CMR UNIVERSITY

Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

CAPSTONE REPORT
ON

"BONPOD"

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

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

PRANAV KUMAR
(16UG08032)
PRANJAL GUPTA
(16UG08034)
SAPNA KUMARI
(16UG08046)
SARANYA T
(16UG08047)

Under the Guidance of:
Dr. Rubini P
Associate Professor
Dept. of CSE, SOET
Bengaluru

Department of Computer Science and Engineering
CMR University
Off Hennur - Bagalur Main Road,
Near Kempegowda International Airport, Chagalahatti,
Bengaluru, Karnataka-562149
2020

CMR UNIVERSITY

Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

Chagalahatti, Bengaluru, Karnataka-562149

Department of Computer Science and Engineering

CERTIFICATE

Certificate that the capstone work entitled **Bonpod** carried out by Mr. Pranjal Gupta USN:16UG08034, Mr. Pranav Kumar USN:16UG08032, Ms. Sapna Kumari USN:16UG08046, Ms. Saranya T USN:16UG08047, a bonafied student of CMR UNIVERSITY in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering, Bengaluru for the academic year 2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

Signature of Guide	Signature of HOD	Signature of Dean
……………………	……………………	.....………………
Dept. of CSE SoET, CMRU, Bangalore	Dept. of CSE SoET, CMRU, Bangalore	SoET, CMRU, Bangalore

External Viva:

Name of the Examiners:	Signature with Date:
1 …………………….	………………………….
2 …………………….	………………………….

DECLARATION

We, **PRANJAL GUPTA** bearing USN **16UG08034**, **PRANAV KUMAR** bearing USN **16UG08032**, **SAPNA KUMARI** bearing USN **16UG08046**, **SARANYA T** bearing USN **16UG08047**, student of Bachelor of Technology, Computer Science and Engineering, CMR University, Bengaluru, hereby declare that the capstone work entitled "BONPOD" submitted by us, for the award of the Bachelor's degree in Computer Science and Engineering to CMR University is a record of bonafide work carried out independently by me under the supervision and guidance of Dr. Rubini P Asst. Prof, CSE Dept. CMR University.

I further declare that the work reported in this capstone work has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this university or any other institute or University.

Place: Bengaluru
Date:

Pranav Kumar
(16UG08032)
Pranjal Gupta
(16UG08034)
Sapna Kumari
(16UG08046)
Saranya T
(16UG08047)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain.

We consider ourselves privileged to express gratitude and respect towards all those who guided us through the completion of the project. We express our thanks to our Internal Project Guide **Dr. Rubini P, Associate Professor**, Computer Science and Engineering, School of Engineering and Technology, CMR University for her constant support.

We would like to express our thanks to **Dr. Rubini.P, Professor and Head**, Computer Science and Engineering, School of Engineering and Technology, CMR University, Bangalore, for his encouragement that motivated me for the successful completion of Project work.

We express our heartfelt sincere gratitude to **Dr. Jayaprasad M, Dean,** School of Engineering and Technology, CMR University for his support.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

India, a place of diverse culture, climate, terrain fascinates people from all over the world to visit mountainous region of Jammu and Kashmir to beaches of Goa. To do so, a pod hotel is a new affordable type of hotel that attracts backpackers and young tourists. The purpose was to provide accommodation for an individual at last minute stay-over at very low price compared to a lodge.

These hotels provide pods to lay down in two-tire and next to each other and steps provided to climb (like in 2 tire train). The size varied from 24-inch x 60 inch and higher just to accommodate single person. Toilets and bathrooms are common in ratio of 10:1, i.e. for every 10 pods there is 1 Bathroom/ toilet. This ratio can be lower in a high-end pod hotel, which are slightly expensive. For luggage a small locker is provided.

Now depending upon the standard and rate, pod is providing with a tiny TV, Wi-Fi connectivity, A/C. This concept is picking up in India as well and has a different kind / purpose market.

This concept has to be targeted with Sports Association, Biker's Associations, etc. OR have to come up in the places where there is lot of activities like, Common Entrances Exams, Mass Interviews, Interstate / district sports or cultural or art contests etc.

CHAPTER 1

INTRODUCTION

1.1 Overview

A “BONPOD” hotel is also known in the western world POD hotel is a typed of hotel developed in Japan that features many small bed-sized rooms known as capsules. Bonpod hotels provide cheap, basic overnight accommodation for guests who do not require or who cannot afford larger, more expensive rooms offered by more conventional hotels. Pod hotels with capsule-size rooms just enough for a bedding, a sliding shutter for privacy, and walls decked out with just a plasma television, mirror and hangers, are gradually catching travelers fancy. The capsule could be around seven-feet long and three- feet high and carry no extra frills, where a traveler would most certainly use the ‘rooms’ to simple tuck into bed for a good night of rest. Designed primarily for the budget-conscious traveler, pods claim to offer “comfortable sleeping/resting spaces at affordable price-points” and have locker facility for luggage and valuables, besides common restrooms.

If you think contemporary, innovative and smart hotels only boast of four-poster beds, eclectic interiors and luxury bath-tubs on Italian-marble flooring in ‘extra’ large rooms, then you may be wrong. Sleek in design and compact in size, with shared dining, relaxation and bath spaces, all a part of new range of pods, are enticing business and leisure travelers who yearn to experiment and go beyond the cliché. The chamber walls may be made of wood, metal or any rigid material, but are often fiber- glass or plastic. Amenities within the room generally include a small television, air conditioning, an electronic console, and power socket. The Bonpod are stacked side-by-side, two units high, with steps or ladders providing access to the second level rooms, similar to Bunks bed. The open end of the Bonpod can be closed with a curtain or a solid door for privacy, but cannot be locked. Like a hostel many amenities are communally shared, including toilets, showers, wireless internet, and dining rooms. Bonpod hotels vary in size, from 50 or so capsules to 700, and primarily cater to men Some Bonpod hotels offer separate sections for male and female guests, or even separate floors and separate elevators. Luggage and valuables are usually stored in Lockers or if available in-room safes. Guests are asked not to smoke or eat in the Bonpod.

Pods, originated in the Japanese city of Osaka in the late 1970s, were considered an example of Japanese efficiency and futuristic design. The trend, since then, has moved into several other Japanese cities, Singapore, China, the US and most recently, in India. Although a mere dot on the market size of the global hospitality segment, pod hotels are believed to carry high potential. A report by WiseGuy Research Consultants states the global capsule hotel market will reach $226 million by 2022, from about $159 million in 2016.

“It is a generation next, futuristic, smart accommodation and shared-living space with modest prices. The pod- theme and setting makes it a great arena to network and make new friends,” says Hiren Gandhi, co- founder of UrbanPod, a pod hotel in Mumbai.

Pod hotels are now catching on with Indian travelers, say experts. Gandhi says they have witnessed several people in the 30-45 years’ bracket opting for pods and in the last one year have hosted about 10,000 guests. “In addition, a sizeable number of women travelers prefer pods especially since we have an exclusive ladies’ pods section.” Countries like India have modified the Japanese version of pod hotels to slightly larger accommodations and also private baths.

