Final Report of ITA Project

Project Title:

Online Guest House Booking System NITK

Submitted By:

Siddesh L C (16CO144)

Praveen Naik S (16CO133)

in partial fulfillment of the requirements for the award of the degree of

Computer Science & Engineering

Department of Computer Science and Engineering

National Institute of Technology Karnataka

July-Nov 2018

DECLARATION

We certify that the report on ‘Online guest house booking system NITK’ which is being submitted as record of my course is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or Institution for the award of any degree.

Siddesh L C

16CO144

Praveen Naik S

16CO133

TABLE OF CONTENTS

List of Tables and Figures

Abstract

Objective

The objective of the project is to build a web application for guest house booking system in NITK. This system allows students of NITK to book rooms in guest house in advance for their parents and relatives. This website provides information about rooms availability, checkin, checkout details etc.
As of now there is no online facility for advance booking of Guest house rooms in NITK. So, we are developing a website which allows students to book rooms in advance directly through website.
The system allows only NITK students to book the rooms in advance. This will be done by confirming the students using the institute email address (xxxx@nitk.edu.in) and other required credentials.

Frameworks

- Database: SQLite3
- Web Development: Django

This system helps the student by reducing the burden of physically booking rooms at Guest House, taking permission from Registar etc. As the system itself makes sure that the rooms are booked by NITK students only, there is no need of confirming the authenticity of students.

Introduction

1.1 Purpose

The purpose of this project is to serve students of NITK to help them book rooms of guest house in NITK. There are in total three guest houses present in NITK. The present system of booking involves physical presence of students to book rooms and take permission from Registar and present the copy of permission to staff of the guest house. This causes burden on students. In order to solve this issue we come up with a website which allows students to book rooms online by authenticating that they belong to NITK via xxxx@nitk.edu.in account.

1.2 Objective

In this world of growing technologies everything can be digitized. With a higher efficiency, digitized systems are always preferred options. We are living at a time when digitization of room bookings has never been this big. Thus, there is a need for a system which can help students of NITK to book rooms of guest house online. This is where our project comes in.

Requirements Analysis and Specification

This system allows students of NITK to book rooms in guest house in advance for their parents and relatives. This website provides information about rooms availability, checkin, checkout details etc. The main functionality of this application is to allow students to book rooms in NITK guest house online. User needs to login using valid mail provided by the institution and can create an account, check availability of rooms and book rooms accordingly for parents,relatives visiting the campus of NITK.

2.1 Software Requirements

DBMS:
When the development platforms are looked at, Oracle, MS SQL Server 2000,SQLite3 are the possible solutions. As stated, as we chose Django as the development platform SQLite3 was our database of choice as it is available in Django by default. We believe that this was the best solution for us.

Development:
For developing our system, we used the following tools and languages:

- Python3 is the programming language for main development
- Django is the run-time environment that executes Python code outside of a browser.
- HTML5, Bootstrap for making front-end of the webpages. Bootstrap is a collection of CSS files that will help us use CSS modules directly.

Web Server : We have hosted our application on herokuapp. We can run our website using this url: nitk-ghbs.herokuapp.com

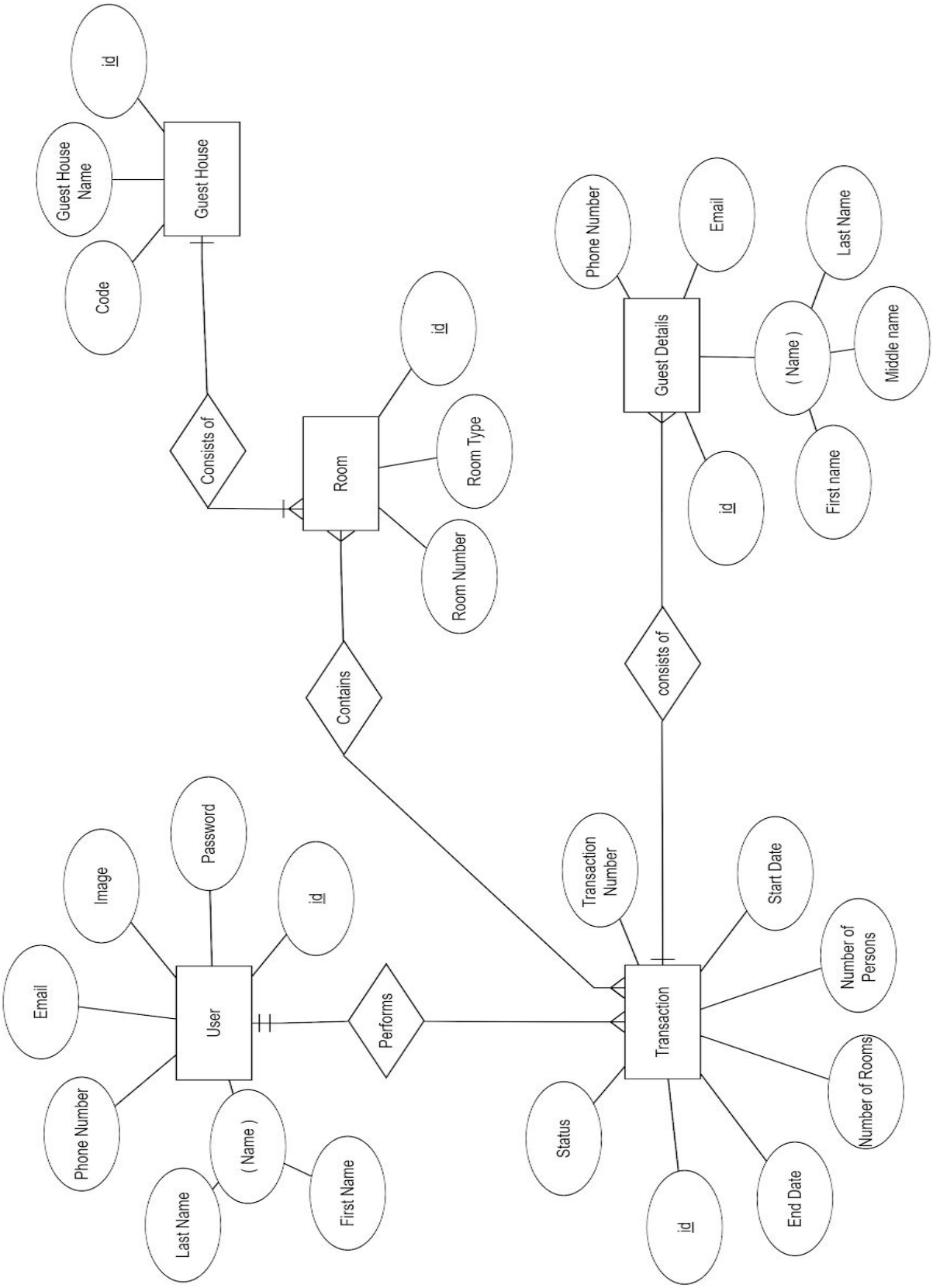
Database Design

3.1 ER Diagram

An **entity relationship diagram** (**ERD**) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.
An Entity Relationship **Diagram (ERD**) is a snapshot of data structures. An Entity Relationship Diagram shows entities (tables) in a database and relationships between tables within that database.

ER Diagram for our app:

The following image is the ER diagram of our project named Football game management system.


id
Guest House Name
Guest House
Code
Consists of
Room
id
Room Type
Room Number
Contains
Guest Details
Phone Number
Email
id
(Name)
Last Name
Middle name
First name
consists of
Image
Password
Email
User
id
Performs
Phone Number
(Name)
Last Name
First Name
Transaction
Transaction Number
Start Date
Number of Persons
Number of Rooms
Status
id
End Date

The ER diagram consists of Entities, Attributes and relationships among entities. The entities are represented using rectangles, Attributes using oval and relationships via lines between entities. There are five entities in our namely User, Guest house, Guest details, Rooms, Transactions.Each entity has a primary key and a foreign key if referenced by other entities.

The ER diagram consists of Different kinds of attributes such as key attribute, derived attribute, multi-valued attributes and composite attribute. Key attribute in given ER diagram is represented with an underline, Composite is like attributes derived from an attribute and multi-valued attribute using double oval and derived attribute using dashed oval.

The relationship(cardinality) between various entities is as follows:
User - Transaction : One-Many relationship.
Room- Transaction : Many-Many relationship.
Guest house - Rooms : One-Many relationship.
Transaction- Guest details : Many-Many relationship.

3.2 Tables in the database

User table

Field	Type	About
id	int(6)	This is the unique ID given to the user. This field is the primary key of the table.
firstname	varchar(30)	The first name of theuser. It is part of the composite attribute name.
lastname	varchar(30)	The last name of the user. It is part of the composite attribute name.
Phone number	int(10)	The phone number of the user.

email	varchar(30)	The email id of the user.
image	varchar(30)	The display picture of the user.
password	binary(100)	The password provided by the user stored after hashing it.

Guest details table

Field	**Type**	**About**
id	int(3)	Unique id generated for every guest.
firstname	varchar(30)	The first name of the guest. It is part of the composite attribute name.
middlename	varchar(30)	The middle name of the guest. It is part of the composite attribute name.
lastname	varchar(30)	The last name of the guest. It is part of the composite attribute name.
Phone number	int(10)	The contact number of the guest.
email	varchar(30)	Email of the guest.
id	int(3)	This is the foreign key referenced from transaction id.

Guest house table

Field	**Type**	**About**

Gid	int(6)	A unique id given to every Guest house. This is the primary key of the table.
Guest house name	varchar(30)	The name of the guest house where user books room.
Code	varchar(10)	Unique code given to each of the guest houses.

RoomTable

Field	**Type**	**About**
RoomID	int(5)	The Room id of the Rooms in each guest houses.
Room number	int(5)	Room number of the rooms in guest house.
Room type	varchar(15)	Type of the rooms available in guest house
Gid	int(3)	Foreign key referencing the guest house id in guest house table.

