A Project Report On

Hotel Booking

Submitted in partial fulfillment of the requirement for the award of the degree

MASTER OF COMPUTER APPLICATIONS

From

Gujarat Technological University

Academic Year 2020 – 2021

Janak Siddhpura (195223693077)

Shyam Thakar (195223693076)

Internal Guide

Pr. Jaypalsinh Gohil

Marwadi Education Foundation's Group of Institutions (MEFGI)

Rajkot-Morbi Road, At & PO : Gauridad, Rajkot 360 003. Gujarat. India.

Faculty of Computer Applications (FCA)

Certificate

This is to certify that the project work entitled

Hotel Booking

Submitted in partial fulfillment of the requirement for

the award of the degree of

Master of Computer Applications

Of the

Gujarat Technological University

is a result of the bonafide work carried out by

Janak Siddhpura(195223693077)

Shyam Thakar(195223693076)

During the academic year 2020-2021

Faculty Guide **H.O.D**

External Viva

Name of the Examiners **Signature with Date**

DECLARATION

We hereby declare that this project work entitled **Hotel Booking** is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place :

Date :

Janak Siddhpura (195223693077) **Signature: _____**

Shyam Thakar (195223693076) **Signature: _____**

ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement or success one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our faculty guide **Dr. Jaypalsinh Gohil,** the person who makes us follow the right steps during project work. We express my deep sense of gratitude to his guidance, suggestions and expertise at every stage. Apart from that his valuable and expertise suggestion during documentation of our report indeed helped us a lot.

Thanks to our friends who have been a source of inspiration and motivation that helped us during our project work. We are heartily thankful to our Head **Dr. Sunil Bajeja** and Dean **Dr. R. Sridaran** for providing us an opportunity to work over this report and for their endless and great support. We would like to extend our gratitude towards all who directly or indirectly supported and helped us to fulfil our project work.

Janak Siddhpura (195223693077) **Signature: _____**

Shyam Thakar (195223693076) **Signature: _____**

Figure Index

TABLE INDEX

1.Introductin

Online Hotel Booking System is a website to provide the Customer facility to book room online and to gather information about the room & facilities. Customer needs to register at the site to room at hotel. After book the room, the customers show his booked room in site. Thereafter he is redirected to the payment gateway for making a transaction. User can also view his booking history. The project "Online Hotel Booking System" is a system based on accessing the internet to book for rooms in a hotel. The purpose of this study is to develop and implement an online hotel reservation system for hotels, that will replace the manual method of booking for hotel rooms. The previous system for booking rooms were faced with so many problems like, delay in processing the customer booking or paying for rooms that is below or beyond his standard, causes difficulty for emergency booking.

1.1 Existing System

- The existing system is manual system. Need to convert into automated system.
- Risk of mismanagement of data.
- Less security.
- No proper coordination between different applications & users.
- Fewer users-friendly.
- Accuracy not guaranteed.

1.2 Need For The New System

Hotel Booking System where you can view Hotels Rooms. Hotel Booking System has a facility to view the Rooms as the User want. And on the basis of the Rooms it helps to find the best rooms and also can book the Rooms.

We provide various categories of the Rooms like Deluxe, Non-AC, AC Rooms, Single Bedrooms, and Tween Bedrooms.

The purpose of the project is user can easily book their room by its choice on basis of the room description, Images, Prices, and also basis of room type that they want.

1.3. Objective of the New System

To remove all the disadvantages of conventional methods, a system is proposed. which is an online Booking .The purpose of online Booking is to save time, save money. Through online shopping one can save his valuable time. One can watch and select things he wants to book. Through online Booking we can save our money because prices are less than market prices and we receive our bought things at our home. No need to go anywhere and do Book. We can get different varieties of things online and we can choose which one we want. The new system provide user online facility to client can book online camera instated of go and book.

1.4. Problem Definition

The manual method of booking for hotel rooms problems Some of these are: customers having little or no information about the hotels within their vicinity. A guest checking into a hotel room that is either too expensive or too unsuitable for his/her personality. The foul play that sometimes occurs when information about the guest that checked into a hotel are not officially documented by the receptionist etc. All these problems and more would definitely make a hotel experience a down turn in business

The main purpose of this work is therefore to develop a web application program that would avoid all those problems encountered in the manual hotel booking system, so that customers can easily go online with their mobile phones, tablets or laptops in order to browse the relevant information they need about the hotels within their locality so that they can book for the appropriate suite that is within their budget.

1.5. Core Components

1. Login:

To, make any products available in the viewer's cart or order viewer must be logged in the website. Viewers have to fill some details about them and create an account for the further processes.

2. Admin:

In the Admin Module Admin is the User which has Add the Room Remove the Room from the Website.

3. User:

This module is for the users who do not have their account. Here user is allowed to create an account to login. The account creation is done by filling the registration form with user details such as name, password etc.

4. Home:

This page contains all links and updates about the website. Viewer can navigate from one page to another from this page.

5. Room:

This module has information regarding the room such as its name, model, room type, no of child ,adult or bad, price information, its features etc. The Admin has the authority to Add, Delete, Update etc.

6. Book:

This Module User can select any of the Room and Book. User also removes from the booked room.

7. AboutUs:

Viewer can see the details about "Online Hotel Booking Site".

8. Contact Us:

Viewer can contact admin/owner directly through the number or email.

1.6. Project Profile

Project Details	
Project Title	**Hotel Booking System**
Submitted To	**Marwadi Education Foundation ,Rajkot**
Front End	**HTML,CSS**
Back End	**MySql** **Python 3.2.7(HigherVersion)**
Documentation Tool	**Microsoft Word**
IDE	**Visual Studio Code**
Main Pages	**Registration , login , Home page,** **Book room page**
Web Browser	**Google Chrome, Mozilla Firefox**
Guided by	**Pr. Jaypalsinh Gohil**
Developed by	**JanakSiddhpura** **Shyam Thakar**

1.7. Advantages & Limitations of System:

- **Advantages:**

  - An online booking system will reduce time.
  - Online booking means faster payments.
  - You can choose your favourite rooms.
  - There is no paper slip so no chance of losing it anyhow.
  - Easily check the availability of room & Facility

- **Disadvantages**:

  - Require internet connection
  - You need to be ready for an influx of new customers.
  - If the website is down you won't be able to book it online

2. Requirement Determination & Analysis

2.1. Requirement Determination

Hardware Requirement:

512 MB RAM
Intel i3 or Higher processor
HDD 90 GB or higher

Software Requirement:

Microsoft Windows 10,9,8,7,Vista or Higher OS
Internet explorer/ Mozilla Firefox / Google Chrome
Python 3.7
Django Framework
xampp

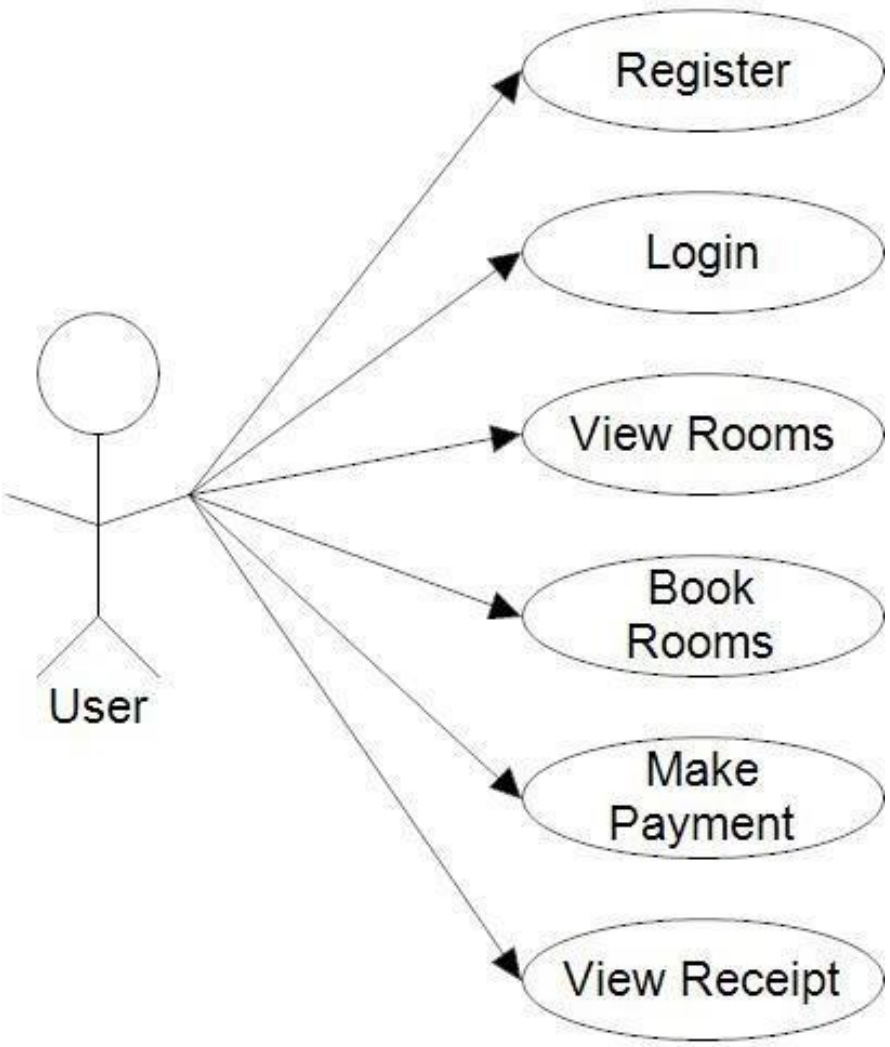
2.2. Targeted Users:

Hotel Booking System is mainly used by the most of the people who are like to visit different places And like to do online booking for their hotel room and also view their room description and features Online