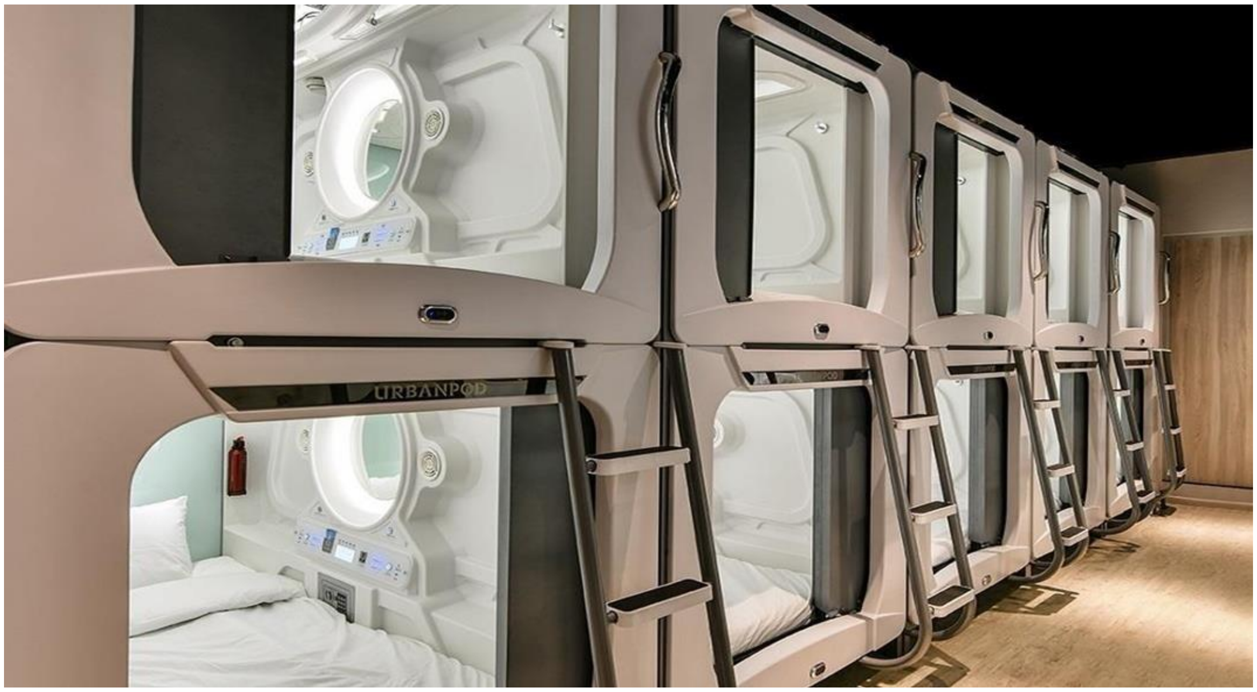


Fig 1.1: Outer View of Pod room

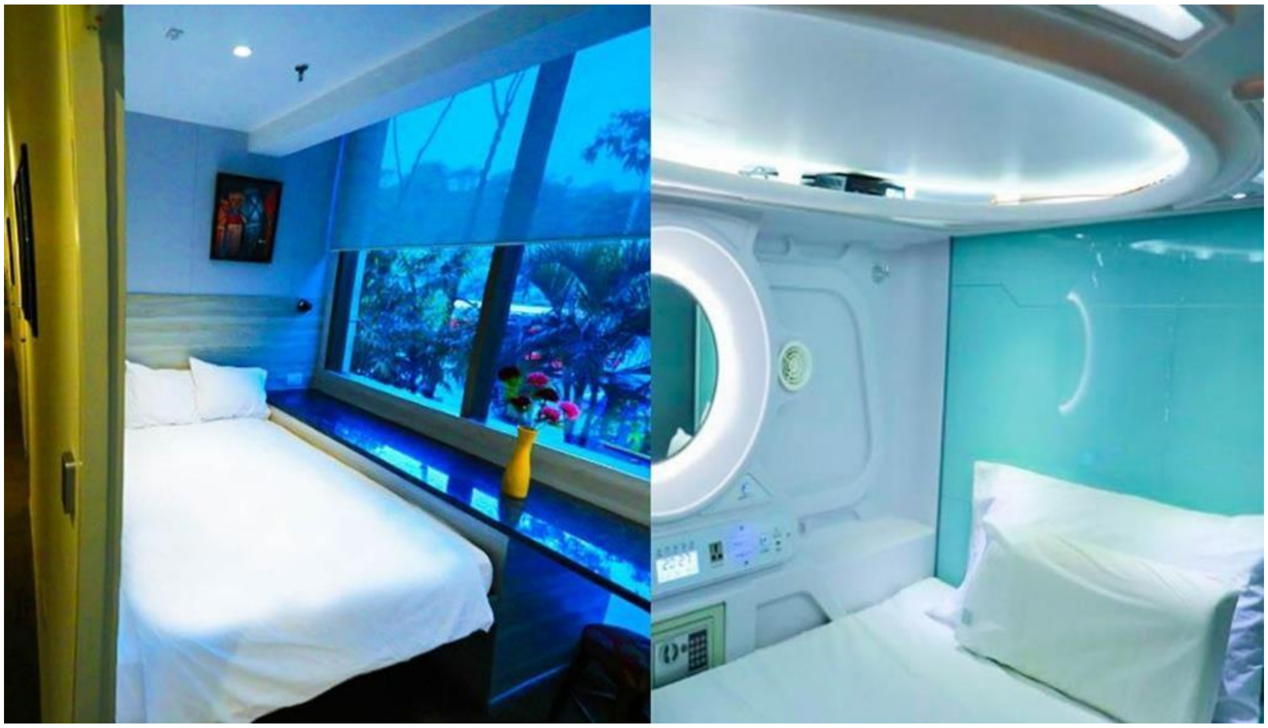


Fig 1.2: Inner View of the pod room

1.2 Problem Statement

Though A “BONPOD” hotel provide cheap, basic overnight accommodation for guests who do not require or who cannot afford larger, more expensive rooms offered by more conventional hotels which is a typed of hotel developed in Japan that features many small bed-sized rooms known as capsules.

But being one of the best accommodation service, it does not have a trend in one of the largest population country like INDIA. A night’s stay in a pod can cost from Rs 1,000-2,000 per head. But in an era when Airbnb, budget hotel networks like Treebo and OYO, and budget brands of international chains like Ibis, Ginger, etc., are making major strides, why would individuals still come to a pod? Wouldn’t travelers prefer paying Rs 500 or even Rs 1000 more to get a decent size room with a ‘king’ or ‘queen’ bed and private baths. “A number of traveler segments require purely sleep or rest infrastructure. Budget hotels have been there, but a sleek hygienic pod is bound to have higher appeal. Apart from business travelers, students travelling to write exams and people going on pilgrimages Could benefit from pod hotels. A number of underserved destinations like Shirdi, Sabarimala or Vaishno Devi can benefit with this format of accommodation,”
So, this Idea of developing a website and taking out the concept of POD hotels in INDIA is one of the good start for moving into a futuristic accommodation.

1.3 Objectives

“Bonpod, A POD hotel” is developed with the main aim of providing cheap, basic overnight accommodation for guests who do not require or who cannot afford larger, more expensive rooms offered by more conventional hotels. The BONPOD are stacked side-by-side, two units high, with steps or ladders providing access to the second level rooms, similar to Bunks bed. The open end of the bonpod can be closed with a curtain or a solid door for privacy, but cannot be locked. Like a hostel many amenities are communally shared, including toilets, showers, wireless internet, and dining rooms. Bonpod hotels vary in size, from 50 or so capsules to 700, and primarily cater to men Some Bonpod hotels offer separate sections for male and female guests, or even separate floors and separate elevators. Luggage and valuables are usually stored in Lockers or if available in-room safes. “This new version is getting popular amongst backpackers and individual travelers”.

1.4 Methodology

For our web portal we used the Prototyping model to take reviews from the user at each and every step to ensure we meet the user requirements.

Prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation.

Prototyping is used to allow the users evaluate developer proposals and try them out before implementation. It also helps understand the requirements which are user specific and may not have been considered by the developer during product design.

Following is a stepwise approach explained to design a software prototype.

1.4.1 Basic Requirement Identification

This step involves understanding the very basics product requirements especially in terms of user interface. The more intricate details of the internal design and external aspects like performance and security can be ignored at this stage.

1.4.2 Developing the initial Prototype

The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed. While, the work around are used to give the same look and feel to the customer in the prototype developed.

1.4.3 Review of the Prototype

The prototype developed is then presented to the customer and the other important stakeholders in the project. The feedback is collected in an organized manner and used for further enhancements in the product under development.

1.4.4 Revise and Enhance the Prototype

The feedback and the review comments are discussed during this stage and some negotiations happen with the customer based on factors like – time and budget constraints and technical feasibility of the actual implementation. The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until the customer expectations are met.

Prototypes can have horizontal or vertical dimensions. A Horizontal prototype displays the user interface for the product and gives a broader view of the entire system, without concentrating on internal functions. A Vertical prototype on the other side is a detailed elaboration of a specific function or a sub system in the product.

The purpose of both horizontal and vertical prototype is different. Horizontal prototypes are used to get more information on the user interface level and the business requirements. It can even be presented in the sales demos to get business in the market. Vertical prototypes are technical in nature and are used to get details of the exact functioning of the sub systems. For example, database requirements, interaction and data processing loads in a given sub system for achieving relevant data in a proper format.

CHAPTER 2
LITERATURE SURVEY

This section reviews the research works carried out by different researchers that are related to the proposed work. The following research papers include papers on languages used for web development and their pros and cons. This section also includes papers published by students on their college website development projects which will be briefed in the later part of this section. This section reviews the research works carried out by different researchers that are related to the proposed work. The data used for the website or processed by the website are stored in the data bases.

In[1]: Sue et al (2005) ‘the Hotel Proprietors Act 1956 provides a clear definition of a hotel: An establish held out by the proprietor as offering food, drink and, if so required, sleeping accommodation, without special contract, to any traveller presenting himself who appears able and willing to pay reasonable sum for the services and facilities provided and who is in a fit state to be received.’

In[2]: Japanblog(2007) “A capsule hotel is a type of hotel accommodation popular mostly among travellers stranded for the night and those who are on a tight budget.” Capsules hotel are usually about 2m * 1m*1.5m in dimension and so is unsuitable for those above six feet tall. Despite the very limited space the customers will be surprised at how well equipped these hotels are. Each capsule usually comes with a built in mini TV, radio, and alarm clock, plus fresh bedding of course. Customers can also use the hotel’s public lounge space including lockers, shower rooms, laundry facilities, restaurants, vending machines, and clothes shop.

In[3]: Chon (2000)has suggested all psychological motives could be classified as basic motives and secondary motives. In the hospitality industry, people’s basic needs from a lodging facility are a good night’s rest in a comfortable and soft room and food. Secondary motives are fulfilled by celebrating special moments, such as theme night, birthdays and so on

In[4]: Middleton and Clarke (2001:19)suggest, in term of customer, marketing is concerned with understanding the needs and desired of existing and future buyers. It is also about which products they choose, when, how much, at what price and how often they purchase them. It is concerned with how they get information about the products, where they buy the products from-whether they are bought directly of through retail intermediary-and how the customers feel after their purchase and consumption of product.

In[5]: Itravel,com(2007)defined since most visitors to a Capsule hotel are Japanese business men who don’t have time to go home, there are amenities there for people who didn’t plan on staying away from home.

In[6]: Capsuleinn (2005) described: “the actual sleeping room is a capsule unit made of reinforced plastic and designed in the image of a jet airplane’s cockpit.”

In[7]: Holloway (2003) suggested motivation for can be divided into different sectors including holiday travel, business travel, visiting relatives, health travel and religious travel, and continuing to state that it is related to five basic needs, which is cultural, physical, interpersonal, status and prestige and commercial.

In[8]: (Darcy, 2010; Hwang and Chang, 2003). According to the Ministry of Culture, Sports and Tourism (2012), the number of foreign tourists in Korea increased between 2006 and 2010, and although the number of hotels in the metropolitan area increased by 2.9% at the same time, supply has not kept up with demand. In addition to the shortage of accommodations, tourism facilities are concentrated around seoul , the capital of korea.