Transaction table

Field	**Type**	**About**
Transaction_ID	int(8)	The transaction Id of each of the bookings.
Start date	Date	The checkin date for the booking of room
End date	Date	The checkout date of the guest from the rooms.

Number of persons	int(3)	Number of persons for which room has to be booked.
Number of rooms	int(3)	Number of rooms booked by each user per transaction.
Status	varchar(10)	Status whether the rooms are avilable or not for each transaction.
Transaction Number	int(10)	Transaction done by each user is numbered and stored in this field.
uid	int(3)	Foreign key of the user referenced from user table who carries out the transaction.

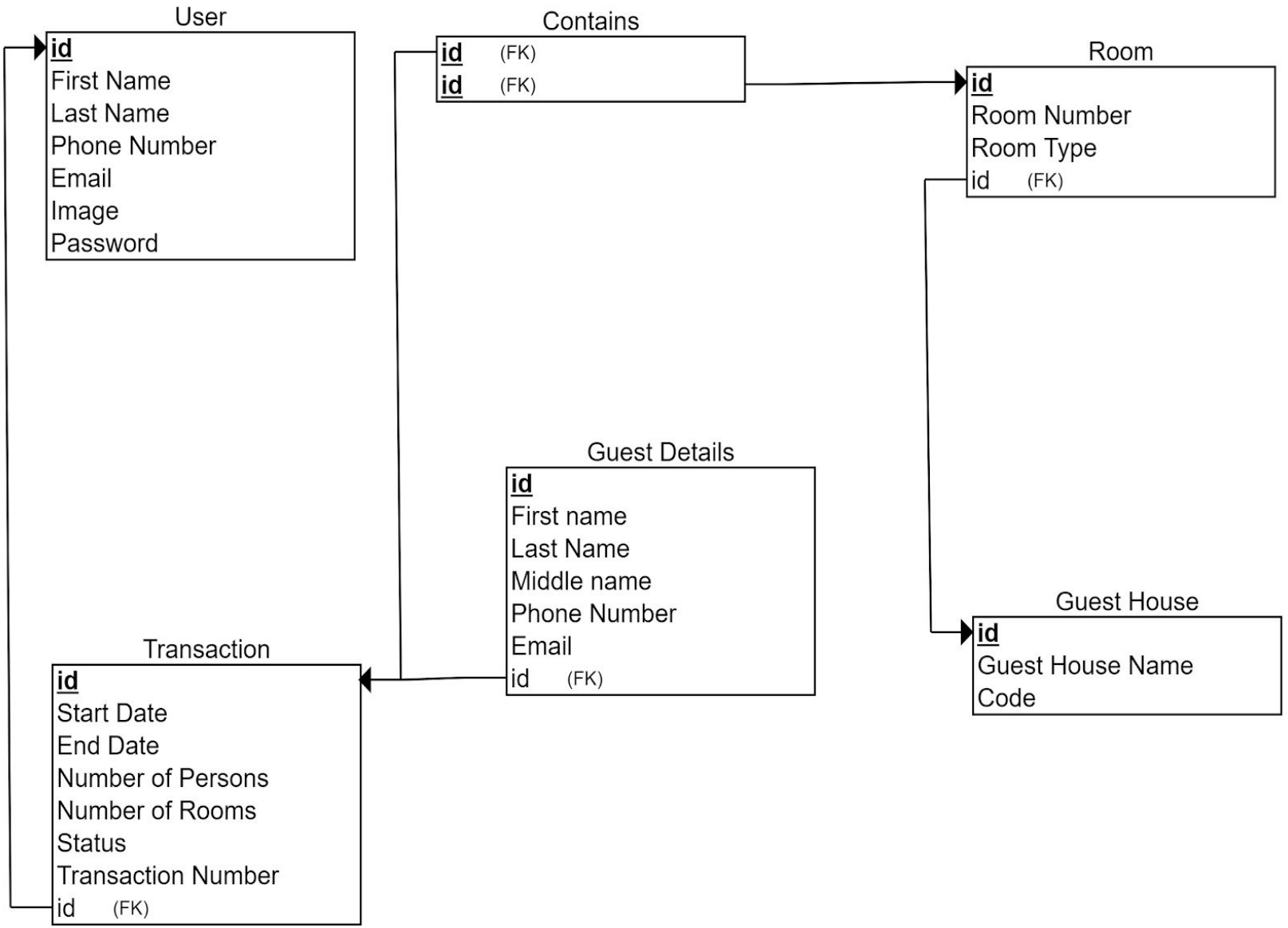
Sessions Table

Field	**Type**	**About**
session_id	varchar(128)	This is the unique id given to any login session of the player.
expires	int(11) unsigned	The time in milliseconds after which the session will expire.
data	text	The field which stores data for the cookie.

The relational schema for our app :

A **relation schema** is essentially the **schema** for a table. In a **relational** database each table can be referred to as a "**relation**" . Hence a **relational schema** is the design for the table.

The relational schema for our app is displayed below.


User
id
First Name
Last Name
Phone Number
Email
Image
Password
Contains
id (FK)
id (FK)
Room
id
Room Number
Room Type
id (FK)
Guest Details
id
First name
Last Name
Middle name
Phone Number
Email
id (FK)
Guest House
id
Guest House Name
Code
Transaction
id
Start Date
End Date
Number of Persons
Number of Rooms
Status
Transaction Number
id (FK)

A relation schema consists of primary key, Foreign key and cardinality between entities.

Primary key : A primary key is a special relational database table column (or combination of columns) designated to uniquely identify all table records. A primary key's main features are: It must contain a unique value for each row of data. It cannot contain null values.

Foreign key : In the context of relational databases, a foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table or the same table. In simpler words, the foreign key is defined in a second table, but it refers to the primary key or a unique key in the first table.

Cardinality : In the context of databases, cardinality refers to the uniqueness of data values contained in a column. High cardinality means that the column contains a large percentage of totally unique values. Low cardinality means that the column contains a lot of “repeats” in its data range.

Project Components

4.1 Front End Design:

4.1.1 Framework:

We have used **Express** as the backend framework. For the front end, we have used **CSS3, Bootstrap 4, HTML5 and JavaScript.**

4.1.2 UI Components:

- **Home Page:** This animative page welcomes users to our web application and tells users about Guest houses in nitk, gallery of images of guest house. The home page has various sections like About, Gallery and contact information along with the kinds of rooms available and its price.

- **Login:** The login page allows users to enter identifier information into our system in order to access that system. Our login form requires users to enter a valid institution mail and a password. And for manager a separate popup login page will be displayed.

- **SignUp:** This form allows people to become registered users. The users should give a valid institution mail, choose a new password which follows a few industry defined rules. Users can also provide a profile picture here.

- **Profile:** This is the page the user is directed to when he/she logs in. The user can view his details here and edit them if he feels necessary. Also he can change his account password.

- **Room booking form:** This page has a form which asks user to fill check in and check out dates for rooms. On submitting the system takes them to rooms availability page in guest houses.

- **Availability :** The availability page shows number of rooms available in each of the three guest houses and has a button which allows users to get details of rooms available in guest house. Under details section , the number of rooms available in each category is shown along with an option to book rooms of certain kind by specifying number of rooms and number of guest.

- **My bookings:** My bookings contains the information about all the bookings done by the user with his/her account and details of the bookings done every time. This also gives option to cancel rooms before checkin and shows the information about cancellation of rooms.

- **User profile :** This page contains the personal information about the user. He/she can edit information at any point of time and allows to provide their profile picture. This allows users to change their passwords incase they want to change the password.

- **Manager profile :** Manager profile page has a bundle of functionalities. This shows the information about recent bookings and rooms in each of the guest houses separately. Also managers can change their passwords by going to manager account page.

- **Room booking statistics :** This page shows the graph of rooms booked by users with respect to date. This helps manager to have a clear picture about the number of rooms available and number of rooms booked.

- **Navbar:** The navbar is present at the top of every page in our application. This UI component allows users to easily navigate from one page to another, often giving users the feeling that they are switching between tabs.

4.2 Security Measures:

1.SQL INJECTIONS

What it is: SQL injection is a type of web application security vulnerability in which an attacker attempts to use application code to access or corrupt database content. If successful, this allows the attacker to create, read, update, alter, or delete data stored in the back-end database. SQL injection is one of the most prevalent types of web application security vulnerabilities.

How we solved it: **Prepared Statements (with Parameterized Queries)**

The use of prepared statements with variable binding (aka parameterized queries) is how all developers should first be taught how to write database queries. They are simple to write, and easier to understand than dynamic queries. Parameterized queries force the developer to first define all the SQL code, and then pass in each parameter to the query later. This coding style allows the database to distinguish between code and data, regardless of what user input is supplied.

Safe Java Prepared Statement Example
The following code example uses a PreparedStatement, Java's implementation of a parameterized query, to execute the same database query.

```
String custname = request.getParameter("customerName"); // This should
REALLY be validated too
 // perform input validation to detect attacks
 String query = "SELECT account_balance FROM user_data WHERE
user_name = ? ";

 PreparedStatement pstmt = connection.prepareStatement( query );
 pstmt.setString( 1, custname);
 ResultSet results = pstmt.executeQuery( );
```

Our website uses one way hashing to store important details like passwords using an optimal salt value of ten.

The salt value is generated at random and can be any length, in this case the salt value is 8 bytes (64-bit) long. The salt value is appended to the plaintext password and then the result is hashed, this is referred to as the hashed value. Both the salt value and hashed value are stored.

Username	Salt value	String to be hashed	Hashed value = SHA256 (Salt value + Password)
user1	E1F53135E559C253	password123E1F53135E559C253	72AE25495A7981C40622D49F9A52E4F1565C90F048F59027BD9C8C8900D5C3D8
user2	84B03D034B409D4E	password12384B03D034B409D4E	B4B6603ABC670967E99C7E7F1389E40CD16E78AD38EB1468EC2AA1E62B8BED3A

As the table above illustrates, different salt values will create completely different hashed values, even when the plaintext passwords are exactly the same. Additionally, dictionary attacks are mitigated to a degree as an attacker cannot practically precompute the hashes.