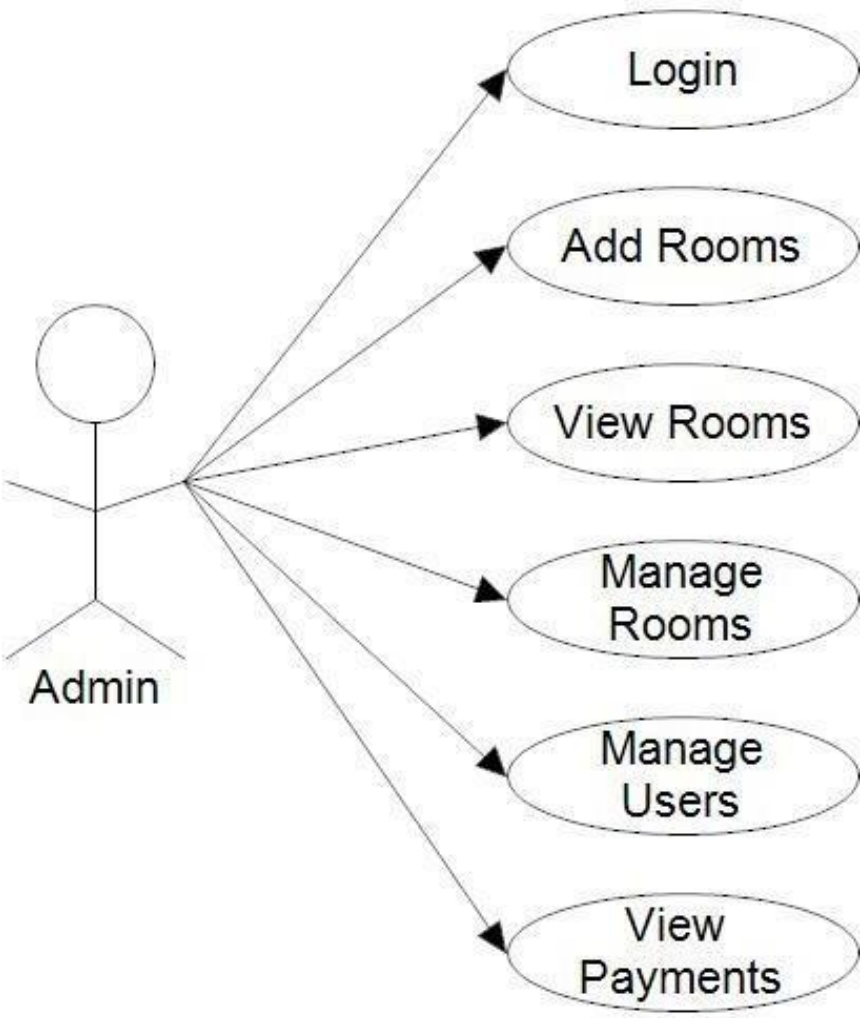
3. System Design

3.1. Use Case Diagram :

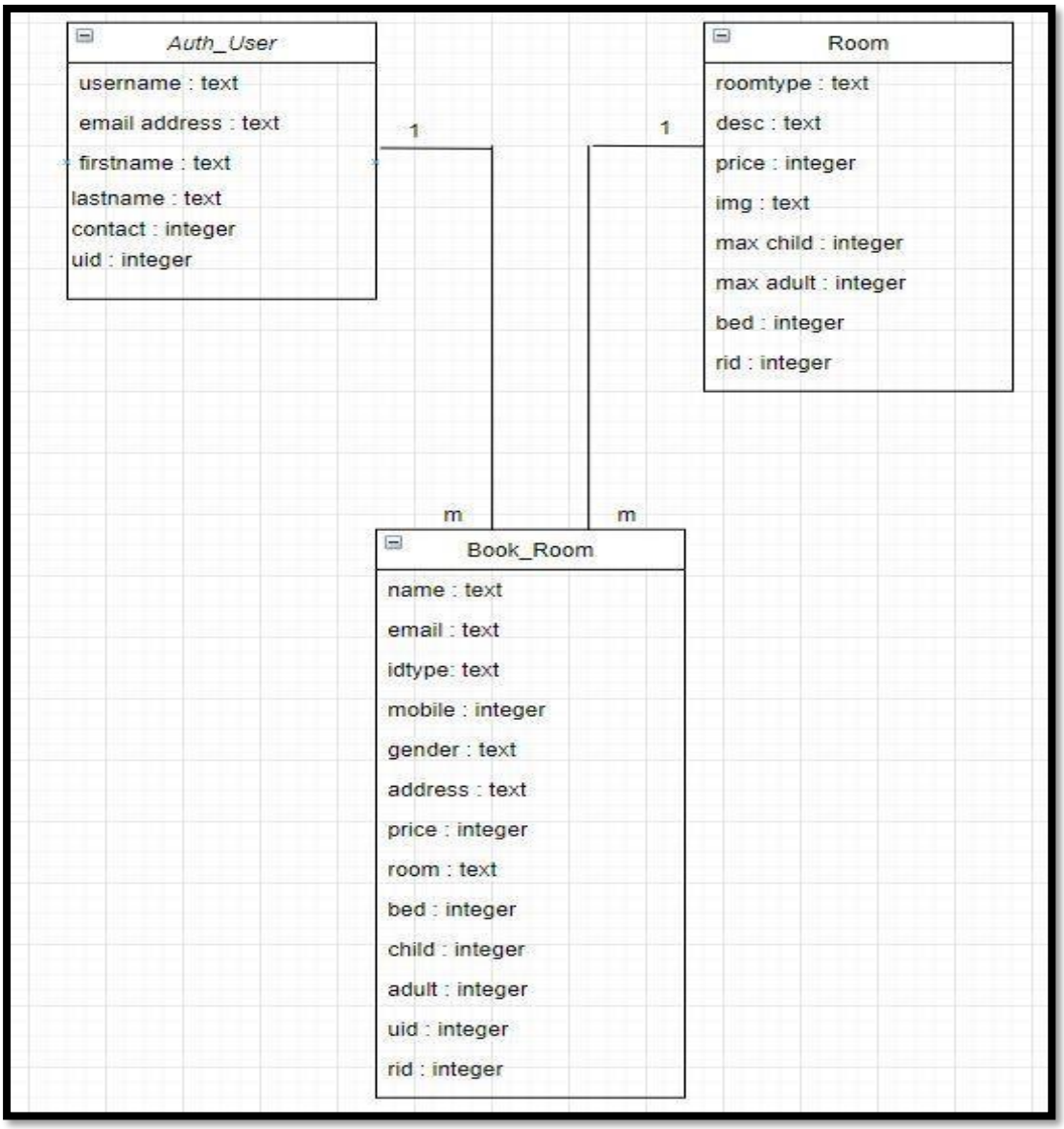
3.1.1.User side:



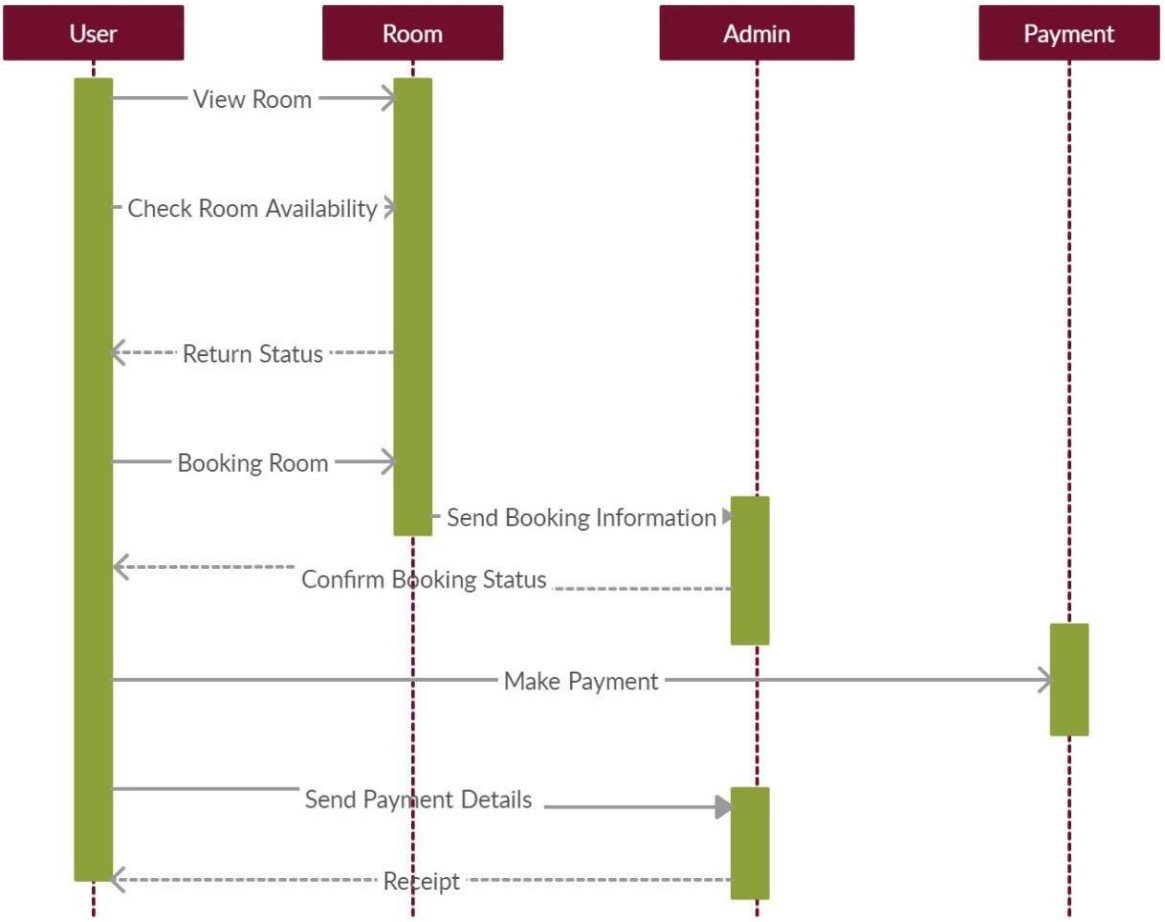
3.1.2 Admin Side:



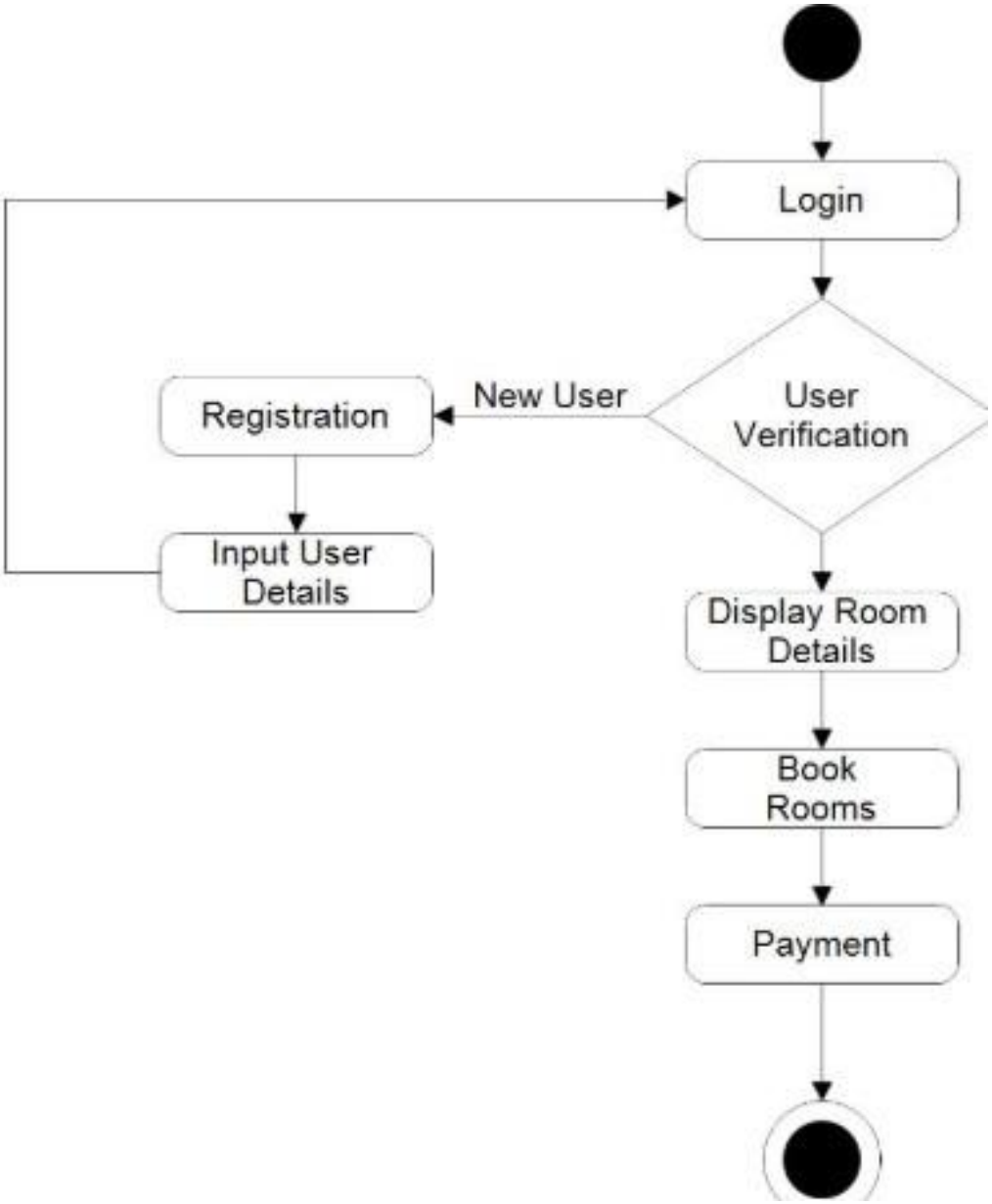
3.2. Class Diagram:



3.3. Interaction Diagram :



3.4. Activity Diagram:



3.7. Data Dictionary

3.7.1 Auth_User:

Field	Type	size
Id	Int	11
Password	Varchar	128
Username	Varchar	150
First_name	Varchar	150
Last_name	Varchar	150
Email	Varchar	254

3.7.2 Room:

Field	Type	size
roomtype	Varchar	11
desc	Longtext	
price	Int	11
date	Date/time	8
img	Varchar	100
max child	Int	11
max adult	Int	11
Bad	int	11

3.7.3 Book_Room:

Field	Type	size
name	Varchar	20
email	Varchar	100
Idtype	Varchar	20
mobile	Number	20
gender	Varchar	20
address	Varchar	20
check-in	Date/time	
check-out	Date/time	
price	Int	20
room	Varchar	100
bad	Int	20
child	Int	20
adult	Int	20