In[9]: Albrecht and Johnson, 2002; Leslie, 2006; Life Style, 2016) the first capsule hotel, Darak Hue, opened at the Incheon International Airport in Korea. In general, a capsule hotel, a concept originally developed in Japan, provides minimum capsule-shaped rooms where only one guest can sleep and provides minimal comfort facilities such as a shower or television.

In[10]: Thakur Ranjit Singh et al, from SSMJ , Bhilai published a paper on web portal features as Knowledge management system in school education in which he stated how the development of Web has affected different aspects of our lives, such as communication, sharing knowledge, searching for jobs, social activities, etc. The web portal as a gateway in the World Wide Web is a starting point for people who are connecting to the Internet. The web portal as the type of knowledge management system provides a rich space to share and search information as well as communication services like free email or content provision for the users. This research aims to discover the university needs to the web portal as a necessary tool for students in the schools to help them in getting the required information. A survey was conducted to gather students‟‟ requirements which can be incorporated in to portal to be developed.

In[11]:Fatemeh Bordbar et al, from Southern Utah University published a paper on the Effectiveness of Website Design in Higher Education Recruitment. This paper explores the effectiveness of visual elements and aesthetics in university websites for recruiting prospective students. College and university websites are the gateway to an educational institution and the public face for both academics and athletics. According to Ruffalo Noel Levitz‟‟s study on consumer behavior (Levitz, 2014), websites play an important role in providing information for prospective students (both new and transfer), current students, parents, and alumni. Universities‟‟ recruiters constantly strive to maximize the utility and depth of information on their websites while offering pleasing and powerful aesthetics to attract potential students and donors. Entirely qualitative in content, this study examines the role of aesthetics in the three key categories: web design, website functionality, and universities‟‟ online recruitment strategies. The website functionality key category divides further into subcategories including website usability, accessibility and credibility.

In[12]:Pierre Genevès et al, published paper on the Analysis of Cascading Style Sheets(CSS). Developing and maintaining cascading style sheets (CSS) is an important issue to web developers as they suffer from the lack of rigorous methods. Most existing means rely on validators that check syntactic rules, and on runtime debuggers that check the behavior of a CSS style sheet on a particular document instance. However, the aim of most style sheets is to be applied to an entire set of documents, usually defined by some schema.

To this end, a CSS style sheet is usually written w.r.t. a given schema. While usual debugging tools help reducing the number of bugs, they do not ultimately allow to prove properties over the whole set of documents to which the style sheet is intended to be applied. We propose a novel approach to fill this lack. We introduce ideas borrowed from the fields of logic and compile-time verification for the analysis of CSS style sheets. We present an original tool based on recent advances in tree logics. The tool is capable of statically detecting a wide range of errors (such as empty CSS selectors and semantically equivalent selectors), as well as proving properties related to sets of documents (such as coverage of styling information), in the presence or absence of schema information. This new tool can be used in addition to existing runtime debuggers to ensure a higher level of quality of CSS style sheets.

In[13]:Ch Rajeshet al, from ANITS, Visakhapatnam presented a research paper on HTML5 in Web Development which said, The latest research on HTML by W3C is to create a standard that handles all the jobs that the proprietary technologies performing currently. W3C to increase web openness and platform independence is developing HTML5 with cooperation of Web Hypertext Application Technology Working Group (WHATWG) as a standard that facilitates the users and developers with intensified functionality without much using the additional plug-ins.

In[14]: Punam Kumari et al from MDU, Rohtak, Haryana submitted a research paper on Website Development Optimization using Xampp/PHP. This research paper discusses the various useful tools and techniques that are used in a development of a website. They also discuss about the procedure followed in a website, mostly focused on a local host named Xampp tool. Next, they compare different development frameworks web application and in addition to that they also discuss life cycle model and framework development of web application. This Paper tells about the technologies used in this development, PHP and explained in result its functionality with Xampp.

In[15]:Andr´e Lu´ıs dos Santos Domingues et al, from Brazil have submitted a paper on the Comparison Study of Web Development Methods as Development methods for Web applications give very few attentions for the conception, planning, testing and client evaluation stages. This paper presents such problems, presenting a comparison study ofWeb applications methods based on their main features and supporting mechanisms. Considering the fact that most of the methods have limitations, they present a case study based on a Web portal in order to describe advantages and disadvantages of some selected methods.

In[16]:Y.Zubritska et al , from Zhytomyr Ivan Franco State University published a paper on recommendation of HTML, CSS and JavaScript usage. The paper aims to give the advice to beginners and professional web developers to avoid problems with slow loading pages, encumbered code and to reduce the time for debugging there code. So, there are guidelines that will help to improve code quality and to facilitate collaboration and to support infrastructure. These recommendations present themselves as the best practices for writing HTML/CSS/JavaScript. The paper suggested ways to use them in code optimize your code and to avoid bugs.

CHAPTER 3

SOFTWARE AND HARDWARE REQUIREMENTS

3.1 HARDWARE REQUIREMENTS:

- Processor (CPU) with 2 gigahertz (GHz) frequency or above
- A minimum of 2 GB of RAM
- Monitor Resolution 1024 X 768 or higher
- Internet Connection Broadband (high-speed) Internet connection with a speed of 4 Mbps or higher

3.2 SOFTWARE REQUIREMENTS:

3.2.1 Hypertext Markup Language (HTML) : is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages.

Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

3.2.2 Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

3.2.3 JavaScript: JavaScript is what is called a Client-side Scripting Language. That means that it is a computer programming language that runs inside an Internet browser The way JavaScript works is interesting. Inside a normal Web page, you place some JavaScript code. When the browser loads the page, the browser has a built-in interpreter that reads the JavaScript code it finds in the page and runs it.

Web page designers use the Java Script in many ways. One of the most common is to do field validation in a form. Many Web sites gather information from users in online forms, and JavaScript can help validate entries.

3.2.4 JQuery: JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensible, j query has changed the way that millions of people write JavaScript.

J query's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax application. This enables developers to create abstraction for low-level interaction and animation, advanced effects and high-level, theme able widgets.

3.2.5 Server Scripting(Python): Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for each user's request to the website. The alternative is for the web server itself to deliver a static web page. A scripting language is a programming language designed for integrating and communicating with other programming languages. The scripting language is basically a language where instructions are written for a run time environment. They do not require the compilation step and are rather interpreted. It brings new functions to applications and glue complex system together. Scripting languages are used in web applications. It is used in server side as well as client side. Server side scripting languages are: JavaScript, PHP, Perl etc. and client side scripting languages are: JavaScript, AJAX, jQuery etc. Scripting languages are used in system administration. For example: Shell, Perl, Python scripts etc. It is used in Games application and Multimedia. It is used to create plugins and extensions for existing applications. Some Applications:

- **Easy learning:** The user can learn to code in scripting languages quickly, not much knowledge of web technology is required.
- **Fast editing:** It is highly efficient with the limited number of data structures and variables to use.
- **Interactivity:** It helps in adding visualization interfaces and combinations in web pages. Modern web pages demand the use of scripting languages. To create enhanced web pages, fascinated visual description which includes background and foreground colors and so on.
- **Functionality:** There are different libraries which are part of different scripting languages. They help in creating new applications in web browsers and are different from normal

programming languages.

3.2.6 Django:

Django is a python framework for web applications, as it allows developers to use modules for faster development. As a developer, you can make use of these modules to create apps, websites from an existing source. It speeds up the development process greatly, as you do not have to code everything from scratch. Using Django, you can create professional web apps and websites in a short window. The platform is known for its advanced functionality like admin panels, authentication support, comment boxes, file upload support, contact forms, app management, and more. There is a large collection of modules that you can use for Django website development projects. It is among the best framework for web development.

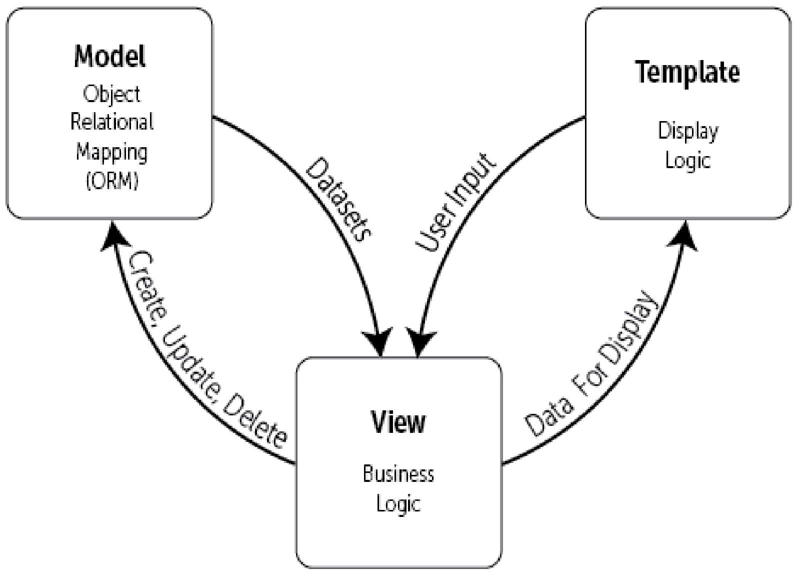


Fig 3.1 MVT architecture of Django

Advantages of Django: -

Fast: Django is very fast enough to code in. There are various libraries provided so, you don't have to reinvent the wheel or built the product from scratch. Plus, there is much pre-

built
code already available for you in Django which makes your task much easy.

Security: It comes with vast security. It can prevent your website or app from most of the attacks like – SQL Injection, XSS, CSRF, Clickjacking, and many more.

Admin Panel: It comes with the built-in administration panel, which makes development much easy and fast. As you don't have to create a separate admin panel for handling the backend.