2. CROSS SITE SCRIPTING (XSS)

What it is: Cross-site scripting (XSS) targets an application's users by injecting code, usually a client-side script such as JavaScript, into a web application's output. The concept of XSS is to manipulate client-side scripts of a web application to execute in the manner desired by the attacker. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites, or redirect the user to malicious sites.

How we solved it: In the forms we use, we use hashing to store passwords and the other fields are accessed via django rather than directly querying the database. So, Cross Site Scripting won't affect the website.

3 . BROKEN AUTHENTICATION & SESSION MANAGEMENT

What it is: Broken authentication and session management encompass several security issues, all of them having to do with maintaining the identity of a user. If authentication credentials and session identifiers are not protected at all times an attacker can hijack an active session and assume the identity of a user.

How we solved it: To ensure that the identity of a user is kept safe, we use password hashing and session timeout. As already mentioned, the hashing ensures that access to passwords cannot be cracked even if one has access to the database. Also, we use session managers which ensure that a user session stops after a certain amount of time and he/she would have to login again.

4. INSECURE DIRECT OBJECT REFERENCES

What it is: Insecure direct object reference is when a web application exposes a reference to an internal implementation object. Internal implementation objects include files, database records, directories, and database keys. When an application exposes a reference to one of these objects in a URL hackers can manipulate it to gain access to a user's personal data.

How we solved it: One common way to access objects like profiles is to use profile IDs in the URLs. So, we do not use profile IDs. Instead, we have a single URL for all the profiles. None of the URLs that we use, contain a reference to an internal object. Moreover any irregular or wrong URL is processed to be a 404 error and necessary action is taken.

5. SECURITY MISCONFIGURATION

What it is: Security misconfiguration encompasses several types of vulnerabilities all centered on a lack of maintenance or a lack of attention to the web application configuration. A secure configuration must be defined and deployed for the application, frameworks, application server, web server, database server, and platform. Security misconfiguration gives hackers access to private data or features and can result in a complete system compromise.

How we solved it: The security configurations used by us are the ones provided by django itself. The django modules have been known to use custom cryptographic libraries and multi-step authorisation and authentication and permission guarantees in order to avoid such vulnerabilities.

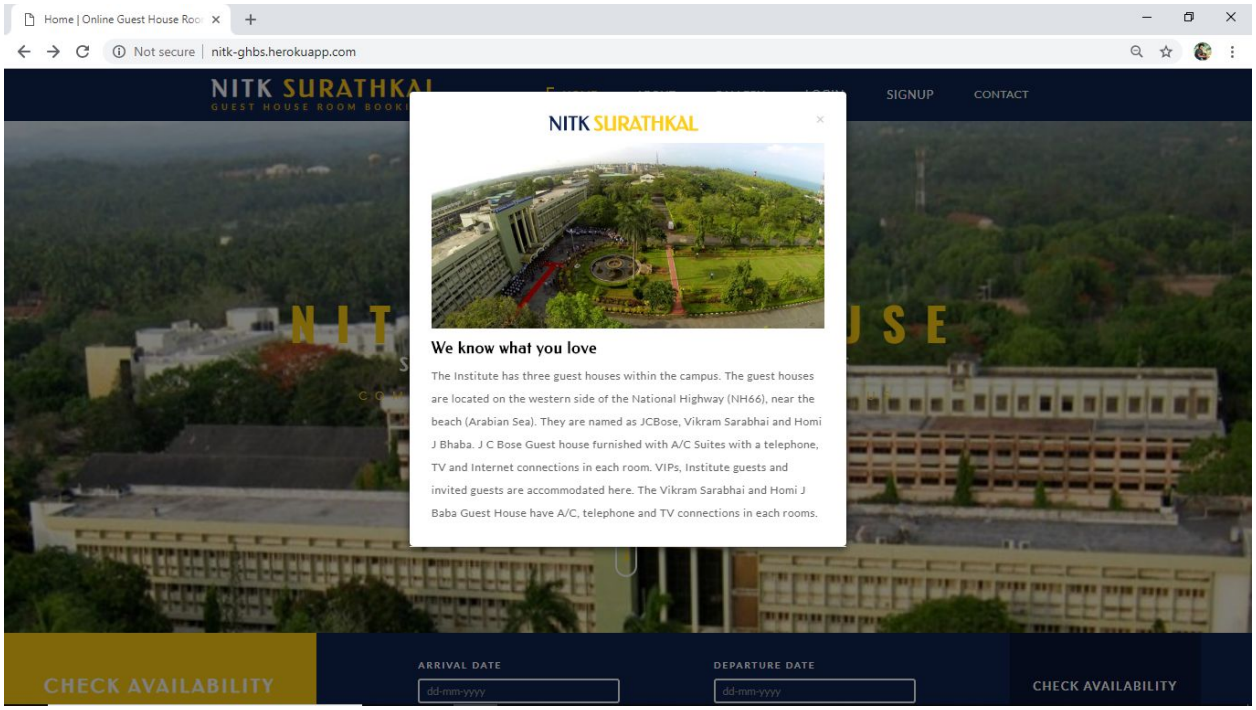
6. CROSS-SITE REQUEST FORGERY (CSRF)

What it is: Cross-Site Request Forgery (CSRF) is a malicious attack where a user is tricked into performing an action he or she didn't intend to do. A third-party website will send a request to a web application that a user is already authenticated against (e.g. their bank). The attacker can then access functionality via the victim's already authenticated browser. Targets include web applications like social media, in browser email clients, online banking, and web interfaces for network devices.

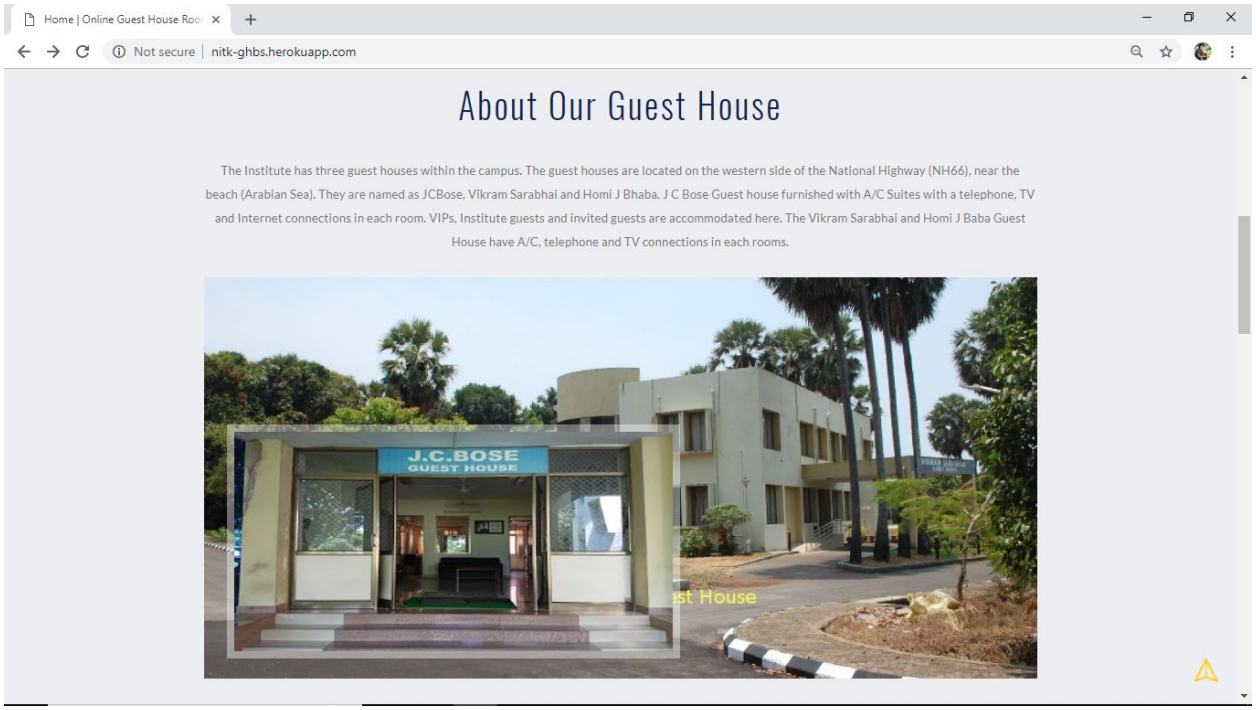
How we solved it: On each of our sessions, we use CSRF tokens, which are generated randomly with timestamp and a short lifetime.