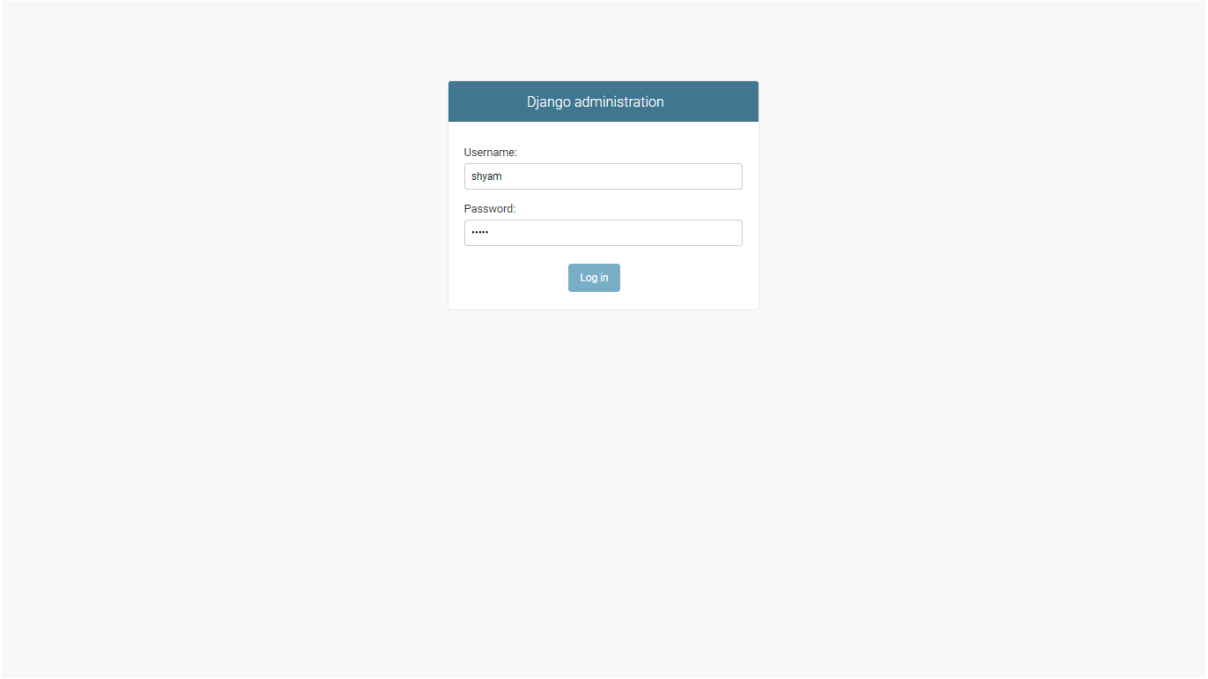
3.7.4 Contact

Field	Type	size
Message	Longtext	
Name	Varchar	20
Email	Varchar	100
subject	Varchar	100

4.Development

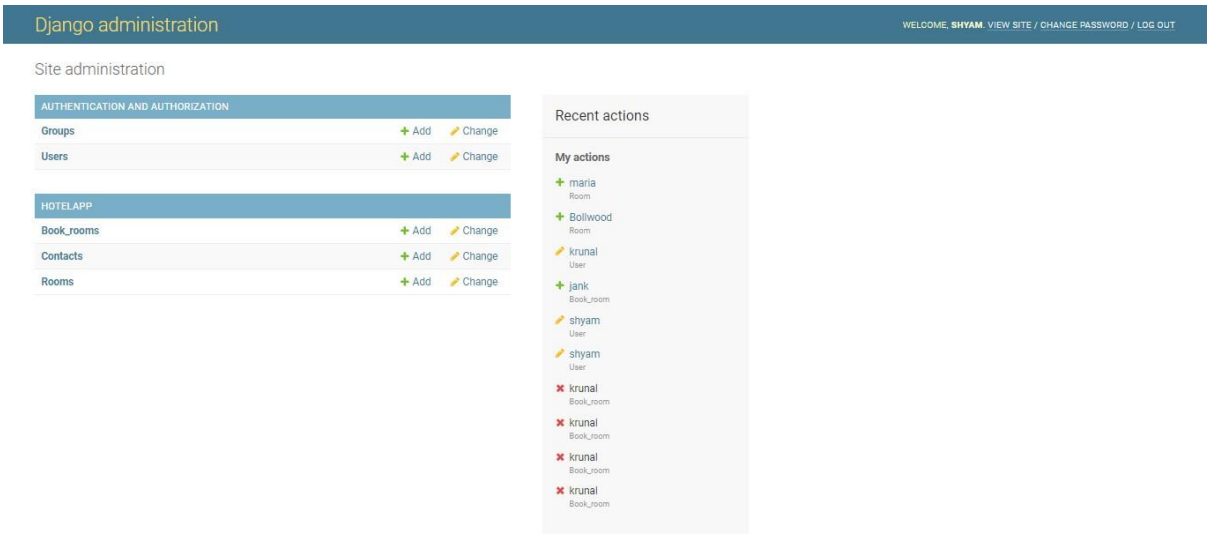
4.1 coding standard:

4.1.1 Admin Login Page:

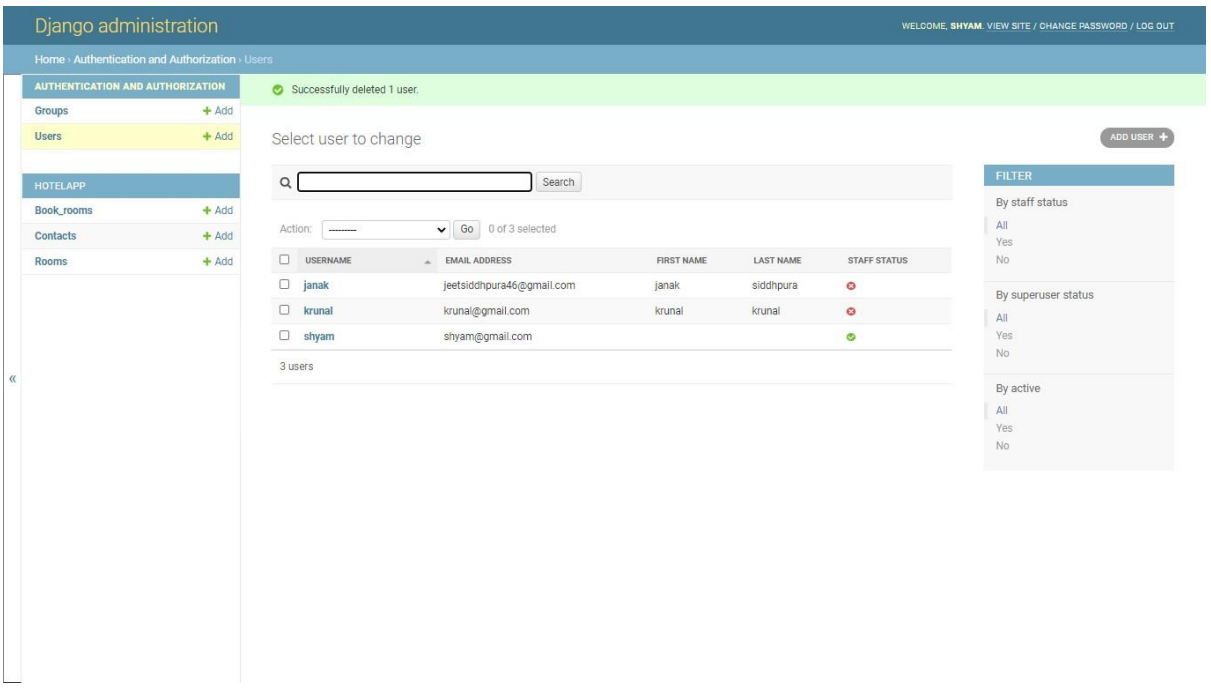


Python Manage.py MakeMigration
Python Manage.py Migrate
PythonManage.py createUser

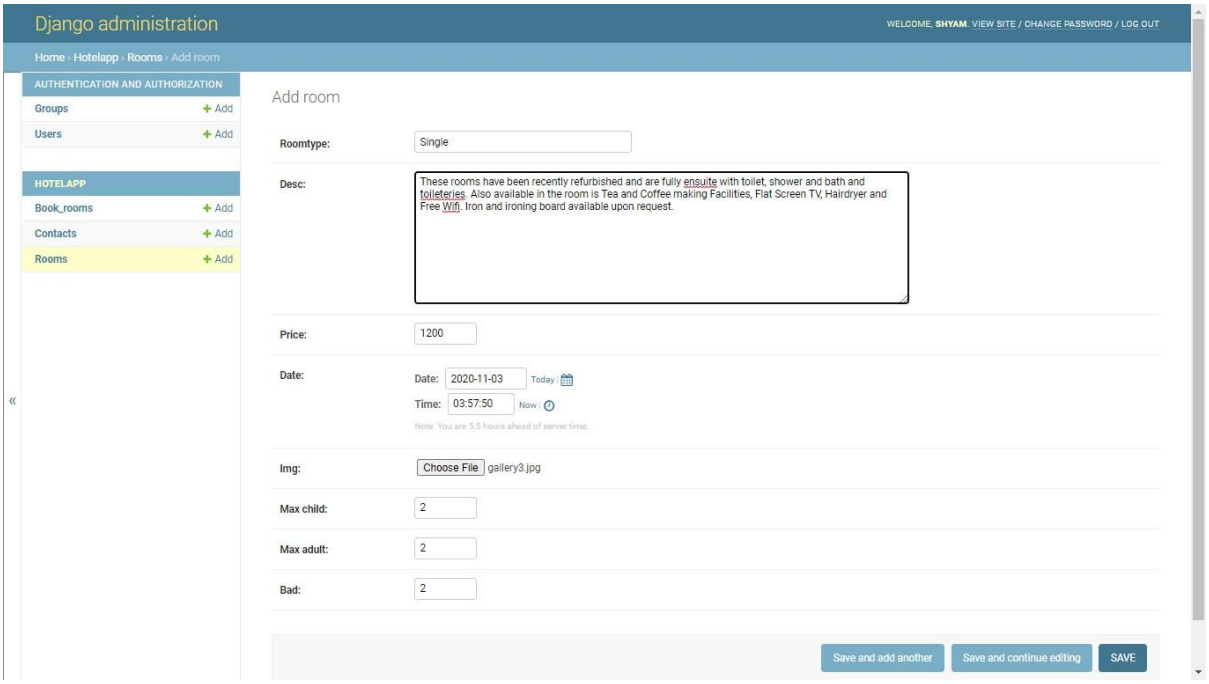
4.1.2 Admin Home Page:



4.1.2 Admin User Page:



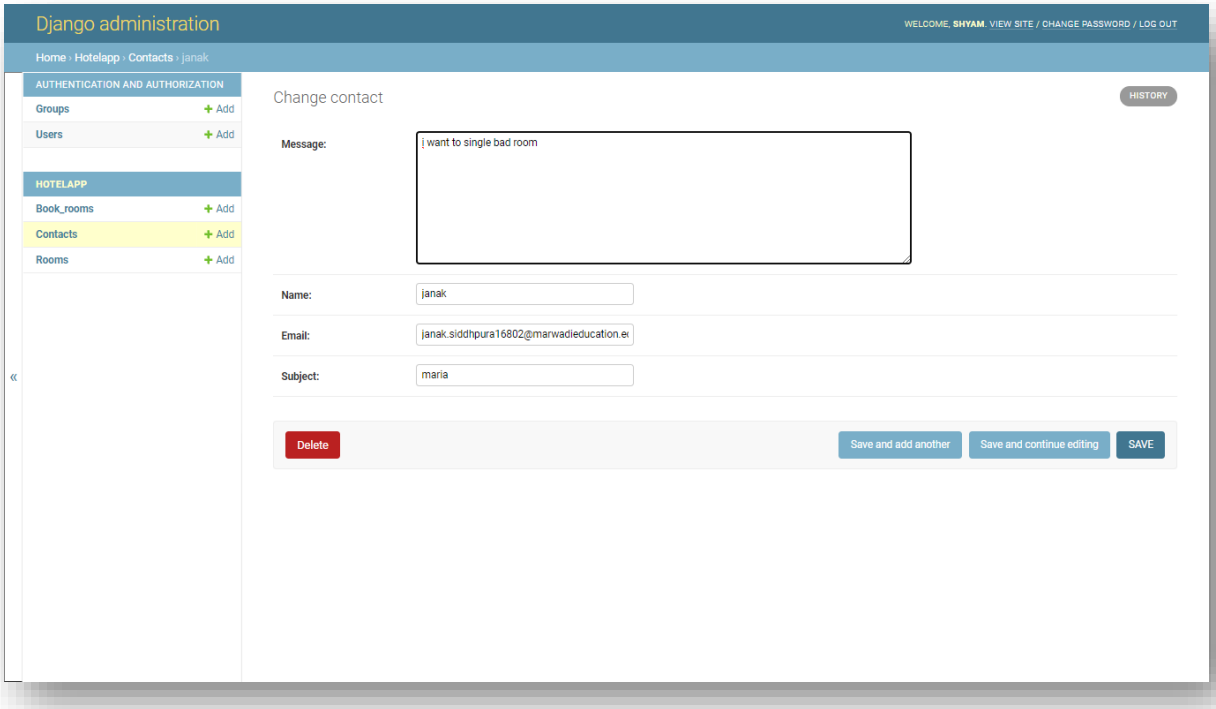
4.1.3 Admin Room Page:



Models.py

```
class Room(models.Model):
    roomtype=models.CharField(max_length=20)
    desc=models.TextField() price=models.IntegerField()
    date=models.DateTimeField()
    img=models.ImageField(upload_to='pics')
    max_child=models.IntegerField()
    max_adult=models.IntegerField()
    bad=models.IntegerField()

    def___str___(self): return
        self.roomtype
```