Best IDE's for Django development: -

PyCharm: It is specifically used for Python programming, and it is developed to operate across multiple platforms, including Windows, Mac OS, and Linux. The IDE consists of code analysis tools, debugger, testing tools along with version control options. Developers can build their own Python plugins with the help of various APIs available in Pycharm. You can directly work with multiple databases from the IDE itself without getting it integrated with another tool. Some features of it:

- It includes creative code completion for classes, objects and keywords, auto-indentation and code formatting, and customizable code snippets and formats.
- It shows on-the-fly error highlighting. It also contains PEP-8 for Python that helps in writing neat codes that are easy to support for other languages.
- It has features for serving fast and safe refactoring.
- It includes a debugger for Python and JavaScript with a graphical UI. One can create and run tests with a GUI-based test runner and coding assistance.

Jupyter: It is an open-source and web-based environment to help coders that have just started off in the Data Science field. Due to ease of use along with many informative references, programmers can get acquainted with Jupyter to work with large data sets for analysis. It is available for free and can help analysts with numerical functions, data visualization, and supports many data functionalities. Jupyter contains in-built libraries like Pandas, NumPy, etc. to help coders perform various functions on data. Some features of it:

- It can support up to 40 languages, and it includes languages popular for data science such as Python, R, Scala, Julia, etc.
- It allows one to create and share the documents with equations, visualization, and most importantly live codes.
- There are interactive widgets from which code can produce outputs such as videos, images, and Latex. Not only this, but interactive widgets can also be used to visualize and manipulate data in real-time.
- It has got Big Data integration where one can take advantage of Big Data tools, such as Apache Spark, from Scala, Python, and R. One can explore the same data with libraries such as pandas, scikit-learn, ggplot2, dplyr, etc.

3.2.7 Bootstrap: Bootstrap is a web framework that focuses on simplifying the development of informative web pages. The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The end result is a uniform appearance for prose, tables and form elements across web browsers in addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight. Bootstrap also comes with several JavaScript components in the form of jQuery plug ins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto- complete function for input fields.

3.2.8 Feasibility Study

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are:

3.2.8.1 Economic Feasibility:

Development of this web portal is highly economically feasible. It is cost effective in many sense as in submitting the application form online eliminating the paper work and also visiting the website to check for any sort of information required instead of taking a brochure or flyer. The system is also time effective in two ways:

1. The website provides a complete virtual tour of the campus that can be done in seconds instead of visiting the campus for that sole purpose.
2. All the queries can be raised and answered through the question and answers section or through dropping a mail instead of calling.

3.2.8.2 Technical Feasibility:

The technical requirement for the website is economic and it uses only basic frontend development tools which is free such a s HTML, CSS, JavaScript, Bootstrap and JQuery.

And for the backend it uses the Django web framework. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it. This website does not require much updates except for updating announcements or event related information or media.

3.2.8.3 Operational Feasibility:

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. The user friendly interface is self-explanatory and guides the users in itself as to where to look what they are looking for. The addition of Question and Answer section will further help them to get a direct answer to their questions without having to search for anything.

CHAPTER 4

EXISTING AND PROPOSED SYSTEM

4.1 EXISTING SYSTEM

The existing system POD hotel are not in existing trend in India so, whenever a trip is planned, one of the big questions travelers ask themselves is where they're going to stay. They scour the internet for the best deals available and usually reach a point where they must decide between a hostel or a hotel. If you've ever been in this situation, you probably understand the struggle of wanting to stay somewhere comfortable but also wanting to save as much money as possible- especially if you've already spent a few thousand rupees on your plane ticket.

Hostels provide dormitory-like accommodations, typically with a shared living space, kitchen, bunk beds and a few bathrooms. They also have private and semiprivate rooms for a higher price, but if you go with a standard arrangement, it's very affordable. Indeed, you won't have the same privacy a hotel room provides, and you won't know exactly how comfortable or convenient it will be until you get there, but staying in a hostel will save you money and give you the opportunity to meet fellow travelers. On the other hand, staying in a hotel is a typical go-to arrangement for most travelers. It's much more predictable and, unlike at hostels, you'll have your own television, desk, room service, housekeeping and complete privacy. Depending on the hotel you choose, however, it can be about five times more expensive than a hostel.

4.2 PROPOSED SYSTEM

To overcome the drawbacks of the existing system, the proposed system has been evolved. Capsule hotels have since become a viable option for solo travelers or groups looking to budget their accommodation in India, including its capital city of Mumbai. POD hotels bring latest technology enabled cubical capsules that facilitate different facilities which is an amazing experience for travelers.

Advantages of the proposed system include:

- **Extra Cost Savings:** With prices that are often half of conventional hotels, savings are certainly guaranteed. The average price of capsule hotels ranges between INR 700 to 1000 per night. A pod bed and shower facilities are included, so when you're a travelling solo that's a good deal. If you're looking for EVEN MORE savings.

- **Convenient Location:** Most capsule hotels are just a stone's throw away from train stations, and such prime locations bring major convenience to travelers. Without having to rely on buses, taxis, or even long walks to take you to the nearest train stations, you'll be saving precious holiday time and money.

- **24-Hour Check-in:** Plus, most capsule hotels also have a 24-hour front desk service to assist you even if you check in during the wee hours. This is especially important if you're saving money by flying budget airlines with odd timings – you don't have to worry about missing the check-in period at all!

- **Hygienic Beds & Bathrooms:** As travelers are required to check out every day, each capsule hotel pod is cleaned and sterilized daily. Shared showering facilities are also cleaned regularly, and individually-packed soap and towel sets are just some of the free things you can take from the hotel.

CHAPTER 5

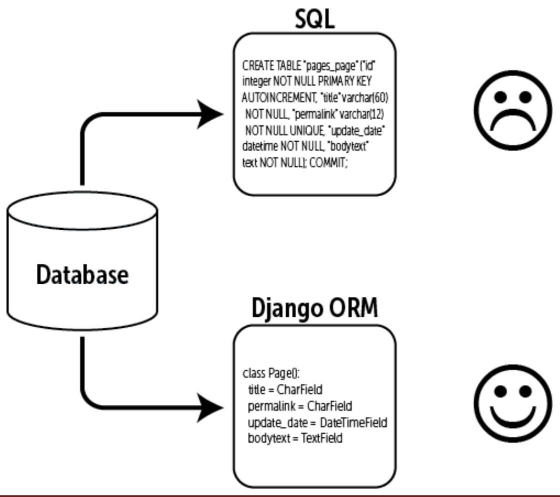
IMPLEMENTATION

Django uses slightly different terminology in its implementation of MVC. In Django:

The **model** is functionally the same. Django’s Object-Relational Mapping (ORM—more on the ORM later) provides the interface to the application database;
The **template** provides display logic and is the interface between the user and your Django application; and
The **view** manages the bulk of the applications data processing, application logic and messaging.

Django’s models provide an Object-relational Mapping (ORM) to the underlying database. ORM is a powerful programming technique that makes working with data and relational databases much easier. Most common databases are programmed with some form of Structured Query Language (SQL), however each database implements SQL in its own way. SQL can be quite complex and difficult to learn. An ORM tool on the other hand, provides a simple mapping between an *object* (the ‘O’ in ORM) and the underlying database, without the programmer needing to know the database structure, or requiring complex SQL to manipulate and retrieve data

Fig 5.1 Django ORM



```
CREATE TABLE "pages_page" ("id"
integer NOT NULL PRIMARY KEY
AUTOINCREMENT, "title" varchar(60)
 NOT NULL, "permalink" varchar(12)
 NOT NULL UNIQUE, "update_date"
datetime NOT NULL, "bodytext"
text NOT NULL); COMMIT;
```

```
class Page():
  title = CharField
  permalink = CharField
  update_date = DateTimeField
  bodytext = TextField
```

Django’s models provide an Object-relational Mapping (ORM) to the underlying database. ORM is a powerful programming technique that makes working with data and relational databases much easier.

Most common databases are programmed with some form of Structured Query Language (SQL), however each database implements SQL in its own way. SQL can be quite complex and difficult to learn. An ORM tool on the other hand, provides a simple mapping between an *object* (the ‘O’ in ORM) and the underlying database, without the programmer needing to know the database structure, or requiring complex SQL to manipulate and retrieve data (Figure 2).

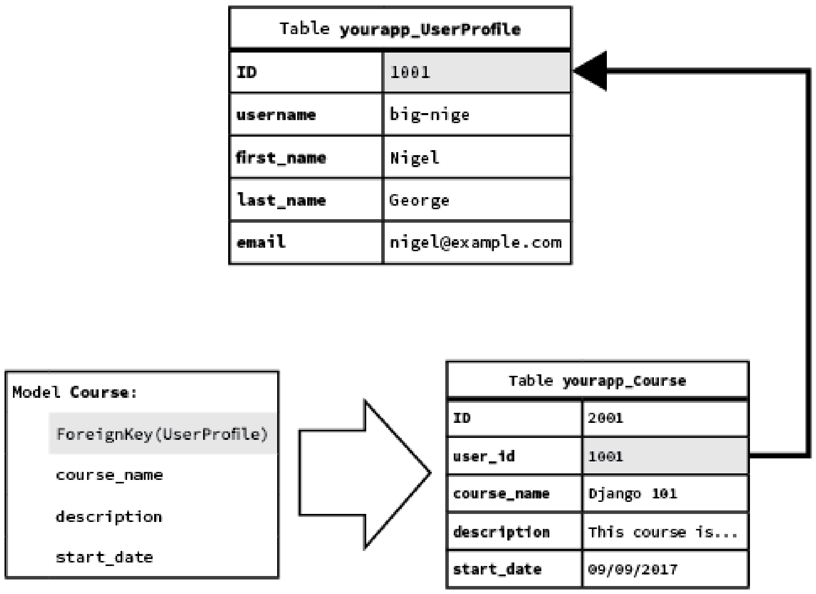


Table yourapp_UserProfile	
ID	1001
username	big-nige
first_name	Nigel
last_name	George
email	nigel@example.com

```
Model Course:
    ForeignKey(UserProfile)
    course_name
    description
    start_date
```

Table yourapp_Course	
ID	2001
user_id	1001
course_name	Django 101
description	This course is...
start_date	09/09/2017

Fig 5.2 Foreign key links in Django models

A Django template is a text file designed to separate an application’s data from the way it is presented. In most cases, Django templates are Hypertext Markup Language (HTML) files for presenting application data in a web browser, however Django templates are not limited to HTML—they can be used for rendering several different text formats.