Results & Discussion

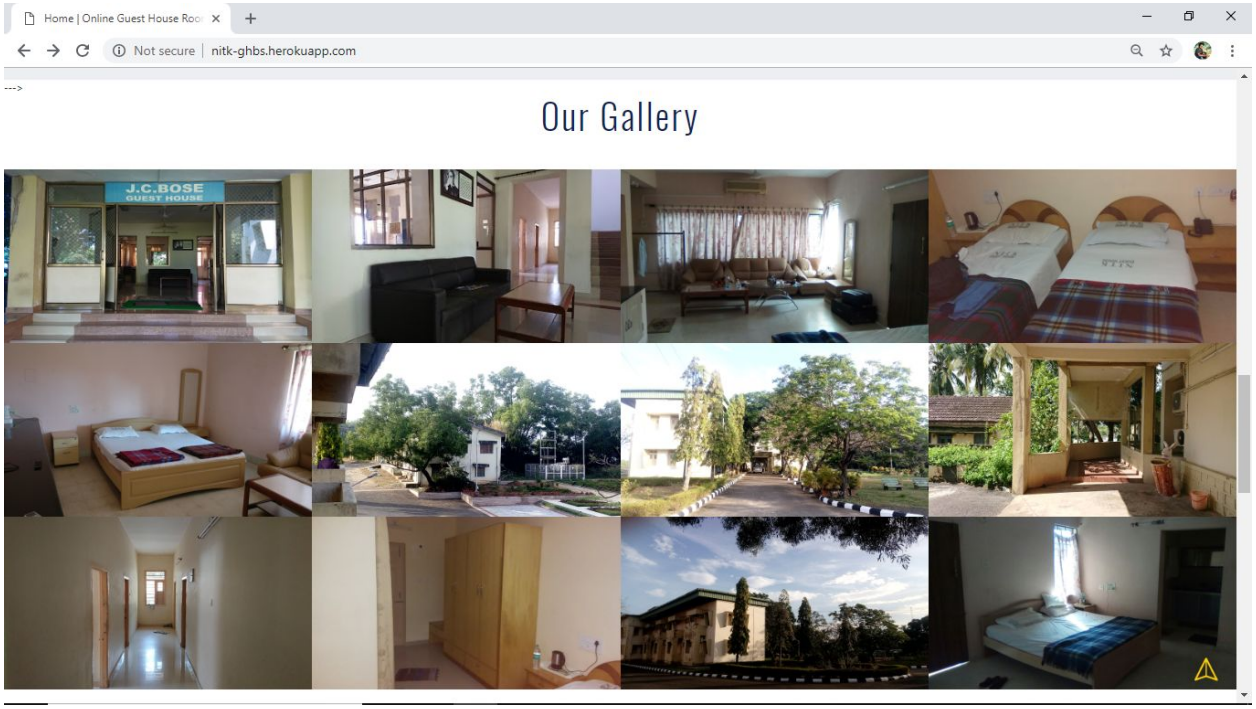
The home page of the website which hosts picture of NITK in the backgroiund.



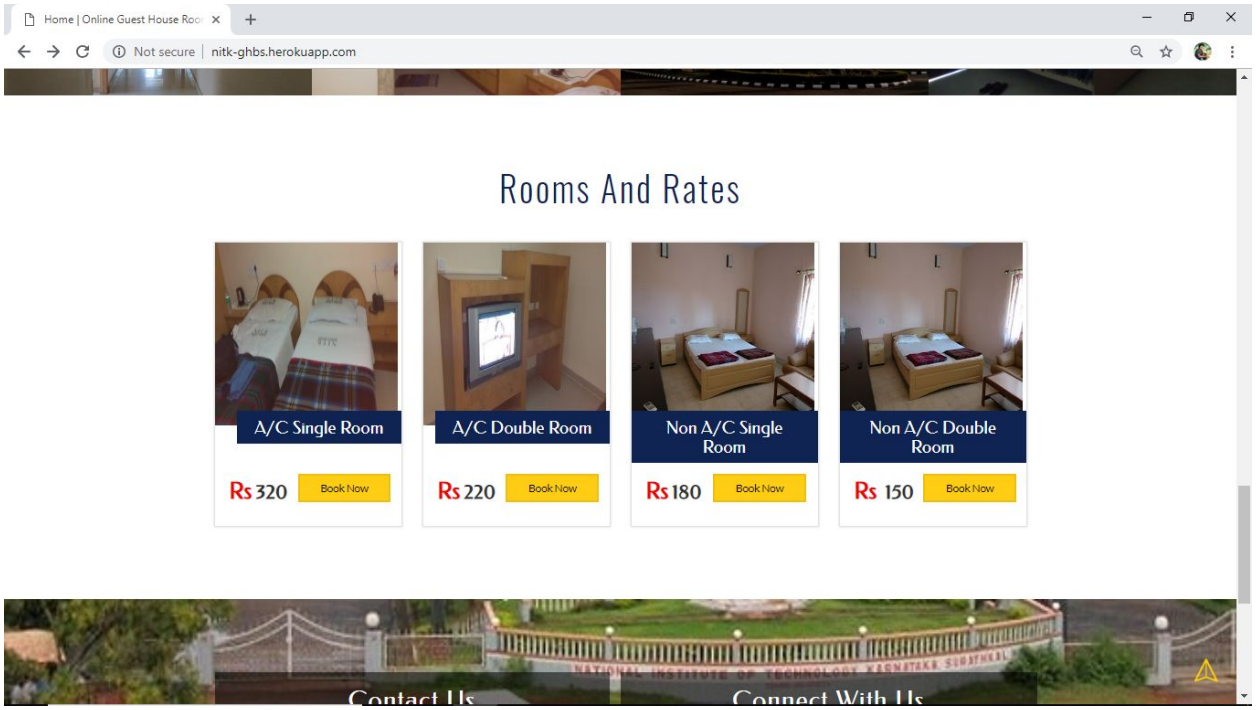
In the learn more section we show the brief view of our guest houses.



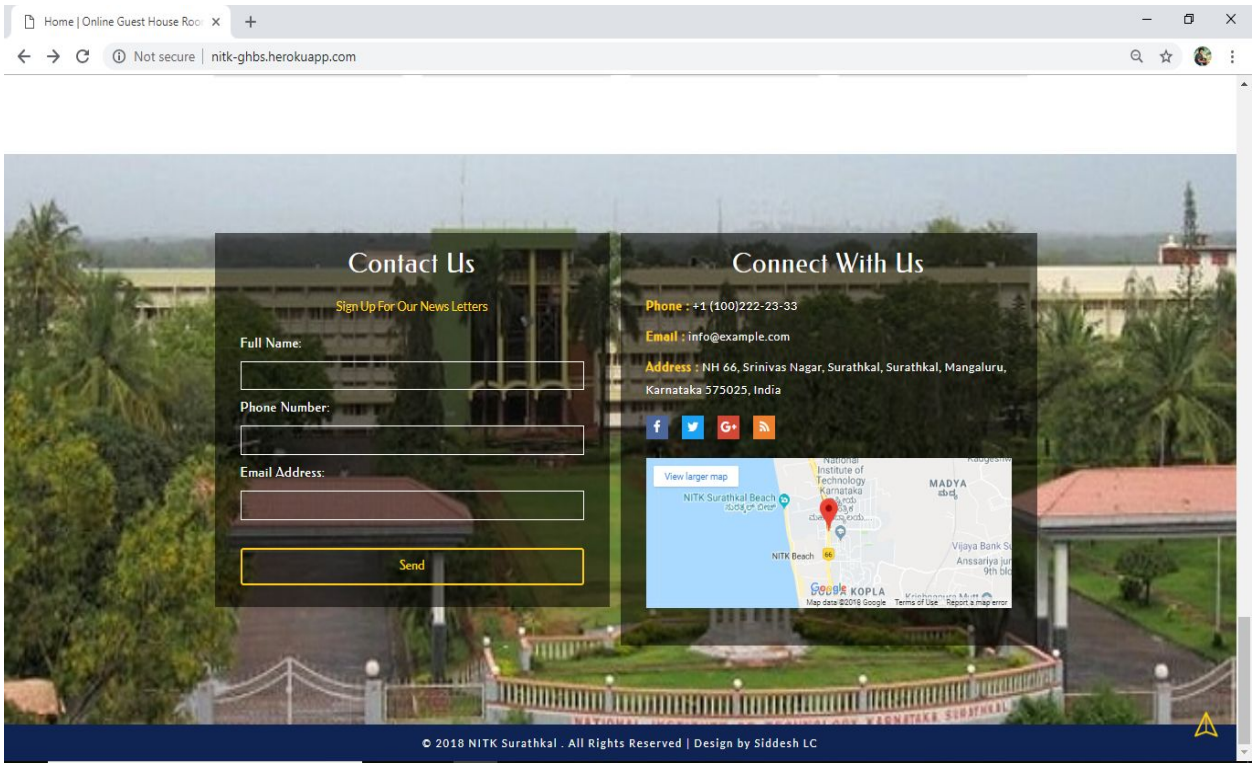
In this section we explain about all the three guest houses along with pictures.



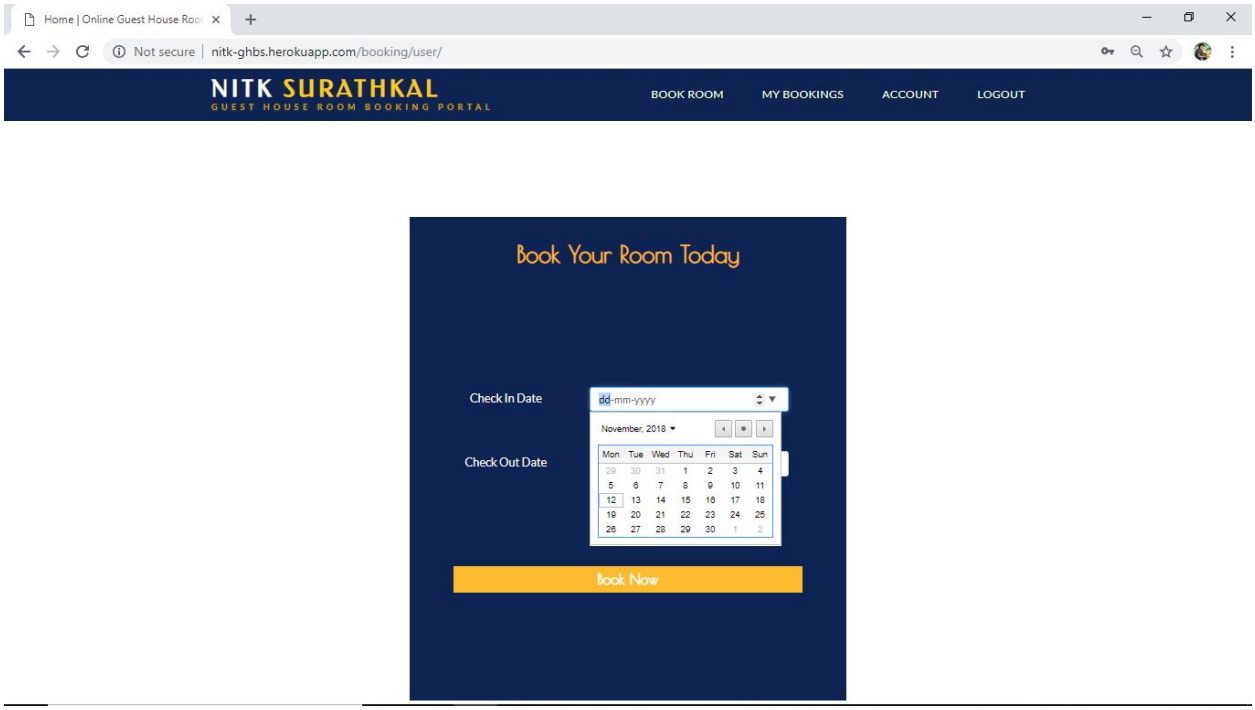
The gallery section displays the pictures of rooms and guest houses of NITK.hjbhjbhb