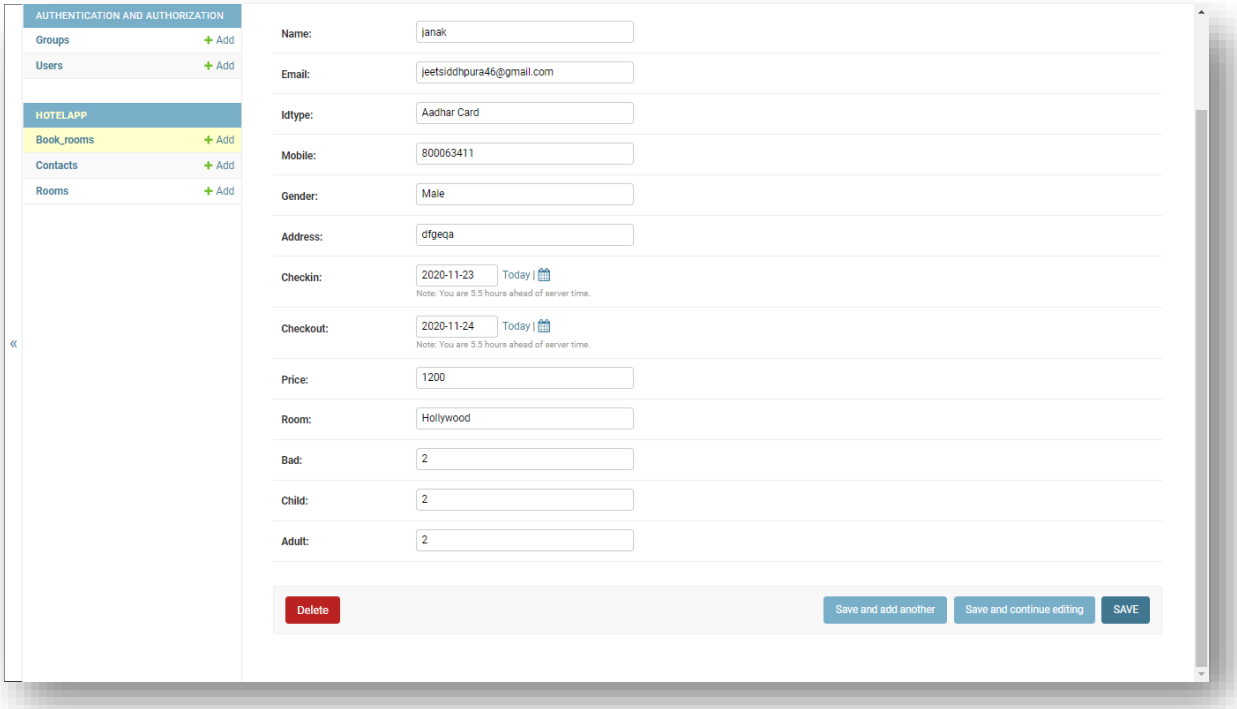
4.1.4 Admin contact Page:



Models.py

```
class Contact(models.Model):
    message=models.TextField()
    name=models.CharField(max_length=20) email =
    models.CharField(max_length=100)
    subject=models.CharField(max_length=100)

    def___str___(self):
        return self.name
```

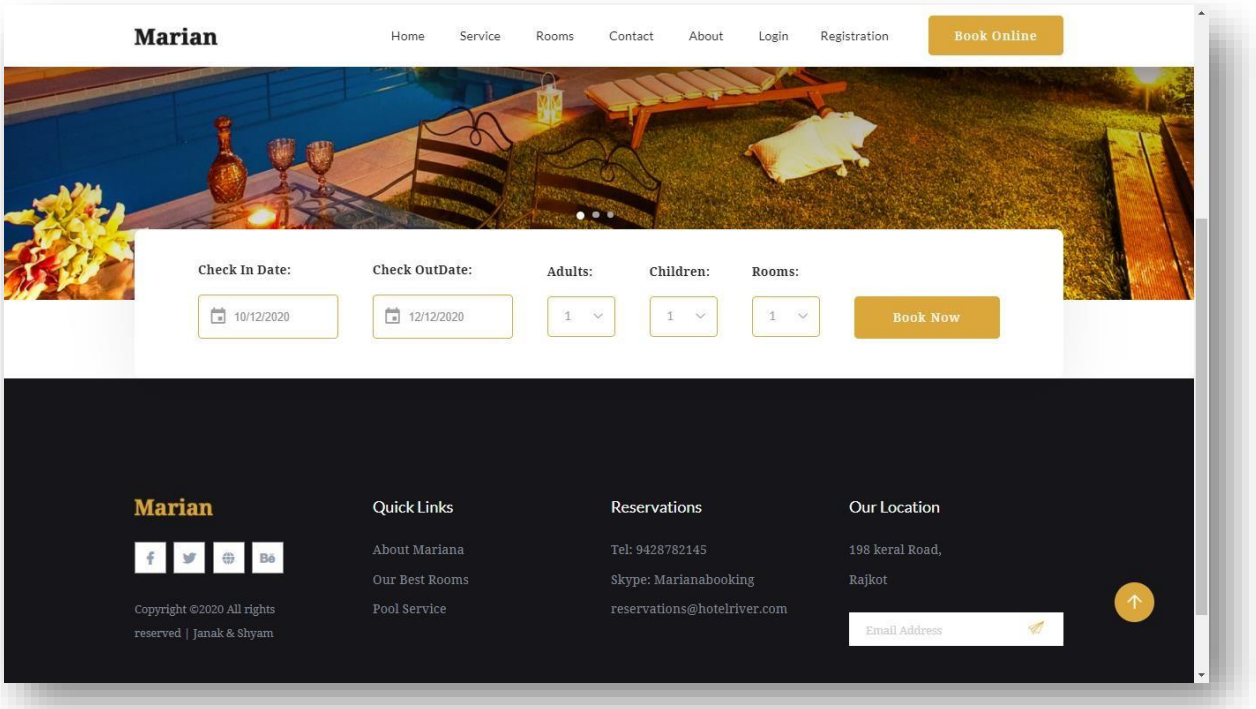
4.1.5 Admin Book_Room Page:



Model.py

```
class Book_room(models.Model):
    name=models.CharField(max_length=20)
    email =
    models.CharField(max_length=100)
    idtype=models.CharField(max_length=20)
    mobile=models.CharField(max_length=10)
    gender=models.CharField(max_length=20)
    address=models.CharField(max_length=20)
    checkin=models.DateField()
    checkout=models.DateField()
    price=models.CharField(max_length=10)
    room=models.CharField(max_length=10)
    bad=models.CharField(max_length=10)
    child=models.CharField(max_length=10)
    adult=models.CharField(max_length=10)

    def_str_(self):
        return self.name
```

4.1.6 User Home Page:



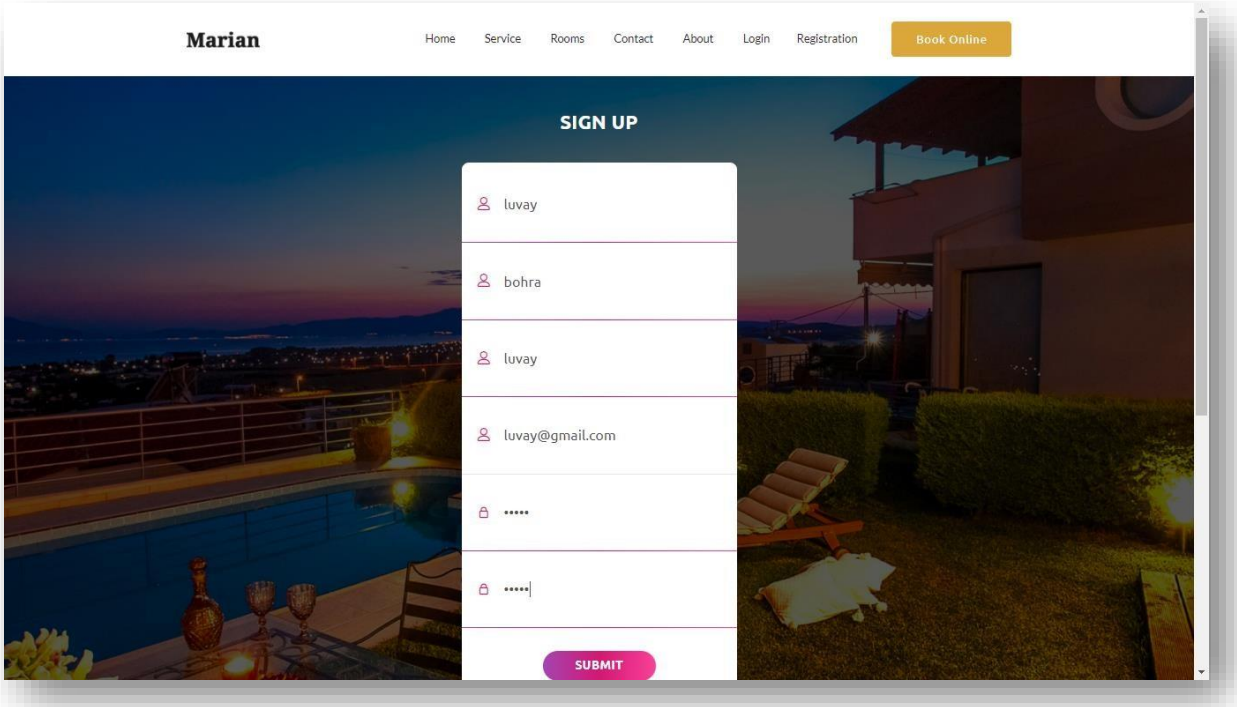
View.py

```
from django.shortcuts import render,redirect
from hotelapp.models import Room,Contact,Book_room
from django.contrib import messages
from django.contrib.auth.models import User,auth
from django.contrib.auth.decorators import login_required #
Create your views here.

def index(request):
    return render(request,'hotelapp/index.html')
```

urls.py

```
from django.urls import path
from . import views

urlpatterns = [
  path('',views.index,name='index'),
```

4.1.7 User Signup Page:



View.py

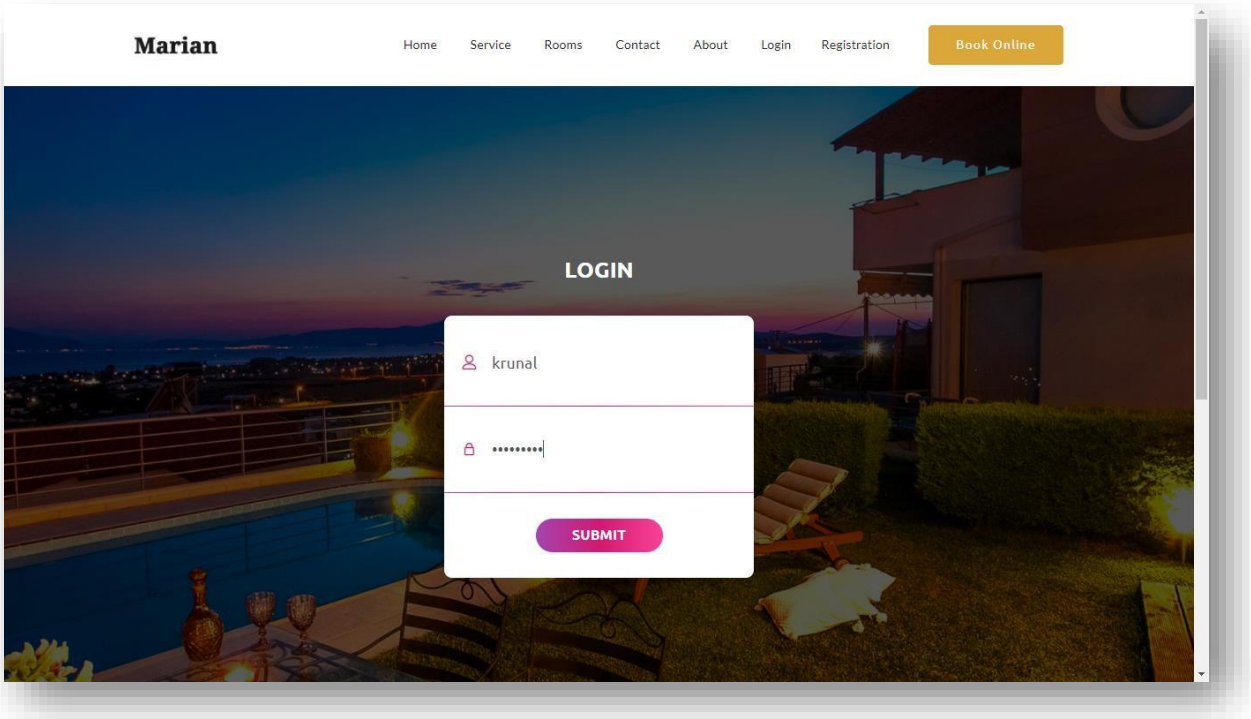
```
def reg(request):
   if request.method=='POST':
      first_name=request.POST['first_name']
      last_name=request.POST['last_name']
      username=request.POST['username']
      email=request.POST['email']
      password1=request.POST['password1']
      password2=request.POST['password2']
      if password1==password2:
         if User.objects.filter(username=username).exists():
            return render(request,'hotelapp/reg.html',{'error':'Username already taken'})
         elif User.objects.filter(email=email).exists():
            return render(request,'hotelapp/reg.html',{'error':'Email Already taken'})

         else:
            user=User.objects.create_user(username=username,password=password1,email=e
mail,first_name=first_name,last_name=last_name)
            user.save()
            print("user created")
            return render(request,'hotelapp/login.html')
      else:
         return render(request,'hotelapp/reg.html',{'error':'Password does not match'})
```

```
else:
      return render(request,'hotelapp/reg.html')
```

