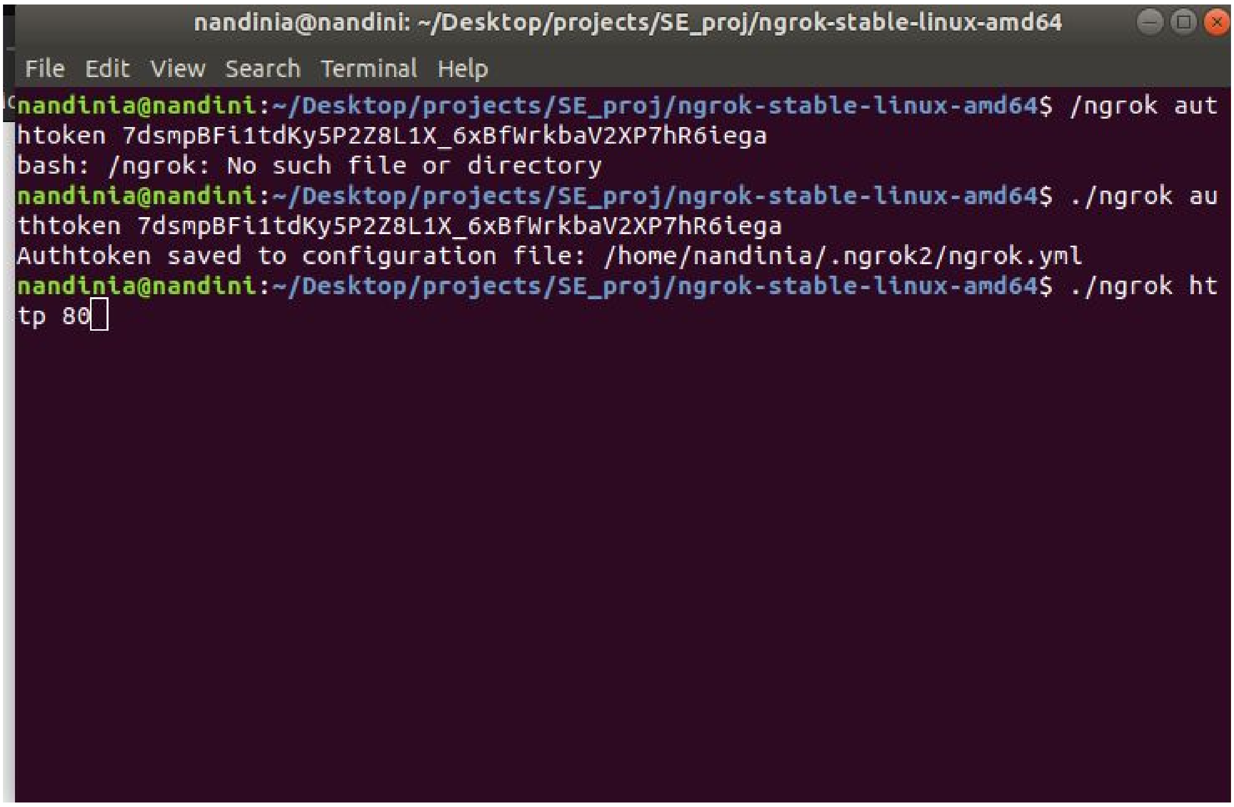

nandinia@nandini: ~/Desktop/projects/SE_proj/ngrok-stable-linux-amd64
File Edit View Search Terminal Help

```
ngrok by @inconshreveable                                       (Ctrl+C to quit)

Session Status                online
Account                       nandinivineeth (Plan: Free)
Version                       2.3.25
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://fde469c8.ngrok.io -> http://localhost:80
Forwarding                    https://fde469c8.ngrok.io -> http://localhost:80

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Fig 10.7.1 ngrok running-creating public Url and tracking


nandinia@nandini: ~/Desktop/projects/SE_proj/ngrok-stable-linux-amd64
File Edit View Search Terminal Help

```
nandinia@nandini:~/Desktop/projects/SE_proj/ngrok-stable-linux-amd64$ /ngrok aut
htoken 7dsmpBFi1tdKy5P2Z8L1X_6xBfWrkbaV2XP7hR6iega
bash: /ngrok: No such file or directory
nandinia@nandini:~/Desktop/projects/SE_proj/ngrok-stable-linux-amd64$ ./ngrok au
thtoken 7dsmpBFi1tdKy5P2Z8L1X_6xBfWrkbaV2XP7hR6iega
Authtoken saved to configuration file: /home/nandinia/.ngrok2/ngrok.yml
nandinia@nandini:~/Desktop/projects/SE_proj/ngrok-stable-linux-amd64$ ./ngrok ht
tp 80
```

Fig 10.7.2 Running ngrok