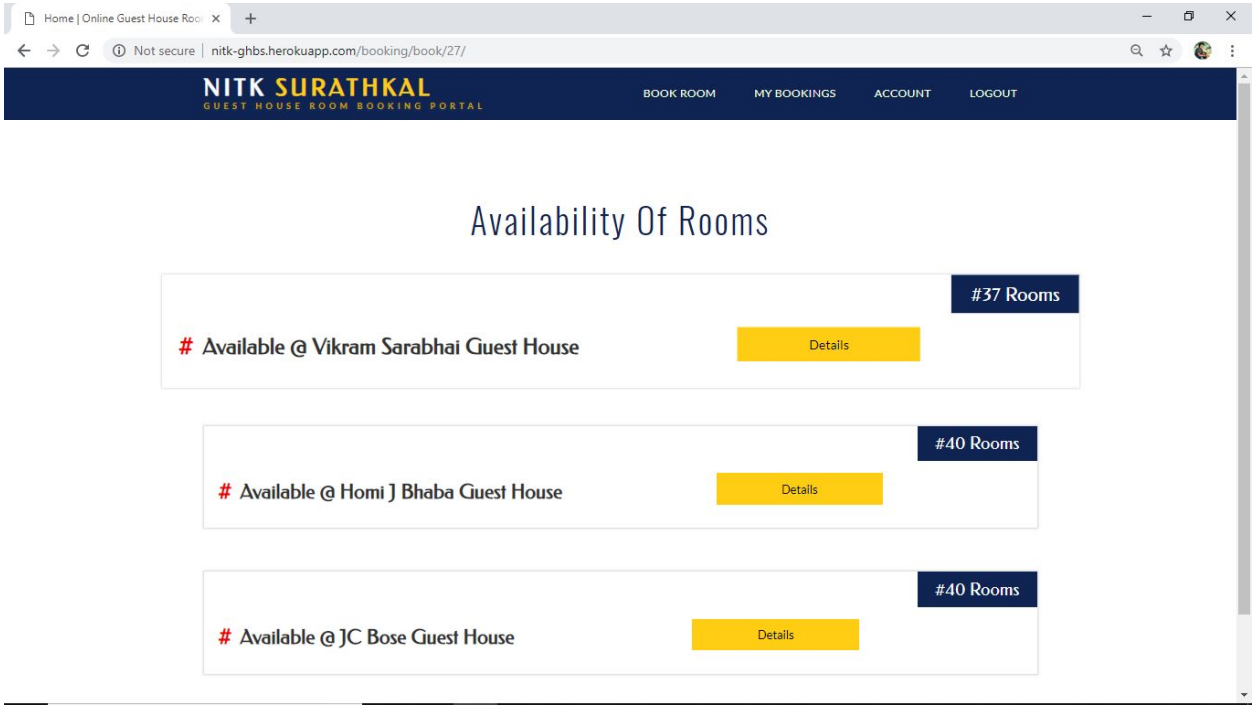
In home page different types of rooms and its prices are shown and an option to book rooms.



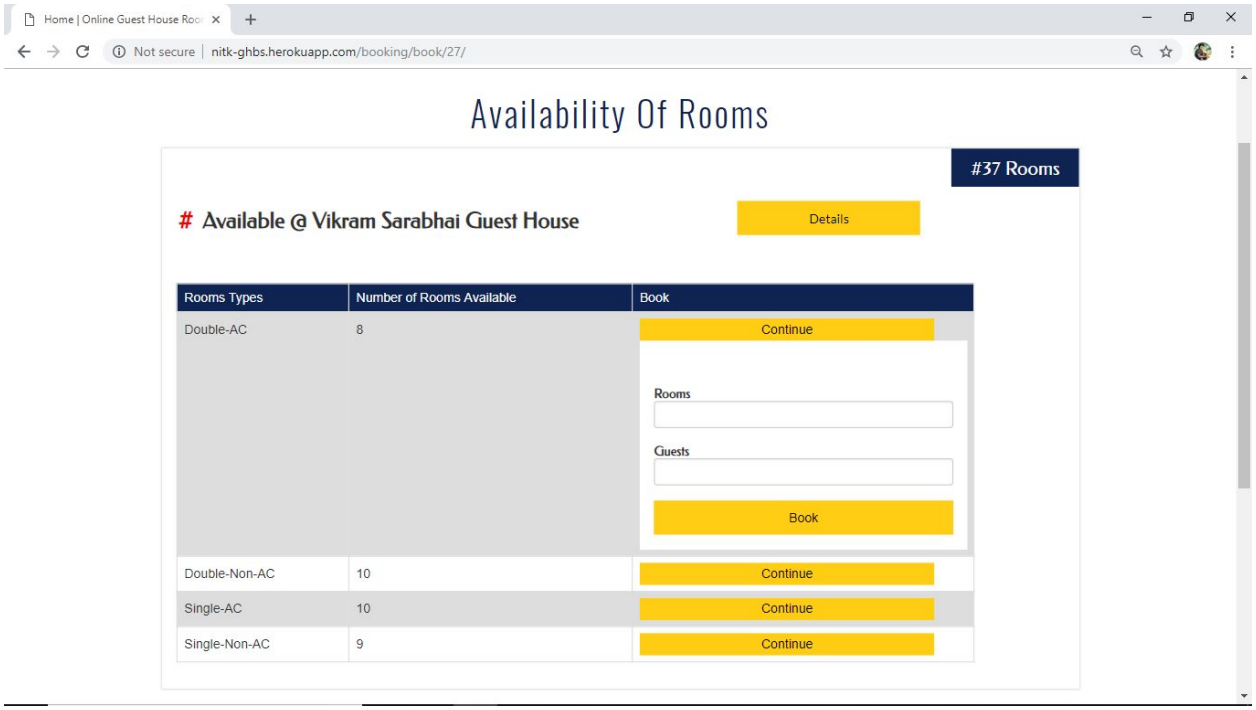
The contact us part allows users to contact the managers for any kind of information regarding booking rooms.Also a map is provided which has location of guest houses.



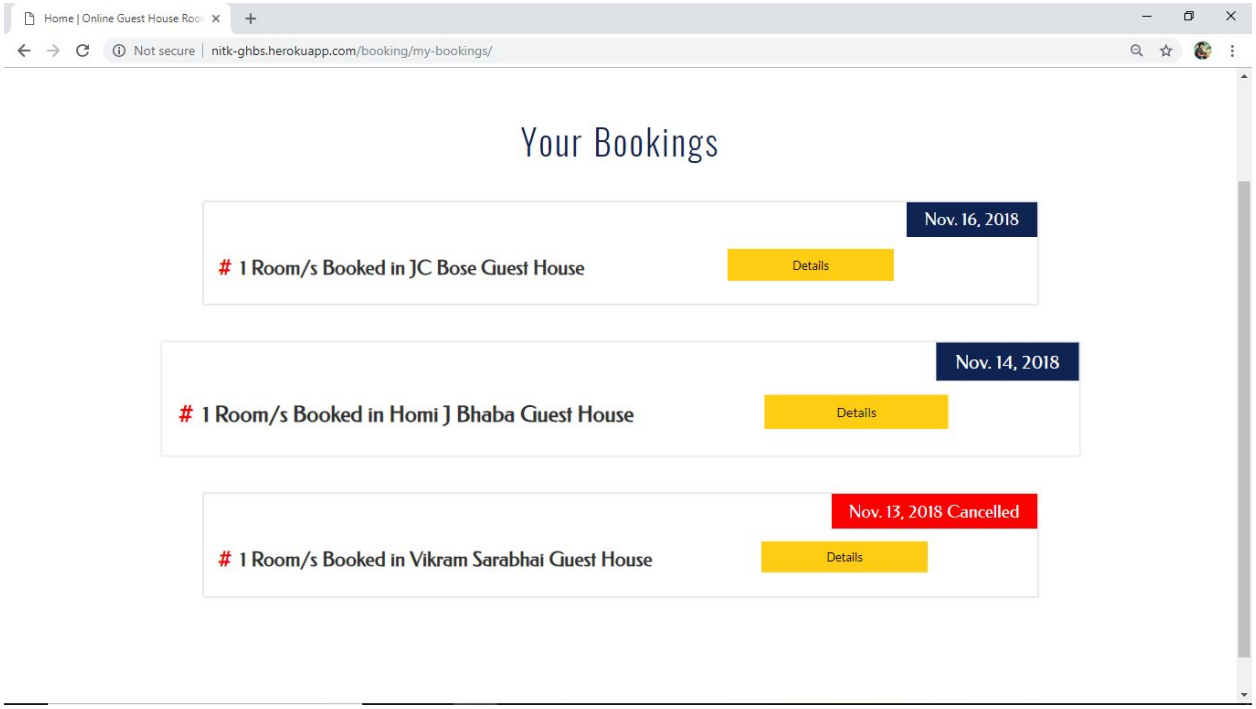
The booking form asks users to enter checkin date and checkout date.