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
path('reg/',views.reg,name='reg'),
```

4.1.8 User Login Page:



View.py

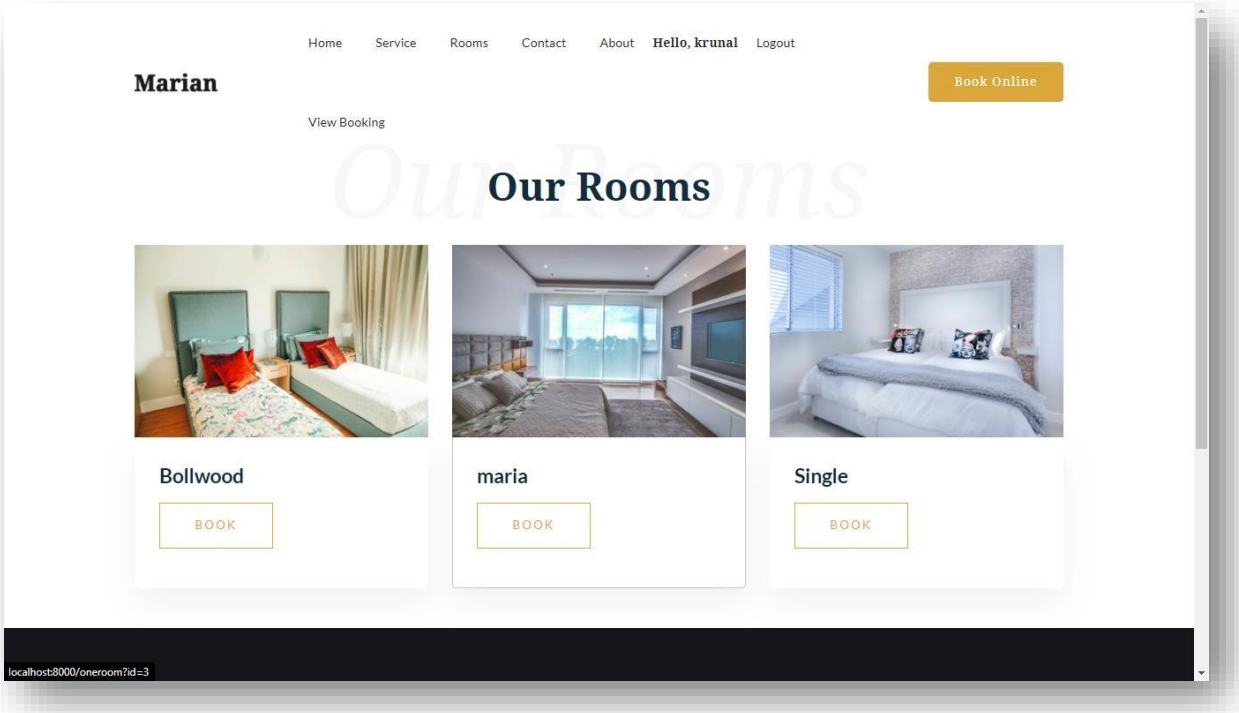
```
def login(request):
   if request.method=='POST':
      username=request.POST['username']
      password=request.POST['password']
      user=auth.authenticate(username=username,password=password) if
      user is not None:
         request.session['username']=username
         auth.login(request,user)
```

```
            return render(request,'hotelapp/index.html')

        else:
            return render(request,'hotelapp/login.html',{'error':'Username or Password Invalid'})
    else:
        return render(request,'hotelapp/login.html')
```

urls.py

```
from django.urls import path
from . import views

urlpatterns = [
 path('login/',views.login,name='login'),
```

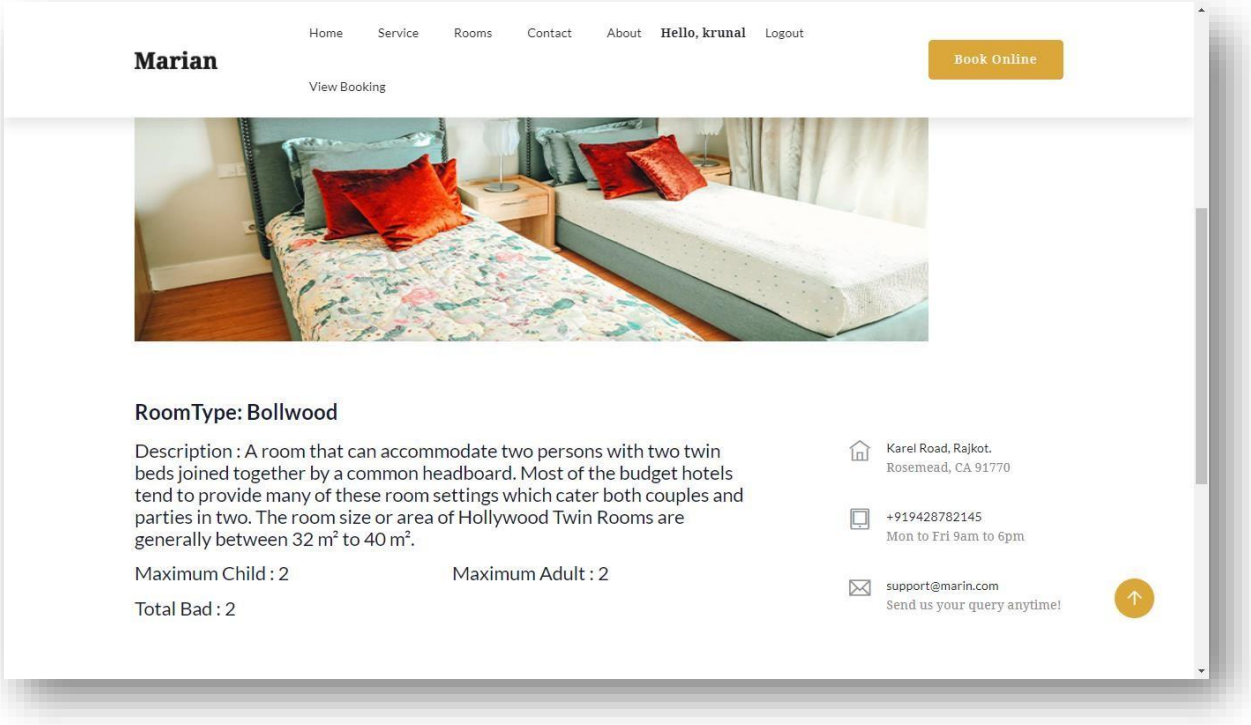
4.1.9 User room Page:



View.py

```
def room(request):
    dests=Room.objects.all()
    return render(request,'hotelapp/rooms.html',{'dests':dests})
```

4.1.10 User Room Dscription Page:



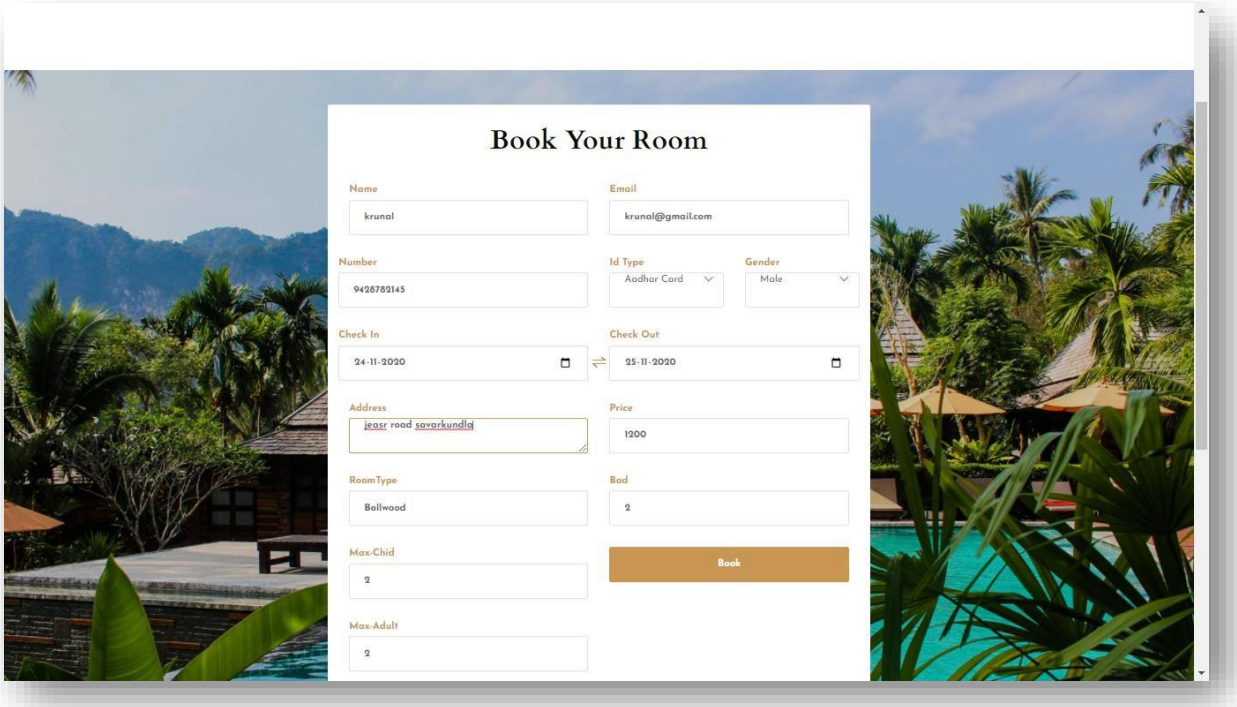
View.py

```
def oneroom(request):
    rec_id = request.GET['id']
    dest= Room.objects.get(pk=rec_id)
    return render(request, 'hotelapp/oneroom.html', {'dest':dest})
```

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
path('oneroom/',views.oneroom,name='oneroom'),
```

4.1.11 User Book_Room Page:



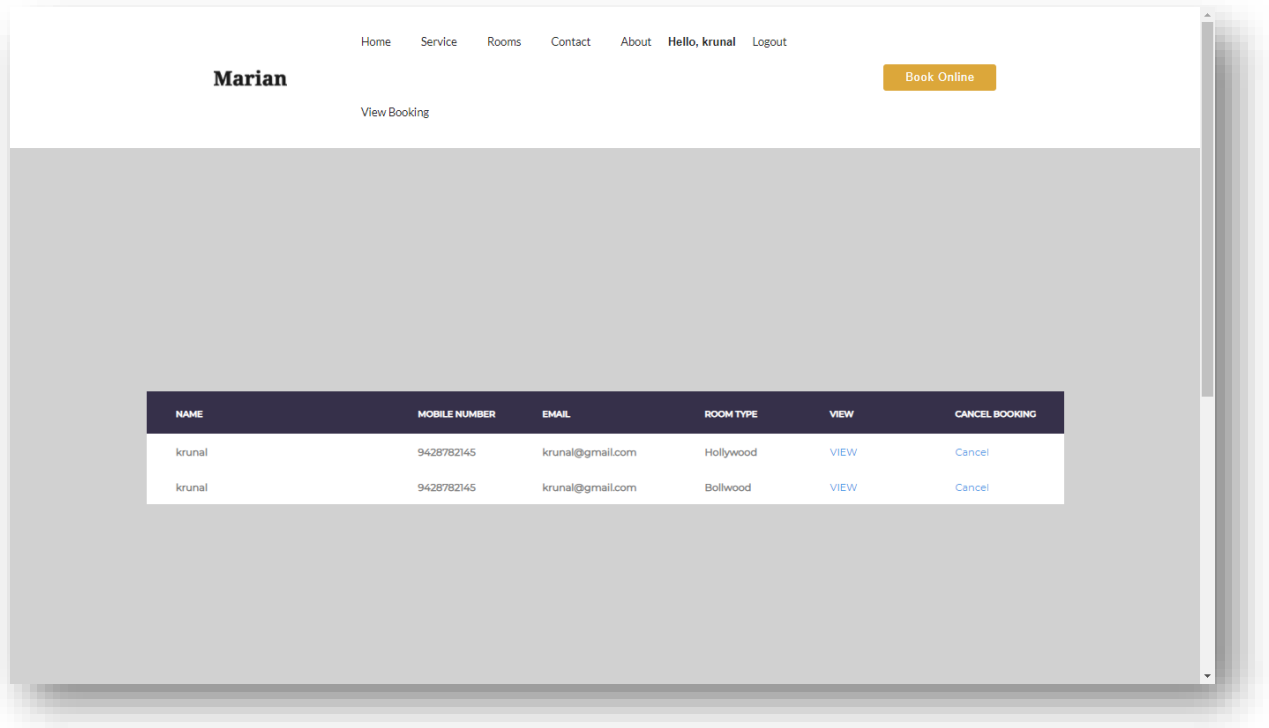
View.py

```
def book(request):
   rec_id = request.GET['id']
   dest= Room.objects.get(pk=rec_id)

   return render(request,'hotelapp/book_room.html',{'dest':dest})
```