The design of Django's templates is based on several core principles, however three are key:

1. A template system should separate program logic from design.

2. Templates should discourage redundancy—Don't Repeat Yourself (DRY).

3. The template system should be safe and secure—code execution in the template must be forbidden.

Django's views are the information brokers of a Django application. A view sources data from your database (or an external data source or service) and delivers it to a template. The view makes decisions on what data gets delivered to the template—either by acting on input from the user, or in response to other business logic and internal processes.

Each Django view performs a specific function, and has an associated template. Views are represented by either a Python function, or a method of a Python class. In the early days of Django, there were only function-based views, however as Django has grown over the years, Django's developers added class-based views to Django.

Class-based views add extensibility to Django's views, as well as built-in views that make creating common views (like displaying a list of articles) easier to implement. Don't worry too much about the differences between function- and class-based views.

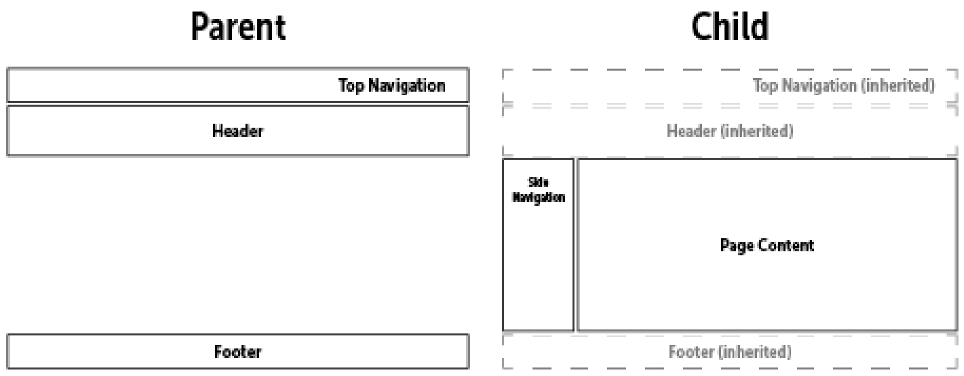


Fig 5.3 All common elements are inherited from the parent template(base)

5.1 E-R Diagram

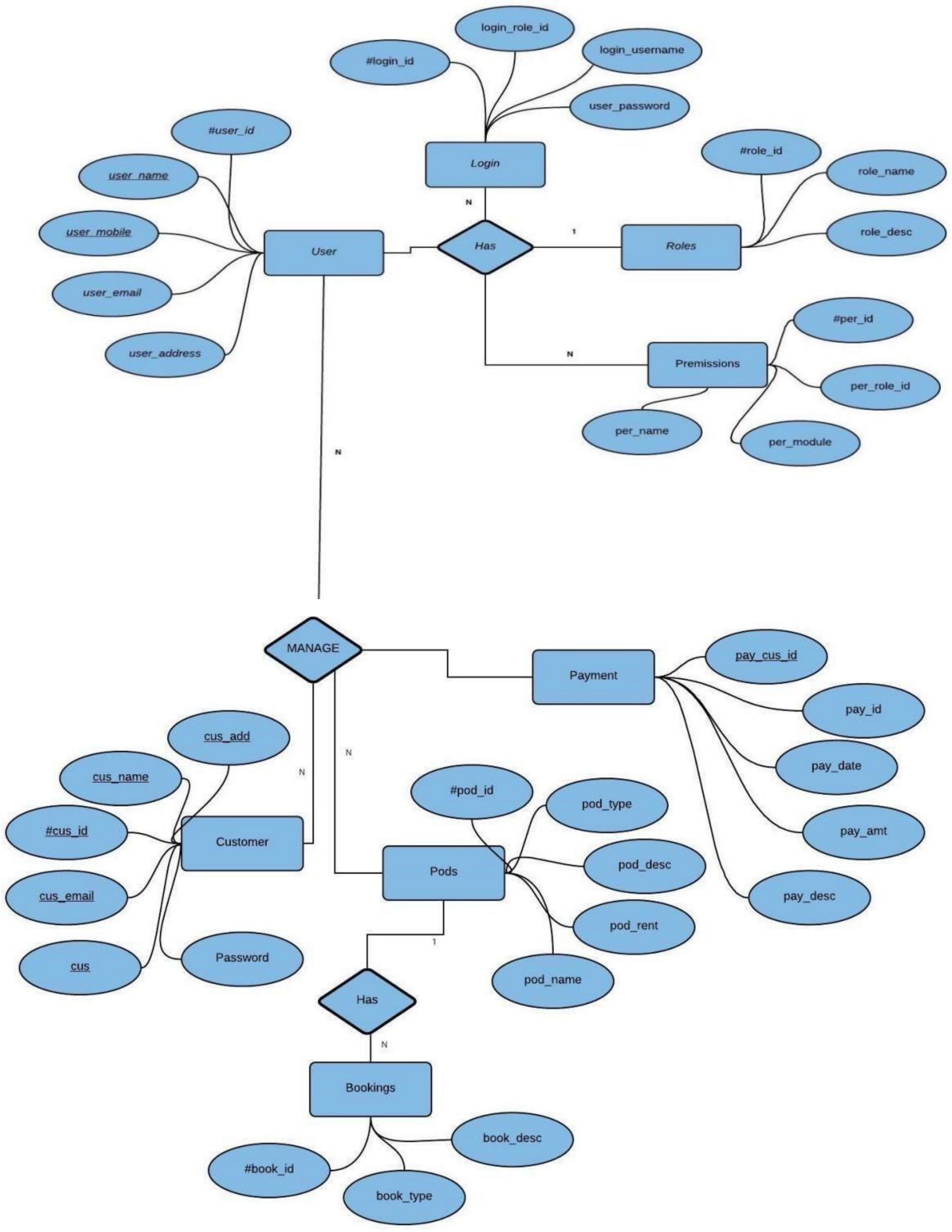


Fig 5.4 E-R Diagram

5.2 Flow Diagram

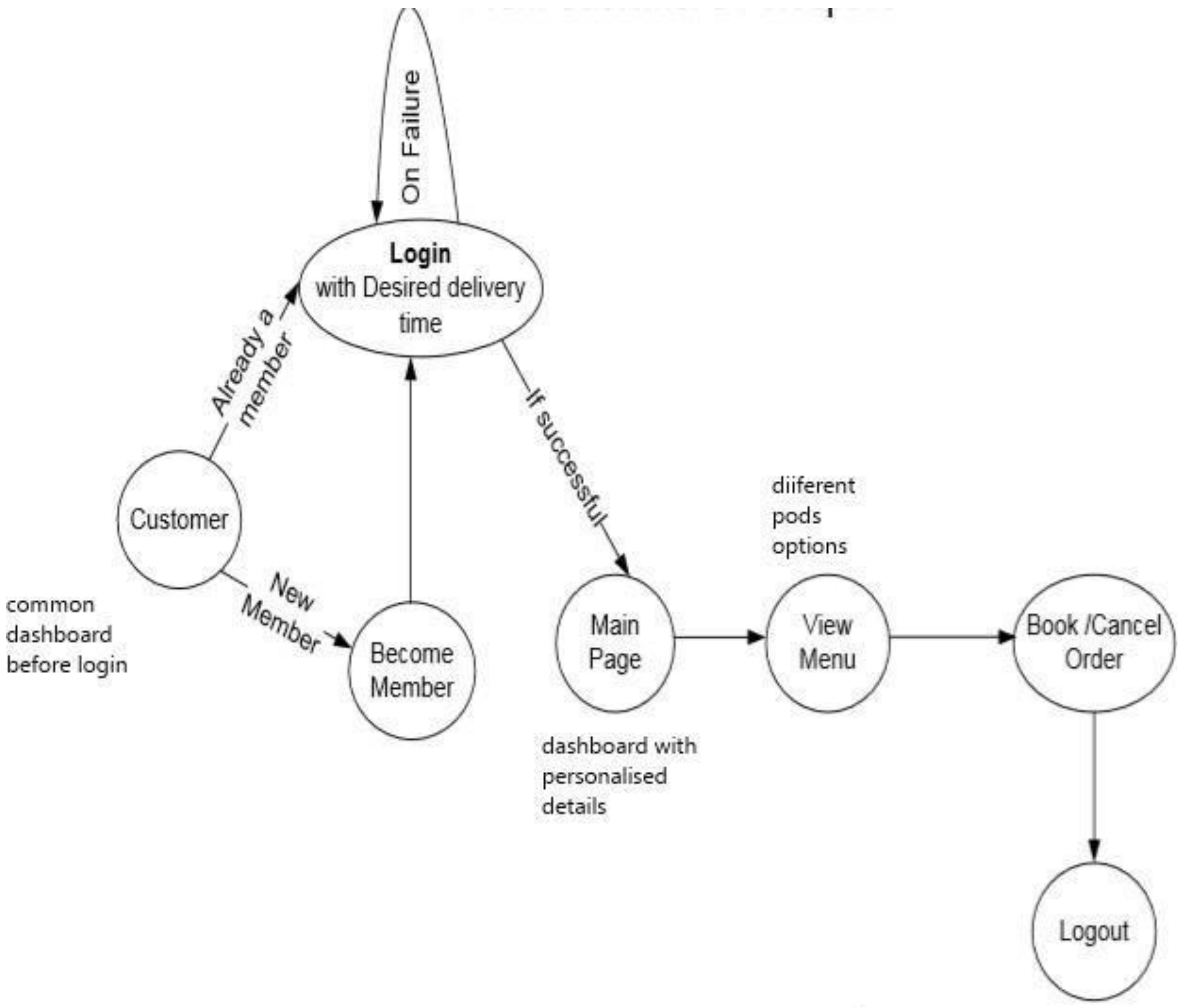
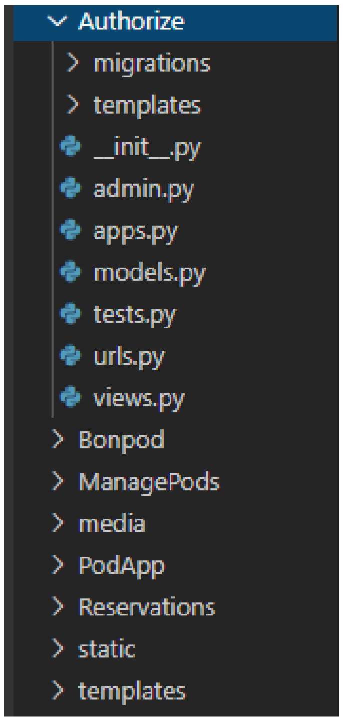
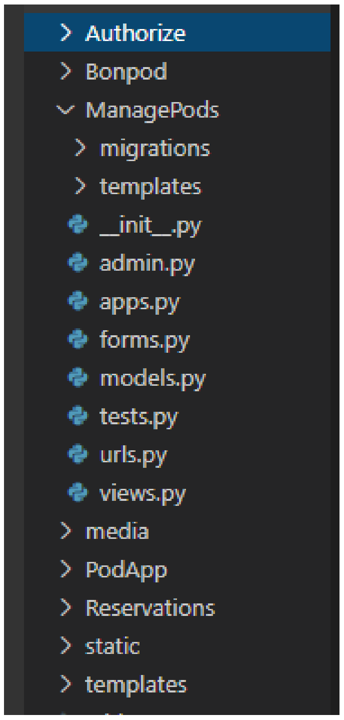