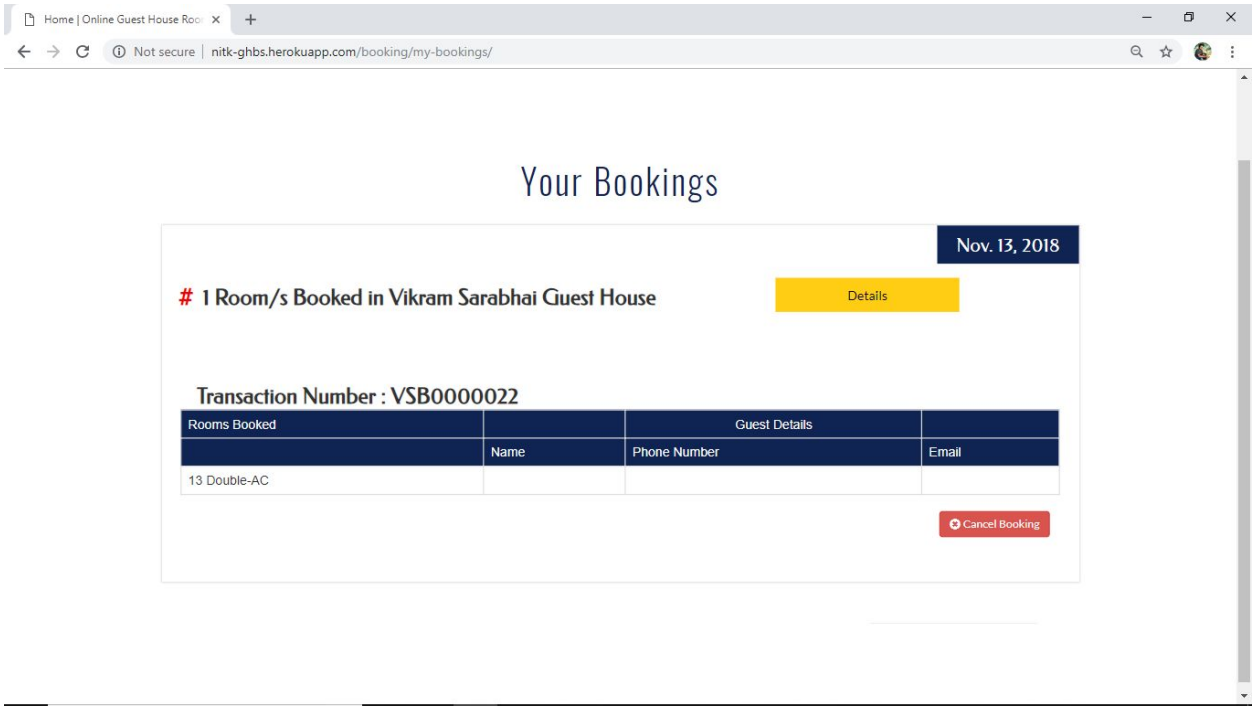
The availability page displays number of rooms available in each of the guest houses



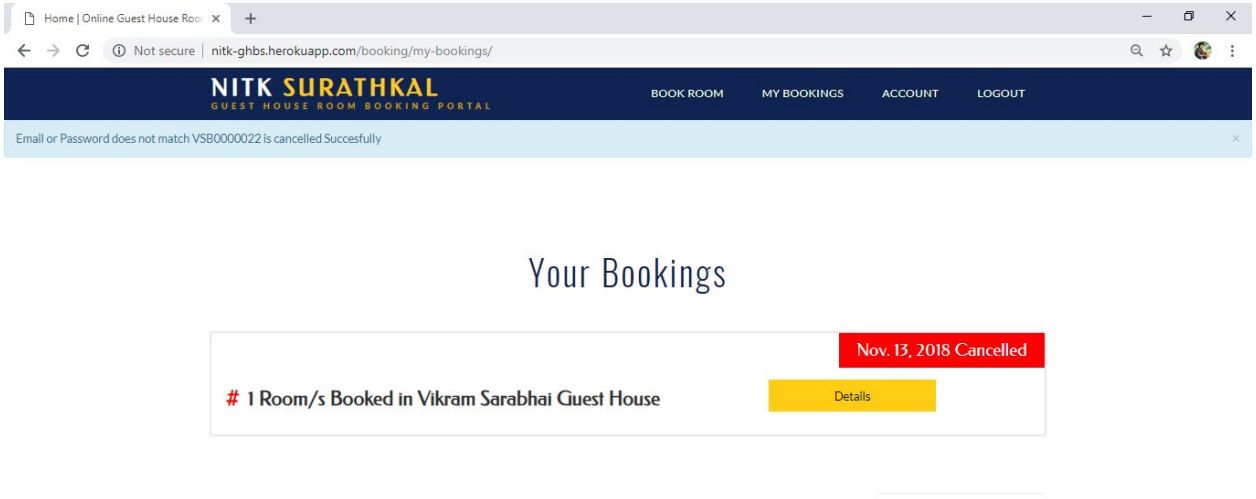
In room availability page users can get information about rooms and can book enter number of rooms required and number of guests.



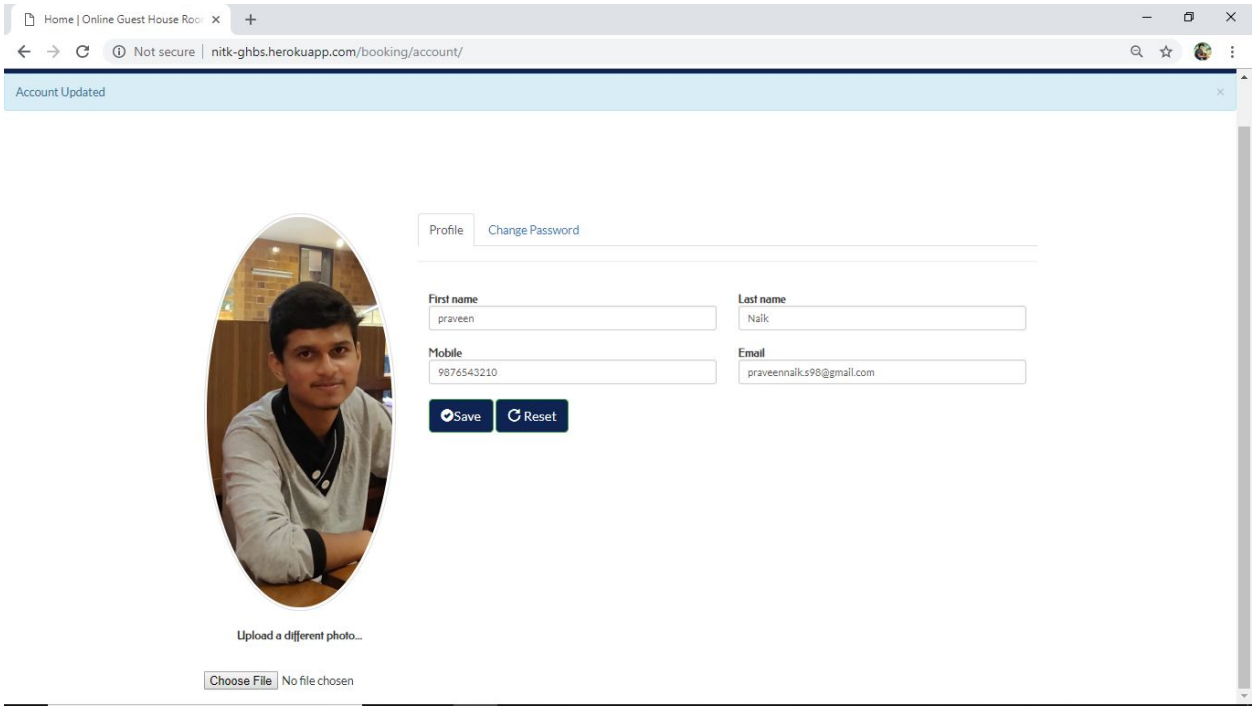
My bookings page will show the bookings done by user using the account.



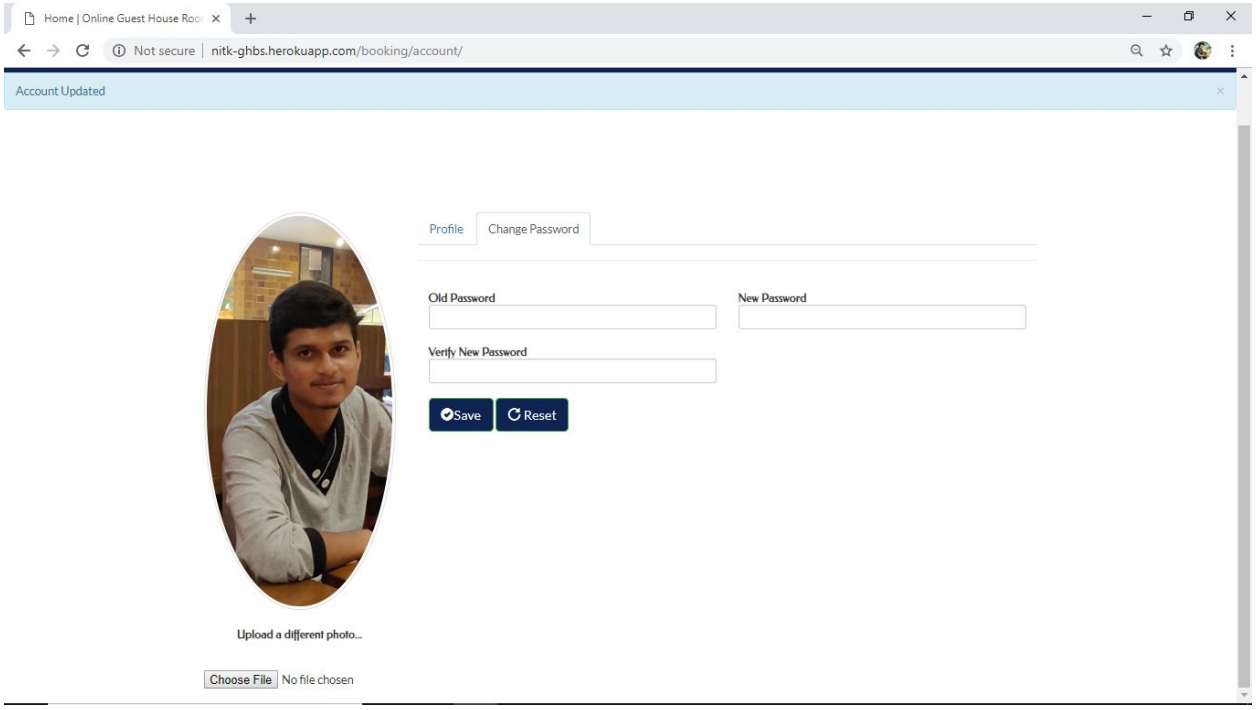
under my bookings user can view description about each of the bookings by clicking details and also can cancel the booking of room before checkin.