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
path('book/',views.book,name='book'),
```

4.1.12 User ViewBooking Page:



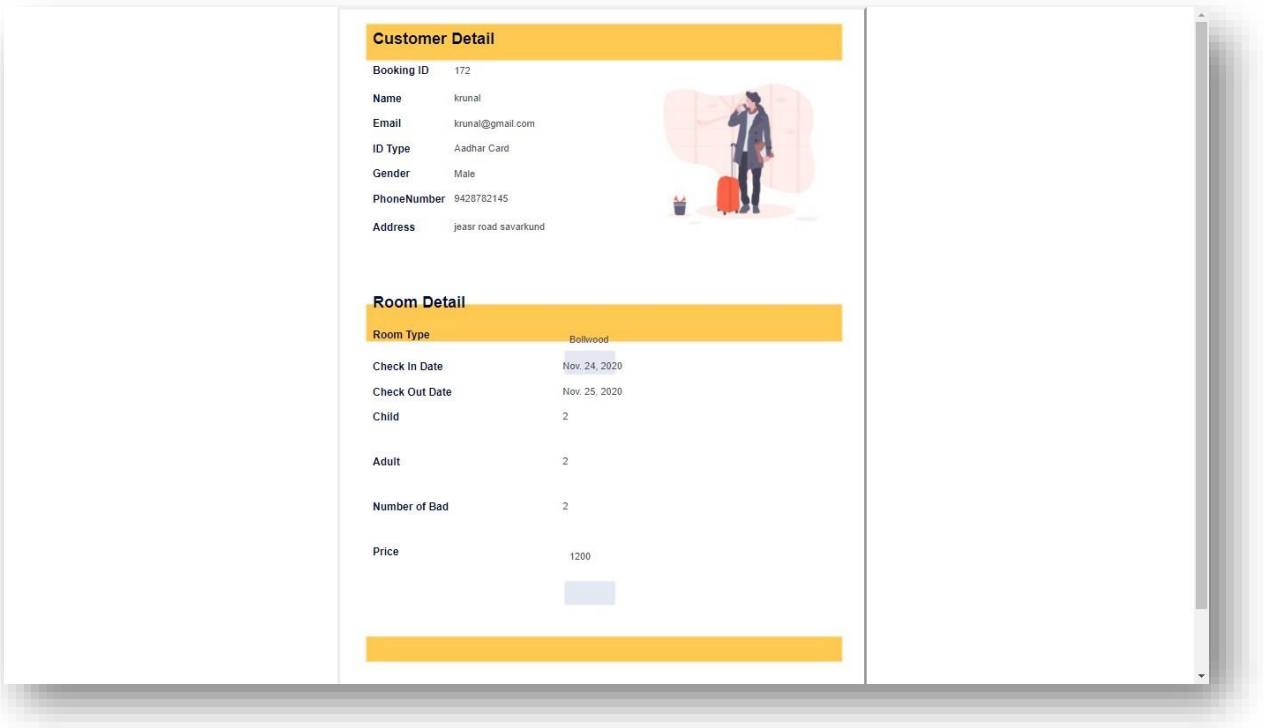
View.py

```
def data(request):
   if User.is_authenticated:
      print(User.username)
      dests=Book_room.objects.all().filter(name=request.session['username'])
      return render(request,'hotelapp/data.html',{'dests':dests})
   else:
      dests=Book_room.objects.all()
   print(dests)
   return render(request,'hotelapp/data.html',{'dests':dests})
```

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
 path('data/',views.data,name='data'),
```

4.1.13 User Booking Deatail Page:



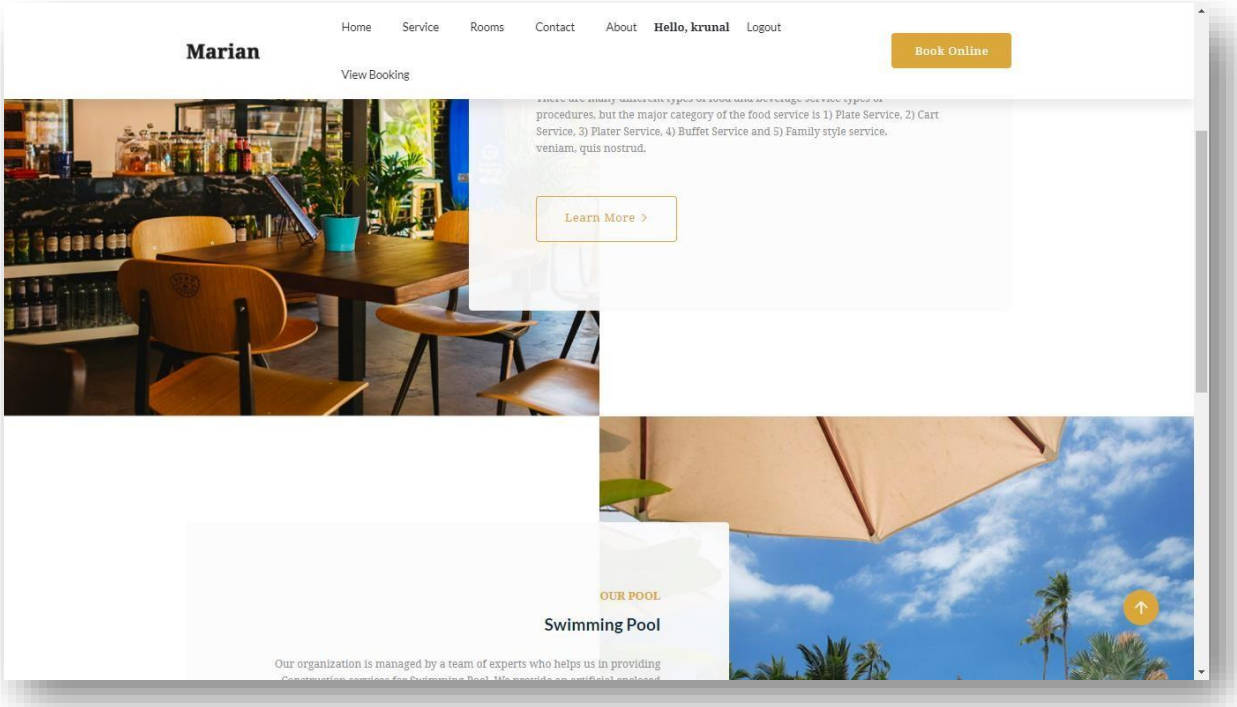
View.py

```
def view(request):
   rec_id = request.GET['id']
   dest= Book_room.objects.get(pk=rec_id)
   return render(request,'hotelapp/booking/final_form.html',{'dests':dest})
```

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [

path('view/',views.view,name='view'),
```

4.1.14 User Service Page:



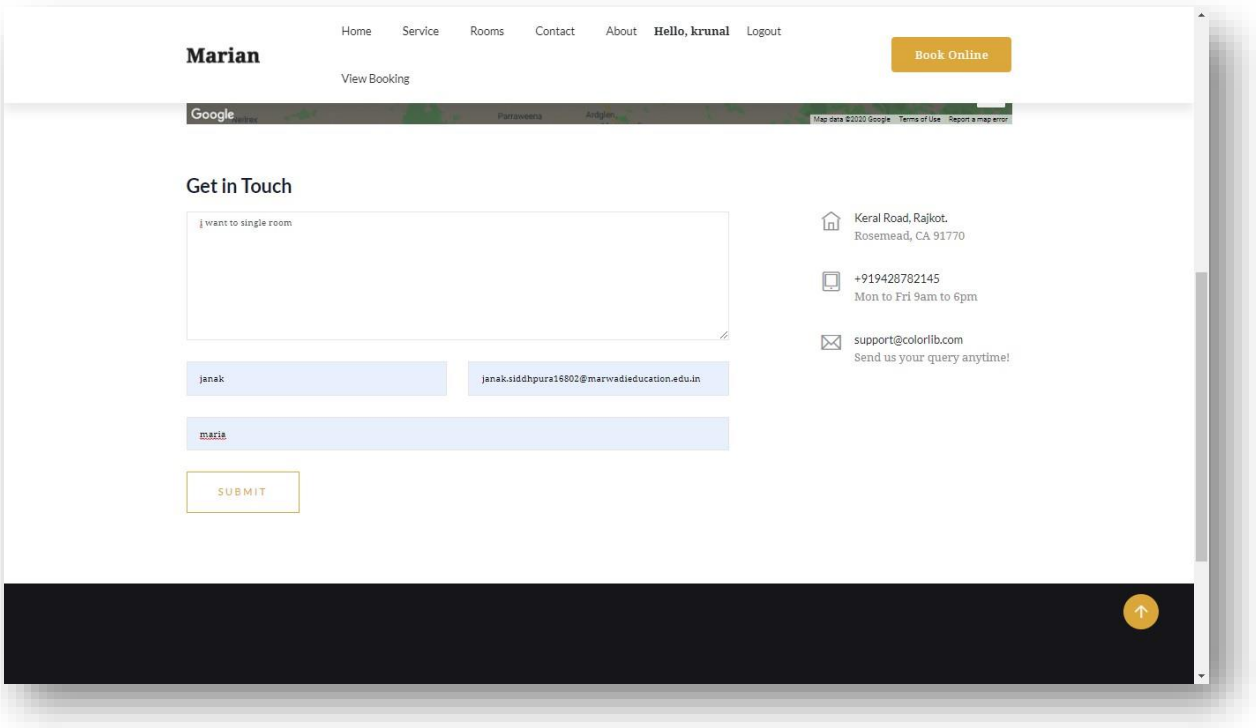
View.py

```
def service(request):
    return render(request,'hotelapp/services.html')
```

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
  path('service/',views.service,name='service'),
```

4.1.15 User Contact Us Page:



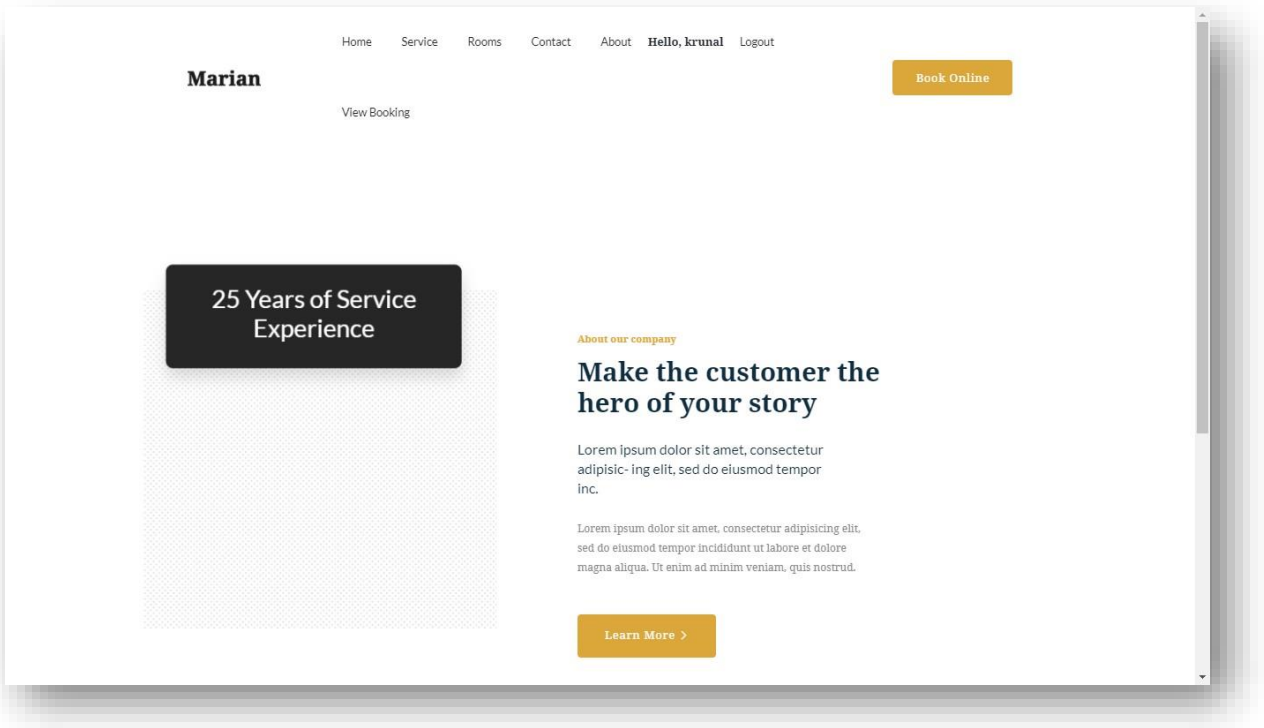
View.py

```
def con_form(request):
   print("This is form contact")
   if request.method=='POST':
      message=request.POST['message']
      name=request.POST['name']
      email=request.POST['email']
      subject=request.POST['subject']
      form=Contact(message=message,name=name,email=email,subject=subject)
      form.save()

      return render(request,'hotelapp/contact.html')
   else:
      return render(request,'hotelapp/index.html')
```