Fig 5.5 Flow Diagram

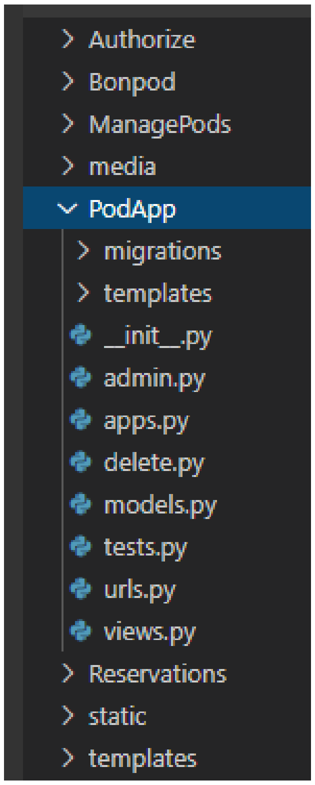
5.3 Modules in Bonpod



```
v Authorize
    > migrations
    > templates
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      urls.py
      views.py
  > Bonpod
  > ManagePods
  > media
  > PodApp
  > Reservations
  > static
  > templates
```



```
> Authorize
> Bonpod
v ManagePods
    > migrations
    > templates
      __init__.py
      admin.py
      apps.py
      forms.py
      models.py
      tests.py
      urls.py
      views.py
> media
> PodApp
> Reservations
> static
> templates
```



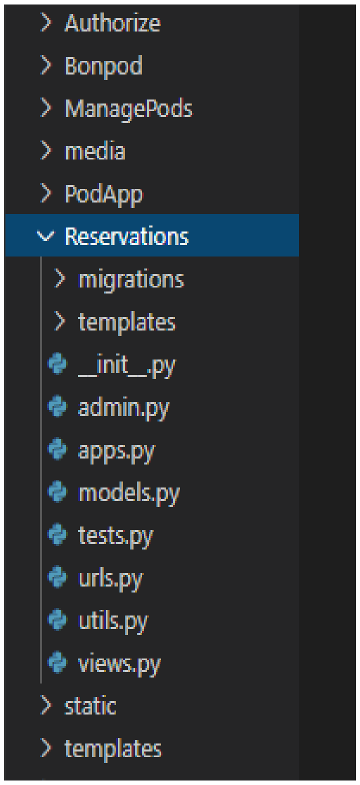
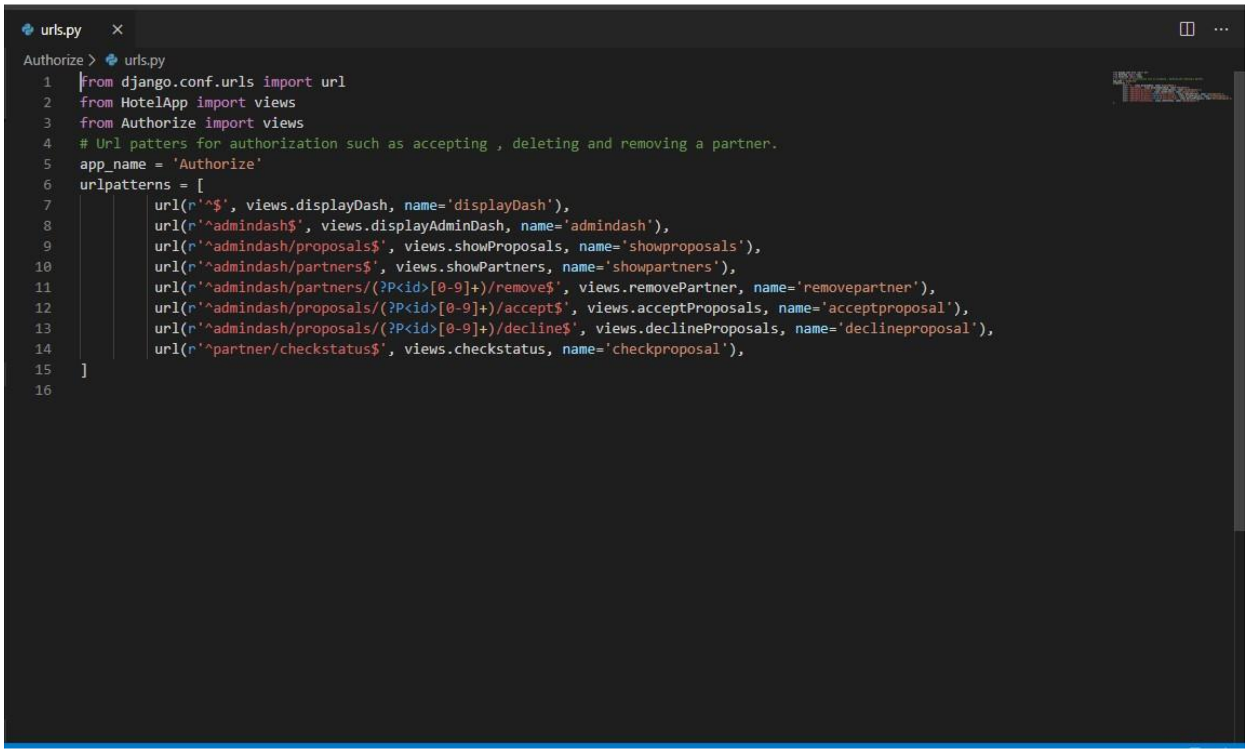
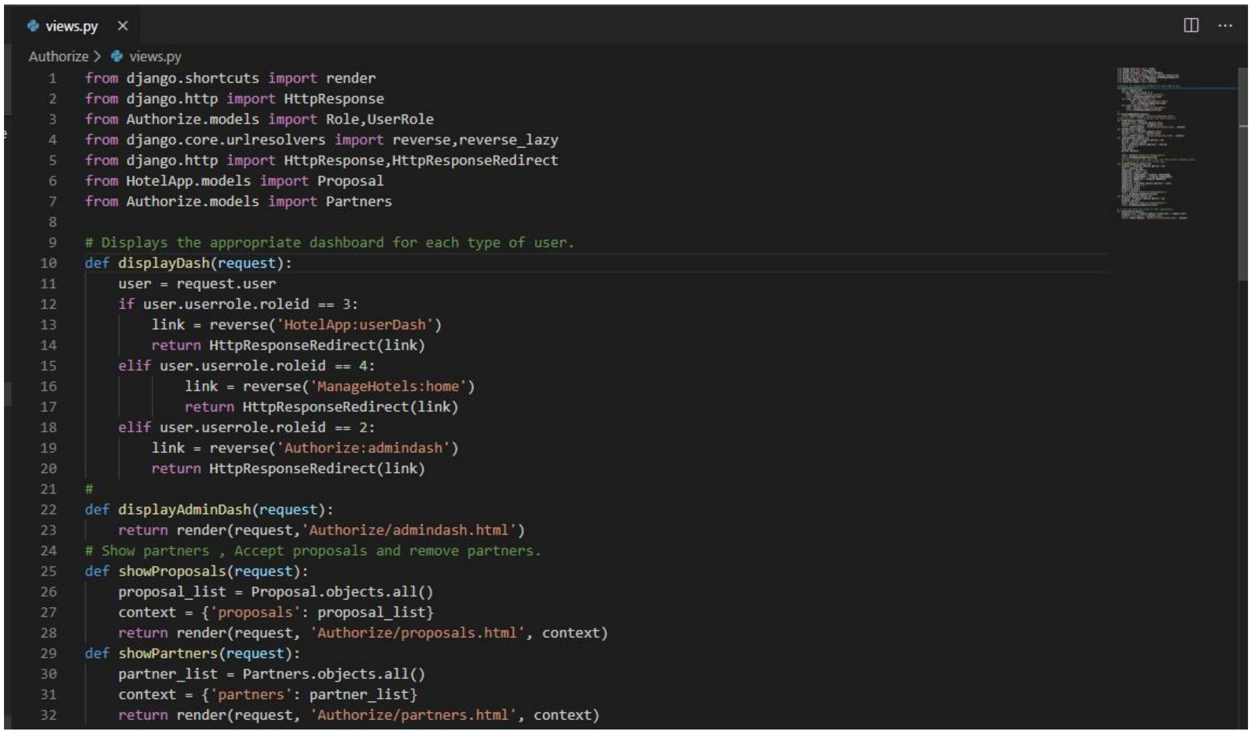