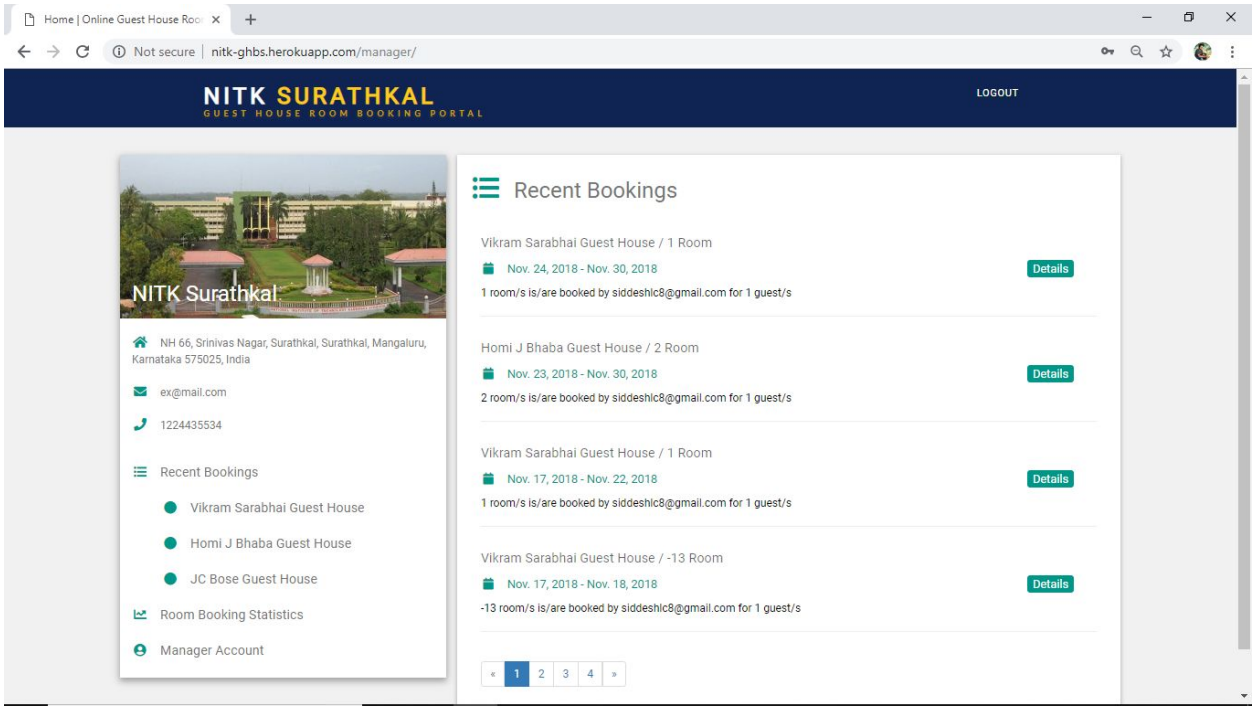
Once a room is cancelled by user, It'll be dispayed as cancelled transaction under my bookings section.



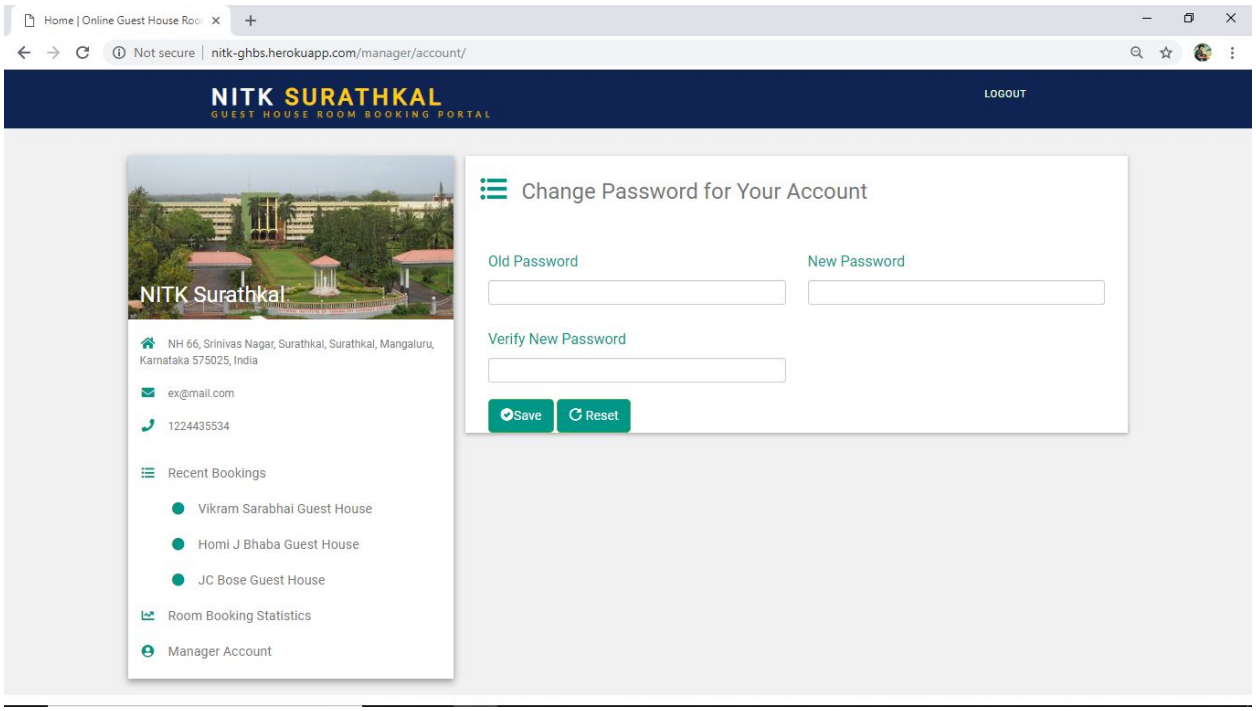
The profile page of the user with his personal information and image which he/she can edit at any point of time.



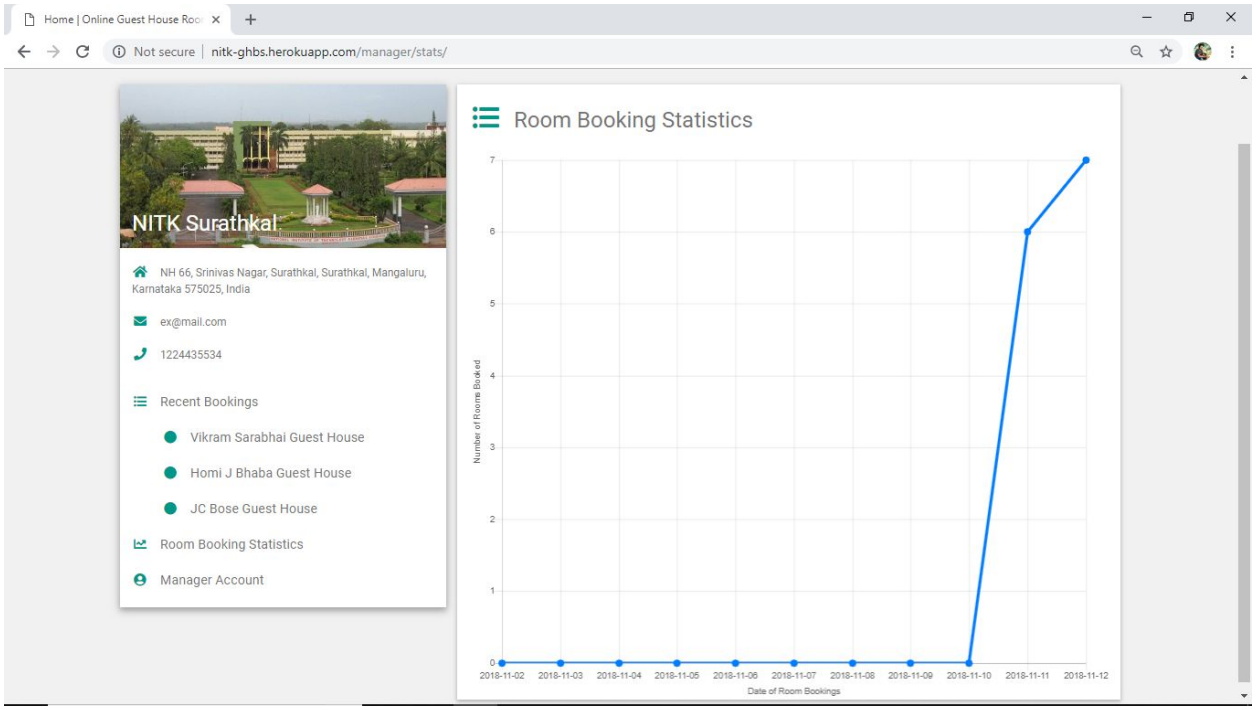
The profile page also allows user to change password if he/she wishes to.