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
path('con/',views.con,name='con'),
```

4.1.16 User About US Page:



View.py

```
def about(request):
   return render(request,'hotelapp/about.html')
```

urls.py

```
from django.urls import path
from . import views

urlpatterns = [
path('about/',views.about,name='about'),
```

5.Agile Documentation

5.1 Agile project charter

A charter is a central document or a set of supporting documents that defines the purpose, nature and characteristics of an about to be undertaken software project.

It is typically constructed early in the project lifecycle, hopefully before the project is staffed and the business is pushing for a delivery date. It is usually created collaboratively as a team and shared with stakeholders upon completion.

It is intended to clearly set the stage for the project to aligning the team and stakeholders by setting goals and expectations.

It is often the case that a charter leads to an early project approval gate as part of an organizational project approval life-cycle phase. Usually the keys to the approval involve cost, schedule, and scope definitions and restrictions- very much of a contractual view.

Project charter of Hotel Booking:

The project includes Hotel Booking system for providing the customers an anytime and anywhere service for booking rooms and providing information about the rooms and their details online.

The manager for the project is Janak Sidhhpura and Shyam Thakar.

The project started on 15th September 2020 and is expected to end on 22th November 2020 with the mission to help the educational institute manage its workflow easily. The vision of this project is to make the website more useful, make the communication more efficient by giving more features.

The Project scope contains functionalities like registration, login, booking rooms, cancel rooms as well as user can also show their bookings.

The project is self-sponsored and self-maintained.

Particular	Details
Project Name	Hotel Booking System
Project Manager	Janak Sidhhpura Shyam Thakar
Stakeholder	Janak Sidhhpura Shyam Thakar
Sponsor	Self-funded
Start Date	15/09/2020
End Date	22/11/2020
Mission	To providing the customers an anytime and anywhere service for booking rooms and providing information about the hotel rooms and their details online.
Vision	To make the Hotel Booking website more useful, make the communication more efficient by giving more features
Scope	Hotel Booking Website will provide many functionalities like book hotel rooms, show room details, user registration etc.

5.2Agile Roadmap/Schedule:

A product roadmap is a plan of action for how a product or solution will evolve over time. Product owners use roadmaps to outline future product functionality and when new features will be released. When used in agile development, a roadmap provides crucial context for the team's everyday work, and should be responsive to shifts in the competitive landscape. Multiple agile teams may share a single product roadmap.

Building the roadmap:

To build a roadmap, product owners take into account market trajectories, value propositions, and engineering constraints. Once these factors are reasonably well understood, they are expressed in a roadmap as initiatives and timelines. Below is a very simple roadmap for a product team. The initiatives are in blue and timelines are indicated by the milestone-markers in red.

Sharing the roadmap:

Once a roadmap is built, it needs to be shared with the entire product team so everyone understands the vision and direction. In many organizations, product owners create their roadmaps in PowerPoint and spreadsheets, and then email the slides and spreadsheets out to the team. While well-intentioned, this strategy is flawed from the start. Each team member has their own copy of the roadmap, and keeping everyone up to speed when and if the roadmap changes are cumbersome.

Most collaboration tools built for this sort of thing will automatically notify all participants of a project letting them know the roadmap has changed. It's important to link your team's work back to the roadmap so you get that whole "context" thing I mentioned above. A tried-and-true way of doing this is to break initiatives down into epics in the product backlog, then further decompose them into requirements and user stories. Tying it all together like that makes it easier for product owners and the development team to make near-term decisions that don't compromise future work. Let's look at an example to see how this plays out.

Roadmap of Hotel Booking Website:

So, this project will start from September, 2020 and will end on November, 2020.

Product (Hotel Booking System):

First 15 days of September, we will decide the roadmap of the product by collecting user requirements. And whole month will be focused on collecting detailed requirements. During August, the team will discuss about the product and will do any changes needed.

After working on the product, in December, Version – 1 of the product will be released. During December, remaining requirements will be collected and work will continue. After final beta testing, on December, the final version will be released.

Development:

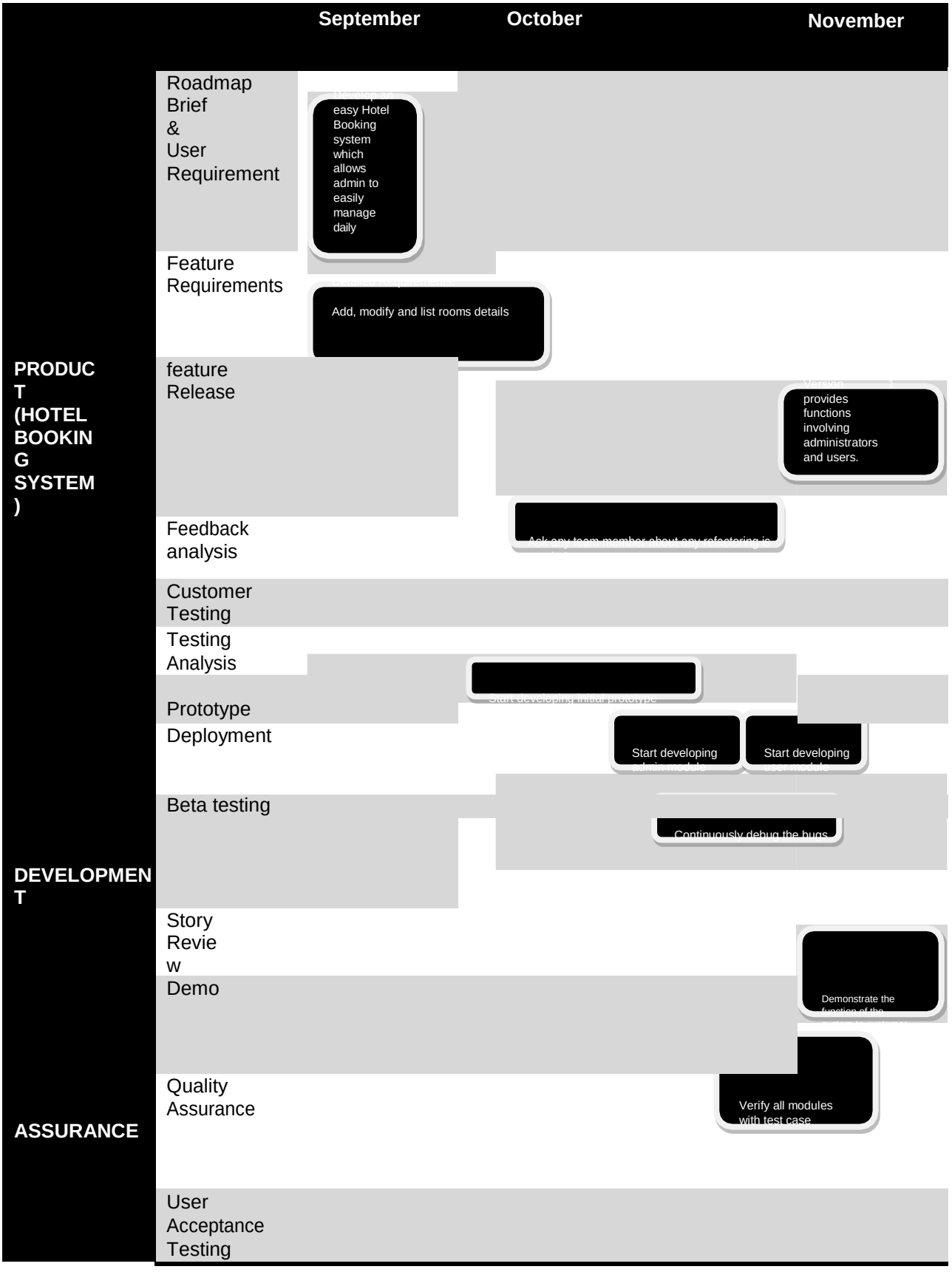
After collecting initial requirements, team will start creating prototype of the product. Consequently, team will start making modules of the system. During this, bugs will be resolved by continuous testing. After version – 1 release, the customer will be given demonstration of the product and reviews will be taken. After additional requirements, the prototype will have to be changed, and the work will continue for final release. At last, User guide will be prepared for making customer familiar with the product.

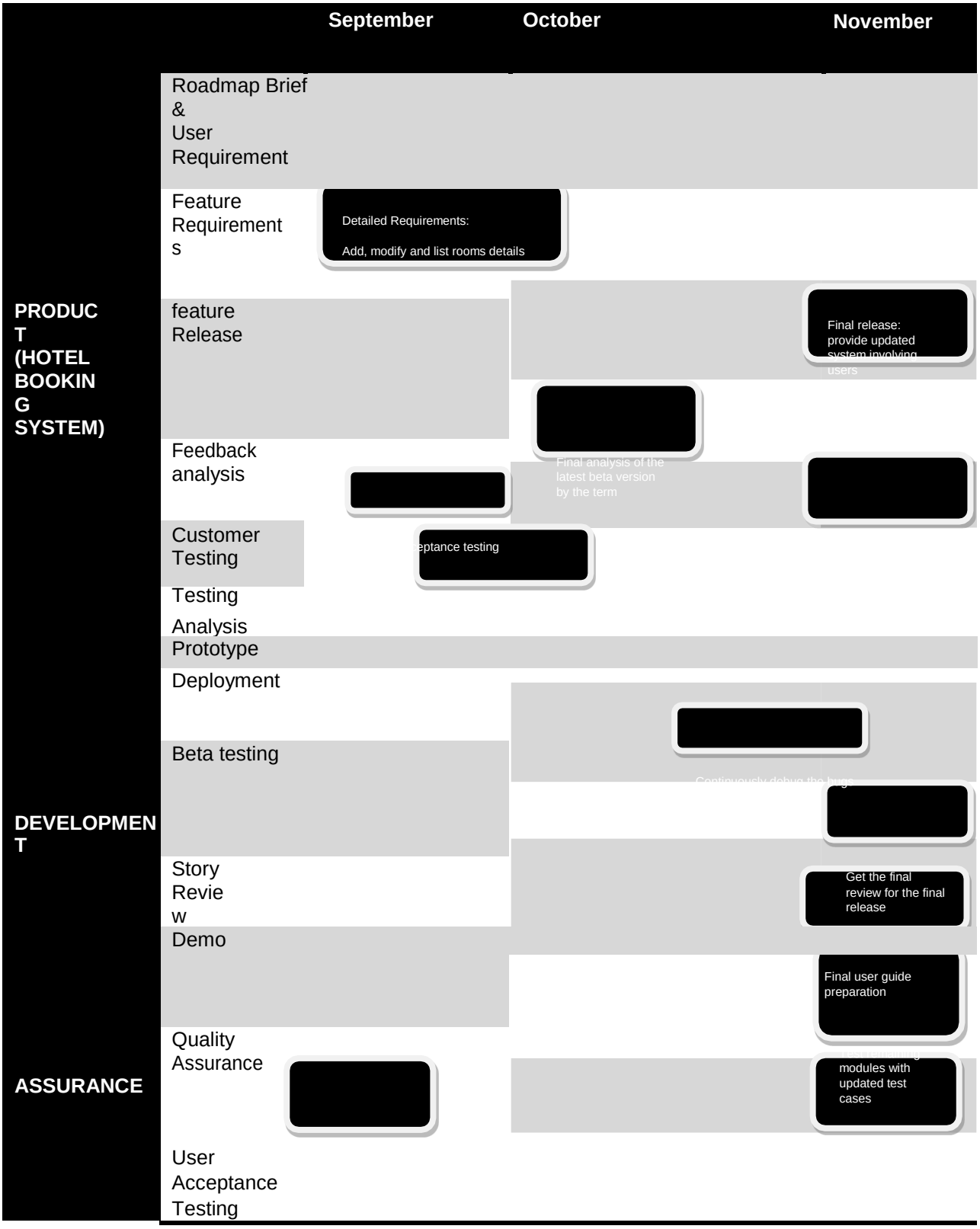
Assurance:

During the development, Quality assurance will be the prime objective of the team. During December, (After v1) all developed modules will be tested by a separate QA team.

After release of v1, User acceptance testing will be done. The same

process will be followed during final release.


September
October
November
Roadmap Brief & User Requirement
easy Hotel Booking system which allows admin to easily manage daily
Feature Requirements
Add, modify and list rooms details
PRODUCT (HOTEL BOOKING SYSTEM)
feature Release
provides functions involving administrators and users.
Feedback analysis
Ask any team member about any refactoring is
Customer Testing
Testing Analysis
Prototype
Deployment
Start developing admin module
Start developing user module
Beta testing
Continuously debug the bugs
DEVELOPMENT
Story Review
Demo
Demonstrate the
Quality Assurance
Verify all modules with test case
ASSURANCE
User Acceptance Testing


September
October
November
PRODUCT (HOTEL BOOKING SYSTEM)
DEVELOPMENT
ASSURANCE
Roadmap Brief & User Requirement
Feature Requirements
Detailed Requirements:
Add, modify and list rooms details
feature Release
Final release: provide updated system involving users
Feedback analysis
Final analysis of the latest beta version by the term
Customer Testing
eptance testing
Testing Analysis
Prototype
Deployment
Beta testing
Continuously debug the bugs
Story Review
Get the final review for the final release
Demo
Final user guide preparation
Quality Assurance
modules with updated test cases
User Acceptance Testing

5.3 Agile Project Plan :

Agile project management is designed to be flexible enough to handle projects with potentially moving, changing and evolving requirements and agile enough to be able to provide the end customer with functional portions of the overall final solutions promptly and as needed.

It is quite common on agile projects for the team to do the planning, not just the manager/coach. Project planning is so important that the organization must make it a top priority to get it right. Organize the project into short iterations.

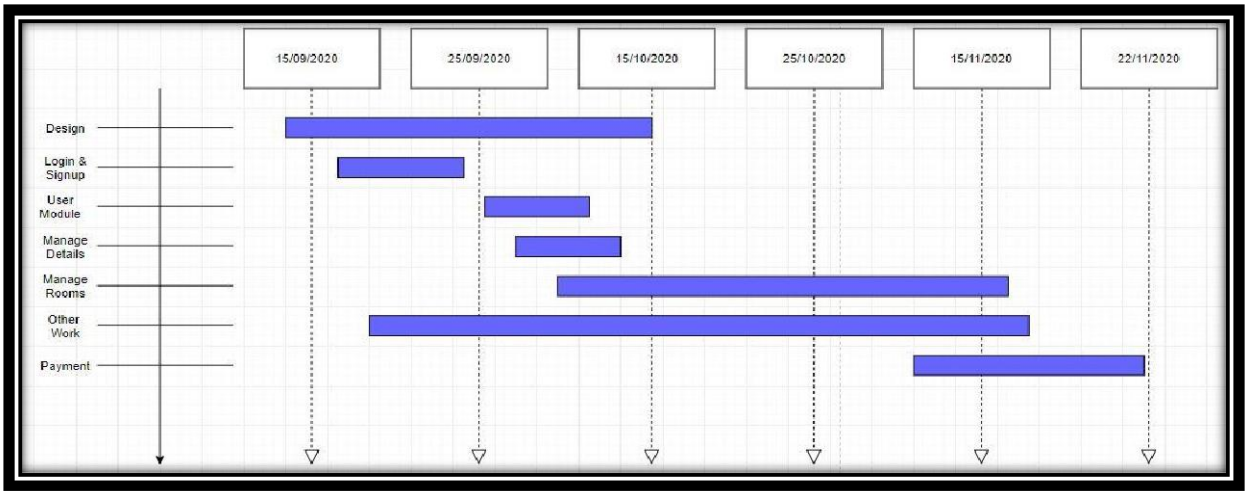
Before each sprint begins, a sprint planning meeting takes place between the product owner and development team members. The user stories and backlog are reviewed to determine the tasks that can be completed during the sprint.

Agile project planning is a collaborative practice which differs from traditional project planning.

The plan projects when features will be delivered to production, without much detail surrounding how those features will be delivered, although the most current iteration tends to have a bit more information.

Agile plans are based on the assumption that we don't really know what conditions will be six months from now, but we can put together a reasonably good guess about what will be delivered when, based on the priority of the features and how much functionality the team can deliver within a given timeframe. Project Plan of Hotel Booking Website:

Figure: Project Plan



5.4 Agile User Story:

Agile is a value-based development methodology, where all those features of the product which can add value to the customer are recognized, prioritized and developed based on customer needs.