Fig5.6 All modules

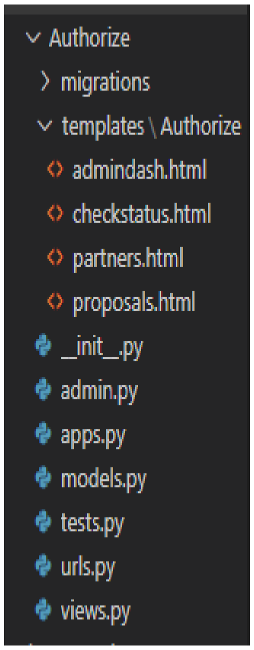
5.4 Authorize Module code

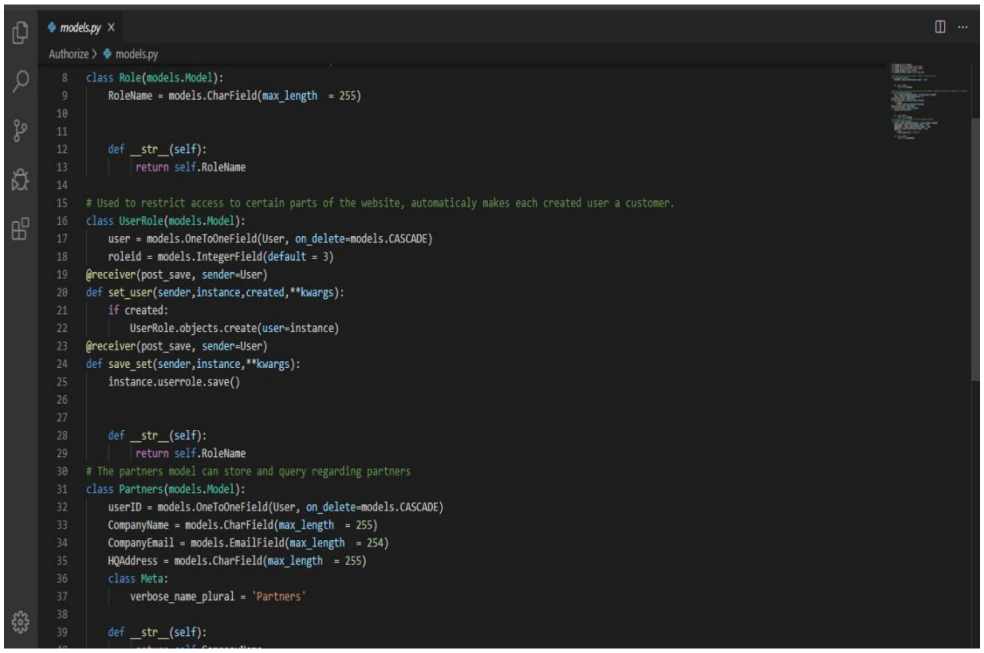


```python
from django.conf.urls import url
from HotelApp import views
from Authorize import views
# Url patters for authorization such as accepting , deleting and removing a partner.
app_name = 'Authorize'
urlpatterns = [
        url(r'^$', views.displayDash, name='displayDash'),
        url(r'^admindash$', views.displayAdminDash, name='admindash'),
        url(r'^admindash/proposals$', views.showProposals, name='showproposals'),
        url(r'^admindash/partners$', views.showPartners, name='showpartners'),
        url(r'^admindash/partners/(?P<id>[0-9]+)/remove$', views.removePartner, name='removepartner'),
        url(r'^admindash/proposals/(?P<id>[0-9]+)/accept$', views.acceptProposals, name='acceptproposal'),
        url(r'^admindash/proposals/(?P<id>[0-9]+)/decline$', views.declineProposals, name='declineproposal'),
        url(r'^partner/checkstatus$', views.checkstatus, name='checkproposal'),
]
```



```python
from django.shortcuts import render
from django.http import HttpResponse
from Authorize.models import Role,UserRole
from django.core.urlresolvers import reverse,reverse_lazy
from django.http import HttpResponse,HttpResponseRedirect
from HotelApp.models import Proposal
from Authorize.models import Partners

# Displays the appropriate dashboard for each type of user.
def displayDash(request):
    user = request.user
    if user.userrole.roleid == 3:
        link = reverse('HotelApp:userDash')
        return HttpResponseRedirect(link)
    elif user.userrole.roleid == 4:
            link = reverse('ManageHotels:home')
            return HttpResponseRedirect(link)
    elif user.userrole.roleid == 2:
        link = reverse('Authorize:admindash')
        return HttpResponseRedirect(link)
#
def displayAdminDash(request):
    return render(request,'Authorize/admindash.html')
# Show partners , Accept proposals and remove partners.
def showProposals(request):
    proposal_list = Proposal.objects.all()
    context = {'proposals': proposal_list}
    return render(request, 'Authorize/proposals.html', context)
def showPartners(request):
    partner_list = Partners.objects.all()
    context = {'partners': partner_list}
    return render(request, 'Authorize/partners.html', context)
```