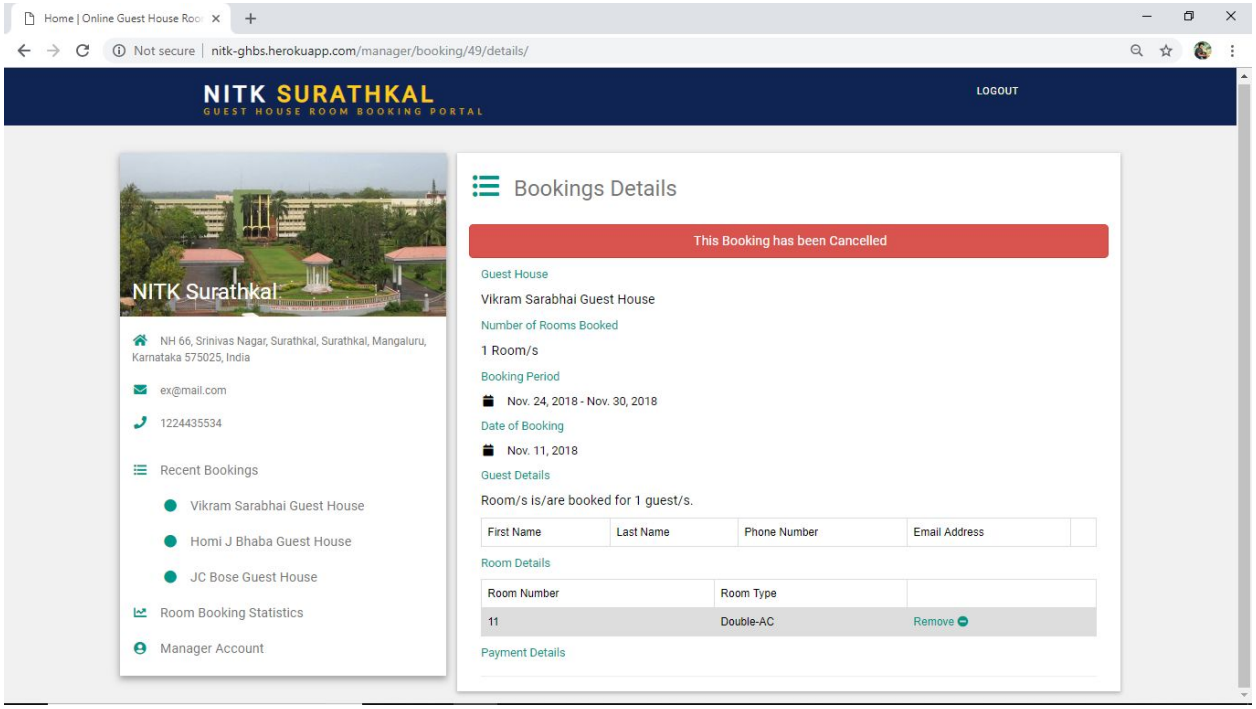
This is the manager page where he can control all the transactions.manager can see the recent bookings in various guest houses seperately.



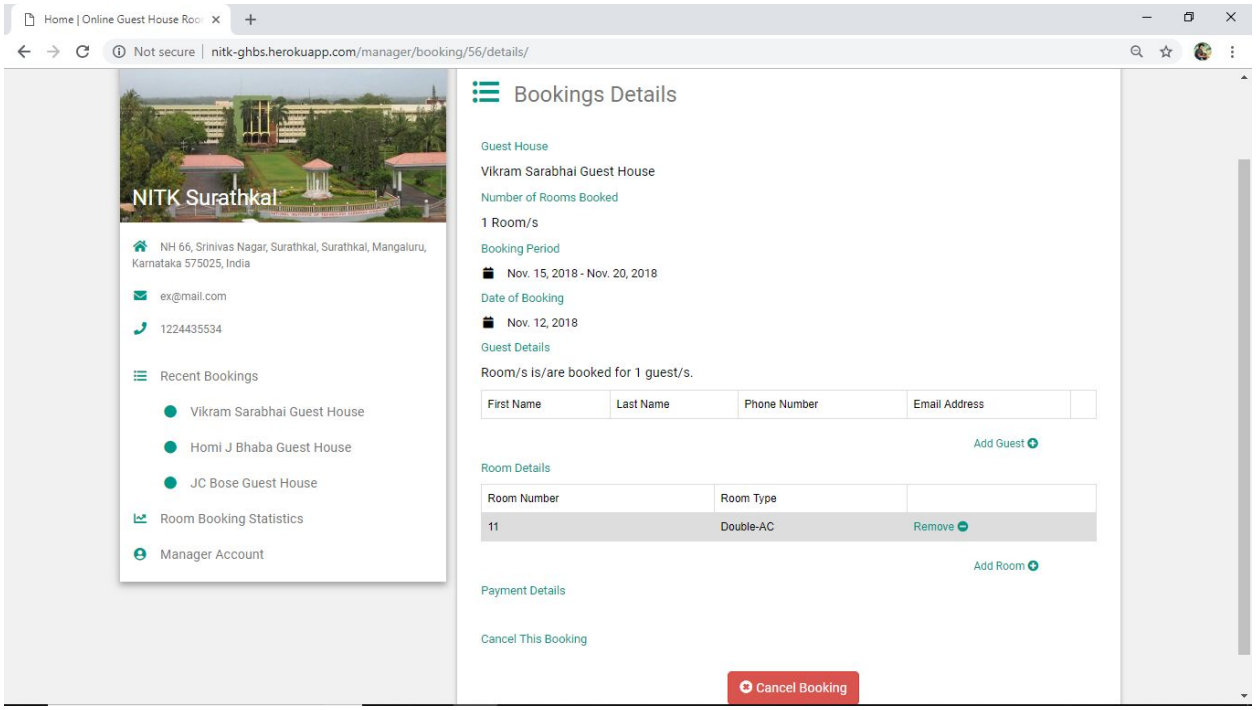
The manager can change password at any point.



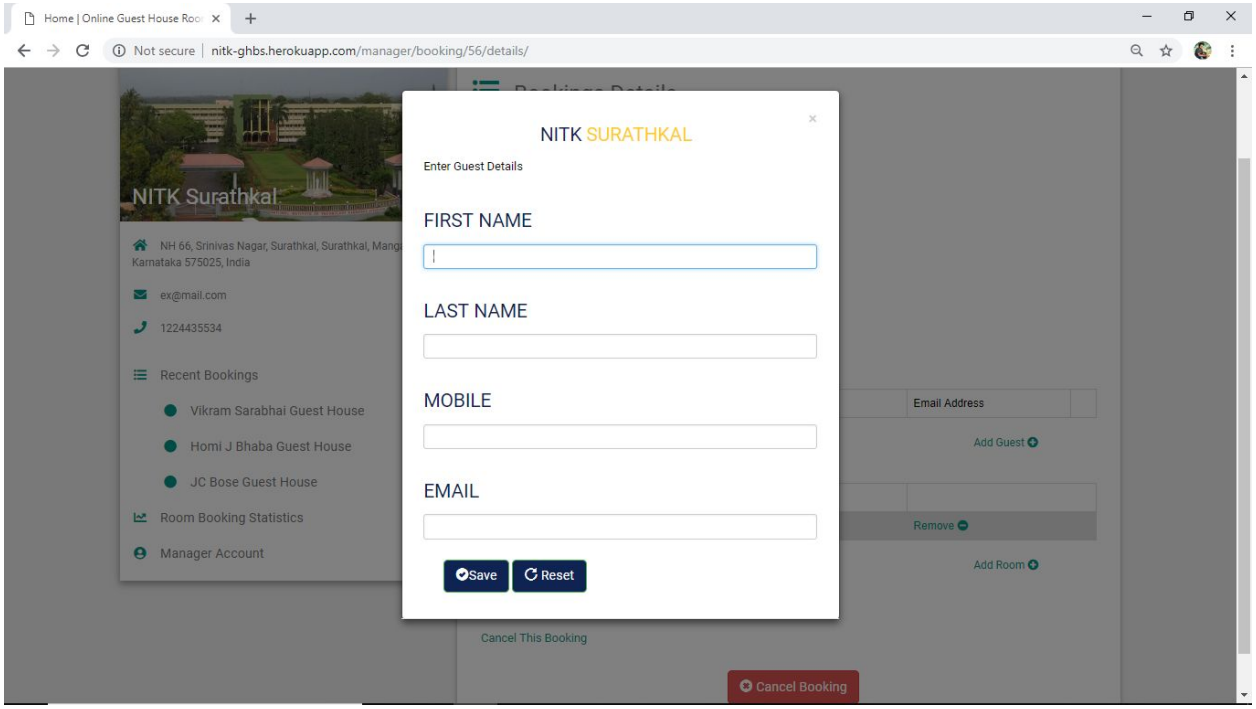
This is the rooms statistics page which shows the graph between rooms booked and date at which rooms are booked. This helps to keep track of rooms.



Manager can see the details of each booking transaction.Add the details of guests and can dynamically change the request for rooms by users.



Here manager can see the details about the guests and can cancel rooms if requested by user. And the details about the guests is entered in below page.



Conclusion

We have made Online guest house booking system-NITK as a web-application developed to provide students of NITK a chance to book rooms online in guest houses and reduce the burden of taking permission from registar and physically visiting guest house to check availability of rooms.As the system makes sure that only users with valid institution mail can only book rooms at the guest house. The tool is made to be simple to understand and to be used by anyone who is familiar with using elementary websites. Also the system helps managers of guest house to handle the rooms well as the statistics about availability of rooms and info about the guest present in rooms. This project is a contribution to our institution which digitalises the guest house booking system completely.