These valuable features are best described in users own words. Hence finding out who are the users is an important task. Once all the users are recognized, requirements which add value to the user are written down considering the needs of those specific users.

Since those requirements are coming keeping users in mind, they are called User Stories.

User Stories are best written in following format. As a

<User>, I want to <Have> so that <Benefit> User stories

of Camera Shopping Website:

ID	As a	I want to..	So that I can…
1	Project Manager	Get room status from team leaders periodically	Track the status and report to the customer
2	Team Leader	-Collect status from each team member. -Motivate members to work more efficiently	-Give whole reports to the Project manager -Deliver a quality Product
3	Testing Manager	Encourage members to find maximum defects	Give a fully working product
4	Developer Advocate	Protect my developers from external interference	Provide them a good workplace
5	QA Manager	Have transparent communication with customer	Understand problems faced by customer easily
6	Administrator	Be able to add and modify password and rooms	Get a detailed view of the system
7	Customer	-Ask admin to change some personal details View rooms -User book and Cancel rooms	-System reflects the actual information -Customer can view easily Inform them about bags
8	Manager	-Ask admin to change some personal details -View rooms uploaded by the faculties -View rooms details -Read notices from admin	-System reflects the actual information -Learn by them -Modify accordingly Stay up-to-date about

5.5 Agile Release Plan:

The foundation of a successful project is a well-laid plan. A release plan acts as a project's map, providing context and direction on product goals, vision, and expectations.

Agile release planning is an approach to product management that takes into account the intangible and flexible nature of software development—as part of this approach, teams plan iterative sprints across incremental releases.

In other words, instead of trying to develop every proposed feature in one large, regimented project, agile planning breaks down the development process into stages called releases.

In this context, releases are essentially periods of time set apart to work on a limited scope of the overall project. An agile release plan maps out how and when features (or functionality) will be released and delivered to users.

By scheduling a project into agile releases, product managers can better manage project constraints and adapt to evolving needs or challenges that arise through the development stage while regularly producing product deliverables for the end user.

One of the most important steps in the planning process is defining the vision for your product. The vision will guide subsequent decisions on which features to prioritize, where to focus effort and resources, and how to adapt if the project requires adjustment during development.Release Plan of Hotel Booking Website:

	Project Name	PROJECT MANAGER		START DATE	END DATE		OVERALL PROGRESS		
	Hotel Booking System	Janak Sidhhpura	Shyam Thakar	15-09-2020	22-11-2020		100%		
AT RISK	TASK NAME	FEATURE TYPE		RESPONSIBLE	STORY POINTS	START	FINISH	DURATION (DAYS)	STATUS
	Design			Janak Sidhhpura		17-09-2020	22-09-2020	6	Complete
	Login & Signup Forms			Janak Sidhhpura		23-09-2020	27-09-2020	5	Complete
	User Module			Shyam Thakar		28-09-2020	03-10-2020	6	Complete
	Manage Details			Shyam Thakar		04-10-2020	10-10-2020	7	Complete
	Update rooms			Shyam Thakar		11-10-2020	17-10-2020	7	Complete
	Manage Room Module			Janak Sidhhpura		18-10-2020	25-10-2020	8	Complete
	Upload room Details to Server			Janak Sidhhpura		26-10-2020	31-10-2020	6	Complete
	Give Feedback			Janak Sidhhpura		01-11-2020	03-11-2020	3	Incomplete
	Cancel Booking			Janak Sidhhpura		04-11-2020	06-11-2020	3	Complete
	Accept Booking Details			Shyam Thakar		07-11-2020	16-11-2020	10	Complete
	Payment			Janak Sidhhpura		17-11-2020	22-11-2020	6	Incomplete

5.6 Agile Sprint Backlog :

Sprint Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

A Sprint Backlog is never complete. The earliest development of it lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves.

The Sprint Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. If a product exists, its Product Backlog also exists. Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog.

This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised.

A typical backlog comprises the following different types of items: Features

-Bugs

-Technical work

-Knowledge acquisition

User stories of Hotel Booking Website:

ID	TASK	ASSIGNED TO	START	FINISH	PRIORITY	STATUS
1	Senior developer will ask for progress from developers	Shyam Thakar	17-09-2020	22-09-2020	Medium	Done
2	Programmers will develop all the modules	Janak Sidhhpura	28-09-2020	03-10-2020	Medium	Done
3	Designers will prepare UI/UX patterns	Shyam Thakar	11-10-2020	17-10-2020	High	Done
4	DBA required to develop database design	Janak Sidhhpura	26-10-2020	31-10-2020	High	Done
5	Senior tester will guide all the testers for testing	Shyam Thakar	04-11-2020	06-11-2020	Medium	Done
6	Testing team will create test cases and perform them	Janak Sidhhpura	17-11-2020	22-11-2020	Medium	Done

5.7 Agile Test Plan :

Unlike the Waterfall method, Agile Testing can begin at the start of the project with continuous integration between development and testing. Agile Testing is not sequential (in the sense it's executed only after coding phase) but continuous.

An agile team works as a single team towards a common objective of achieving Quality. Agile Testing has shorter time frames called iterations (say from 1 to 4 weeks). This methodology is also called release, or delivery driven approach since it gives a better prediction on the workable products in short duration of time.

The agile manifesto clearly favours working software over comprehensive documentation and responding to change over following a plan.

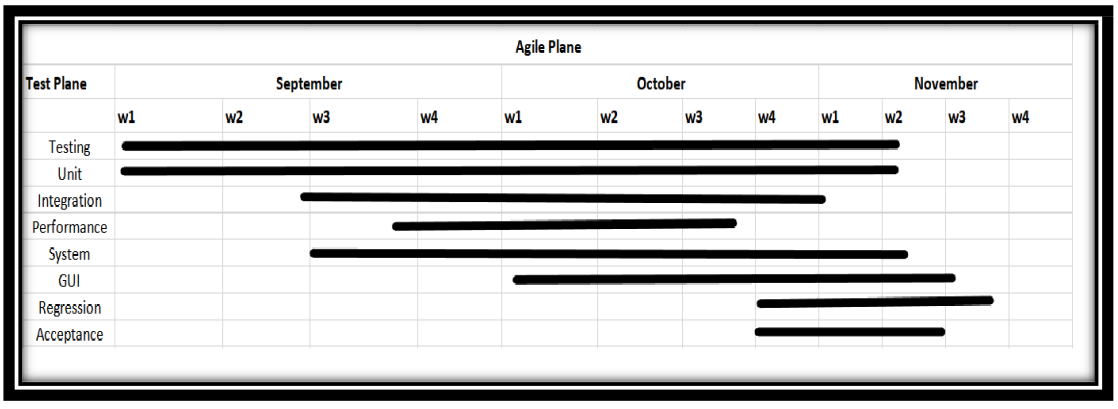
In an agile environment, the contents of a release (the items) are discussed before the sprint so the testing team knows in advance what the scope is and what should be tested.

During the sprint, QA are continuously testing new code/feature. Test planning becomes a dynamic activity as the priorities for the day change. Testing is based on what is the activity for the day and the outcome of the day before.

For Agile, test planning should map risks and rewards of testing any feature. In order to achieve this, a tester should communicate the business importance and priority of features they would be testing in the sprint.

Testers should evaluate the acceptance criteria rather than copy-pasting it. This allows testers to examine the significant features carefully.

Test plan of Hotel Booking Website:



5.8 Earned-value and Burn Charts

A burn down chart is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. That is, it is a run chart of outstanding work. It is useful for predicting when all of the work will be completed.

It shows the total effort against the amount of work we deliver each iteration. The obvious benefit of a burn down chart is that it provides an updated status report on the progress of the project. Having a visual representation of this most important data keeps everyone on the same page.

The burn down chart is extremely helpful, too, because of its simplicity. It's a great way to see the velocity history of the project. Velocity is an agile term that means the total effort estimates associated with user stories that were completed during iteration.

6.Proposed Enhancements

Payment module is to be added to make payment for the bookings online by wallets, debit cards, credit cards or net banking.

Rating to the bookings can also be given for other users.

7.Conclusion

At the end we would like to conclude that from the above discussion it becomes clear computerized **Hotel Booking** is fare better than manual system, due to the advantage of speed, accuracy and other above discussed. Also we would like to say that our project fulfill all field all the requirements of a well organize **Hotel Booking** Moreover we would like to specify that our project have greater scope of a adapting the today's environment in this competitive area. If needed be you concern, you can contact the owner of the project we make any modification as per your requirements. We value your opinion & want to know what we are doing right, what we could do better & other words of wisdom you are willing to pass our any the end.

8.Bibliography

8.1 Online References:

- www.youtube.com
- www.w3cschools.com/python
- www.tutorialspoint.com/python
- https://www.djangoproject.com/

8.2 Offline References :

- R Nageswara Rao,Core Python Programming,2nd Edition,Dreamtech Pearson.
- John V Guttag,"Introduction to Computation and Programming Using Pythonn",Prentice Hall of India.
- Build Your First Website with Django 2.1:Master the Basics of Django while Building a Fully-Functioning Website