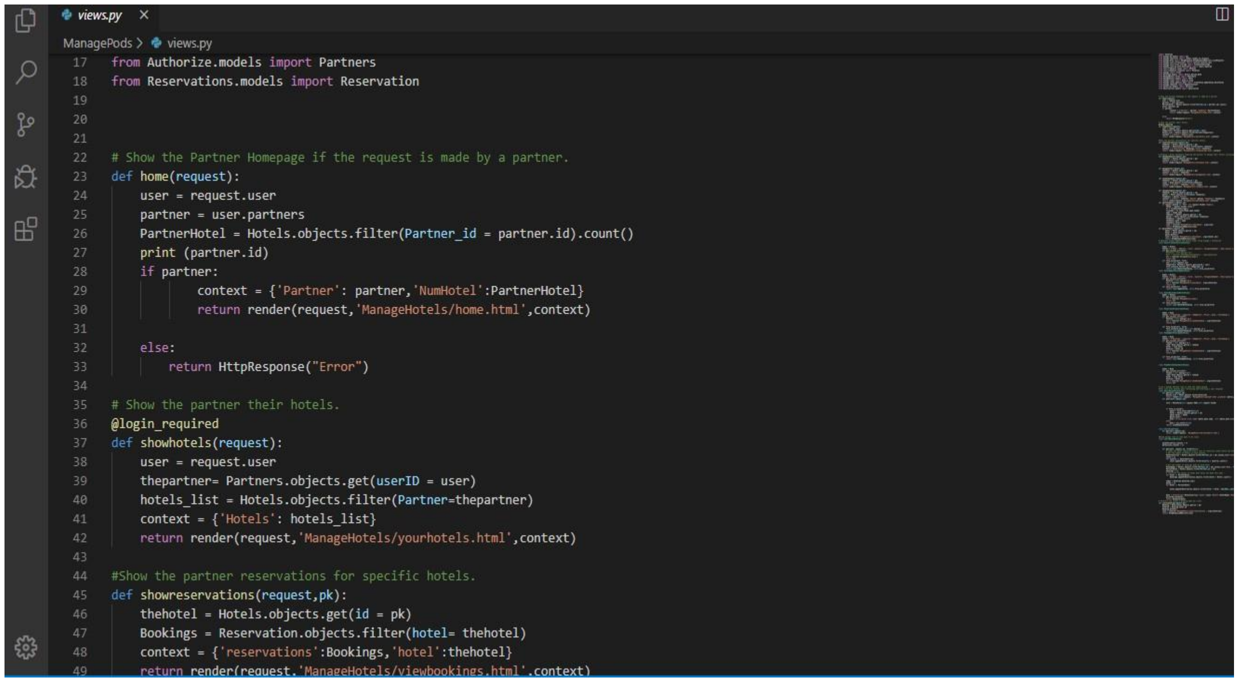




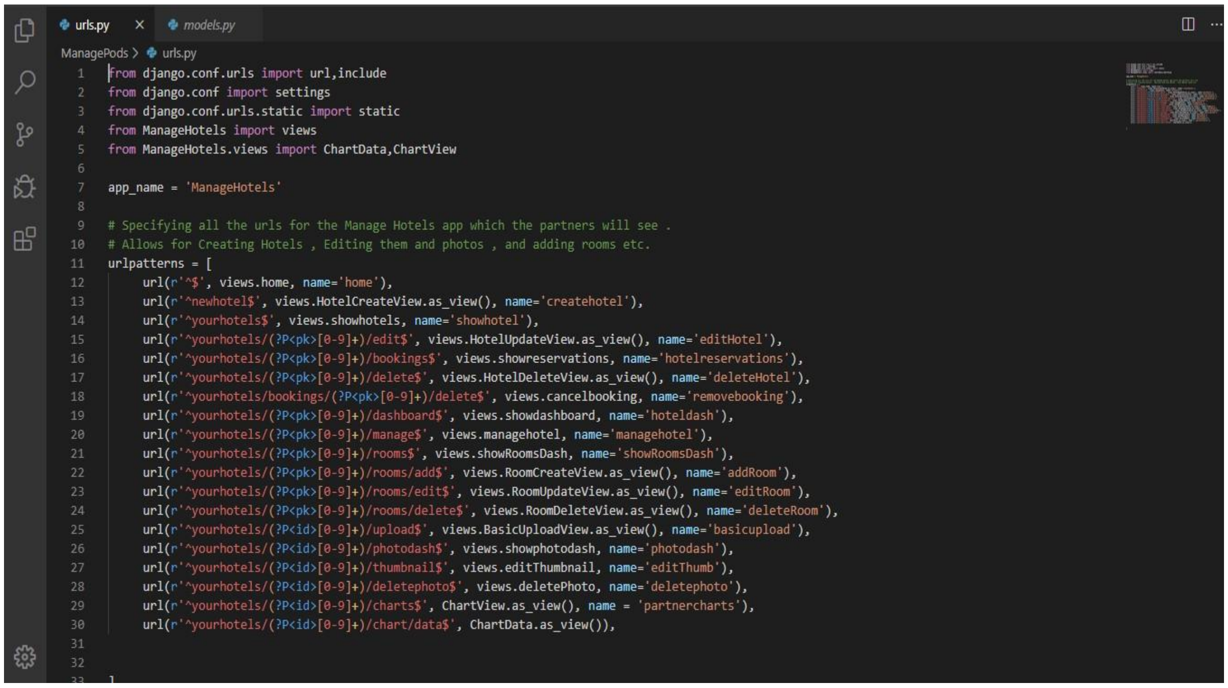
```python
class Role(models.Model):
    RoleName = models.CharField(max_length  = 255)


    def __str__(self):
        return self.RoleName

# Used to restrict access to certain parts of the website, automaticaly makes each created user a customer.
class UserRole(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    roleid = models.IntegerField(default = 3)
@receiver(post_save, sender=User)
def set_user(sender,instance,created,**kwargs):
    if created:
        UserRole.objects.create(user=instance)
@receiver(post_save, sender=User)
def save_set(sender,instance,**kwargs):
    instance.userrole.save()


    def __str__(self):
        return self.RoleName
# The partners model can store and query regarding partners
class Partners(models.Model):
    userID = models.OneToOneField(User, on_delete=models.CASCADE)
    CompanyName = models.CharField(max_length  = 255)
    CompanyEmail = models.EmailField(max_length  = 254)
    HQAddress = models.CharField(max_length  = 255)
    class Meta:
        verbose_name_plural = 'Partners'

    def __str__(self):
```

Fig5.7 Authorize module files and code

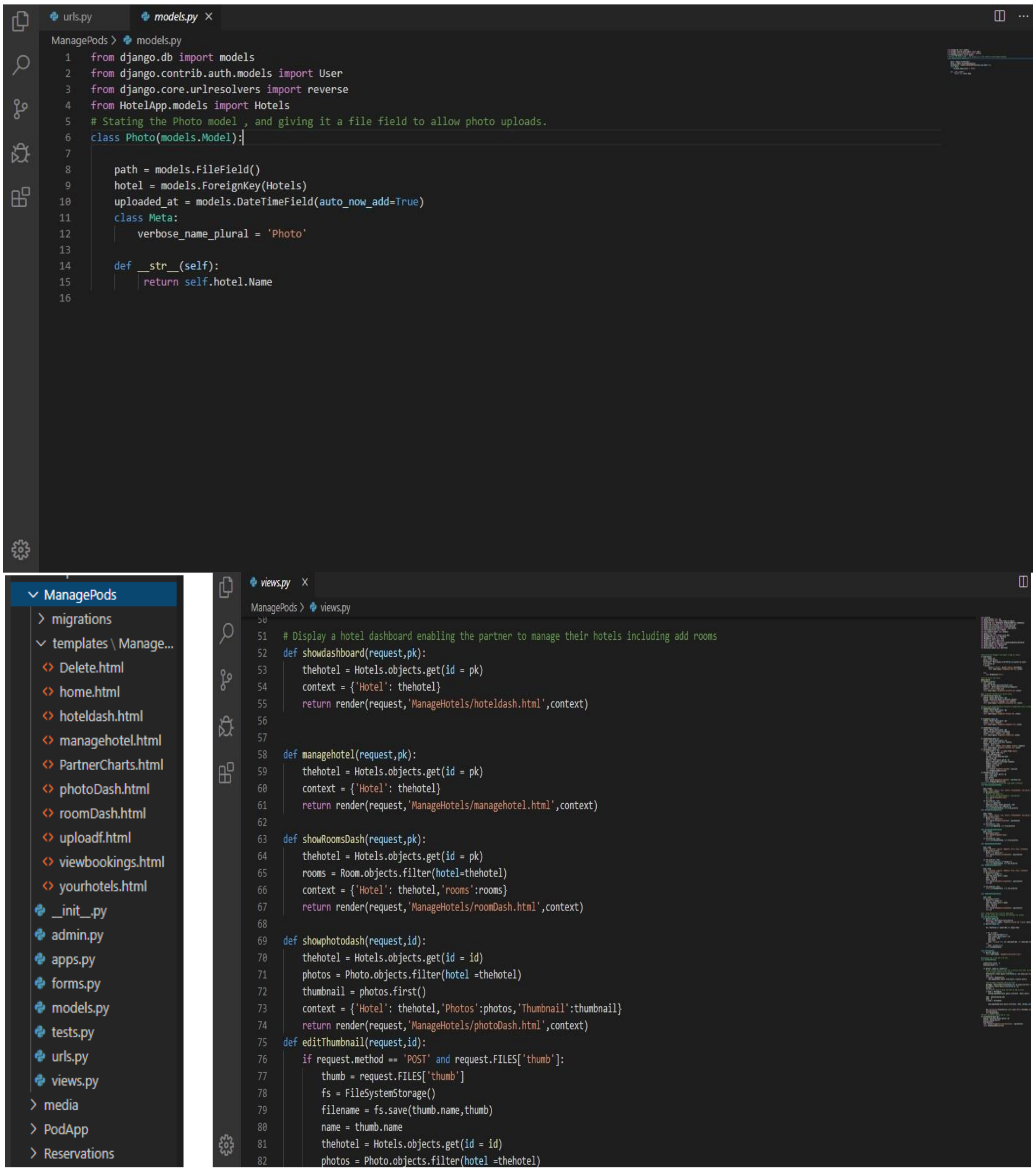
5.5 Pod Module Code


views.py
ManagePods > views.py

```python
from Authorize.models import Partners
from Reservations.models import Reservation



# Show the Partner Homepage if the request is made by a partner.
def home(request):
    user = request.user
    partner = user.partners
    PartnerHotel = Hotels.objects.filter(Partner_id = partner.id).count()
    print (partner.id)
    if partner:
            context = {'Partner': partner,'NumHotel':PartnerHotel}
            return render(request,'ManageHotels/home.html',context)

    else:
        return HttpResponse("Error")

# Show the partner their hotels.
@login_required
def showhotels(request):
    user = request.user
    thepartner= Partners.objects.get(userID = user)
    hotels_list = Hotels.objects.filter(Partner=thepartner)
    context = {'Hotels': hotels_list}
    return render(request,'ManageHotels/yourhotels.html',context)

#Show the partner reservations for specific hotels.
def showreservations(request,pk):
    thehotel = Hotels.objects.get(id = pk)
    Bookings = Reservation.objects.filter(hotel= thehotel)
    context = {'reservations':Bookings,'hotel':thehotel}
    return render(request,'ManageHotels/viewbookings.html',context)
```


urls.py models.py
ManagePods > urls.py

```python
from django.conf.urls import url,include
from django.conf import settings
from django.conf.urls.static import static
from ManageHotels import views
from ManageHotels.views import ChartData,ChartView

app_name = 'ManageHotels'

# Specifying all the urls for the Manage Hotels app which the partners will see .
# Allows for Creating Hotels , Editing them and photos , and adding rooms etc.
urlpatterns = [
     url(r'^$', views.home, name='home'),
     url(r'^newhotel$', views.HotelCreateView.as_view(), name='createhotel'),
     url(r'^yourhotels$', views.showhotels, name='showhotel'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/edit$', views.HotelUpdateView.as_view(), name='editHotel'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/bookings$', views.showreservations, name='hotelreservations'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/delete$', views.HotelDeleteView.as_view(), name='deleteHotel'),
     url(r'^yourhotels/bookings/(?P<pk>[0-9]+)/delete$', views.cancelbooking, name='removebooking'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/dashboard$', views.showdashboard, name='hoteldash'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/manage$', views.managehotel, name='managehotel'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/rooms$', views.showRoomsDash, name='showRoomsDash'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/rooms/add$', views.RoomCreateView.as_view(), name='addRoom'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/rooms/edit$', views.RoomUpdateView.as_view(), name='editRoom'),
     url(r'^yourhotels/(?P<pk>[0-9]+)/rooms/delete$', views.RoomDeleteView.as_view(), name='deleteRoom'),
     url(r'^yourhotels/(?P<id>[0-9]+)/upload$', views.BasicUploadView.as_view(), name='basicupload'),
     url(r'^yourhotels/(?P<id>[0-9]+)/photodash$', views.showphotodash, name='photodash'),
     url(r'^yourhotels/(?P<id>[0-9]+)/thumbnail$', views.editThumbnail, name='editThumb'),
     url(r'^yourhotels/(?P<id>[0-9]+)/deletephoto$', views.deletePhoto, name='deletephoto'),
     url(r'^yourhotels/(?P<id>[0-9]+)/charts$', ChartView.as_view(), name = 'partnercharts'),
     url(r'^yourhotels/(?P<id>[0-9]+)/chart/data$', ChartData.as_view()),


]
```



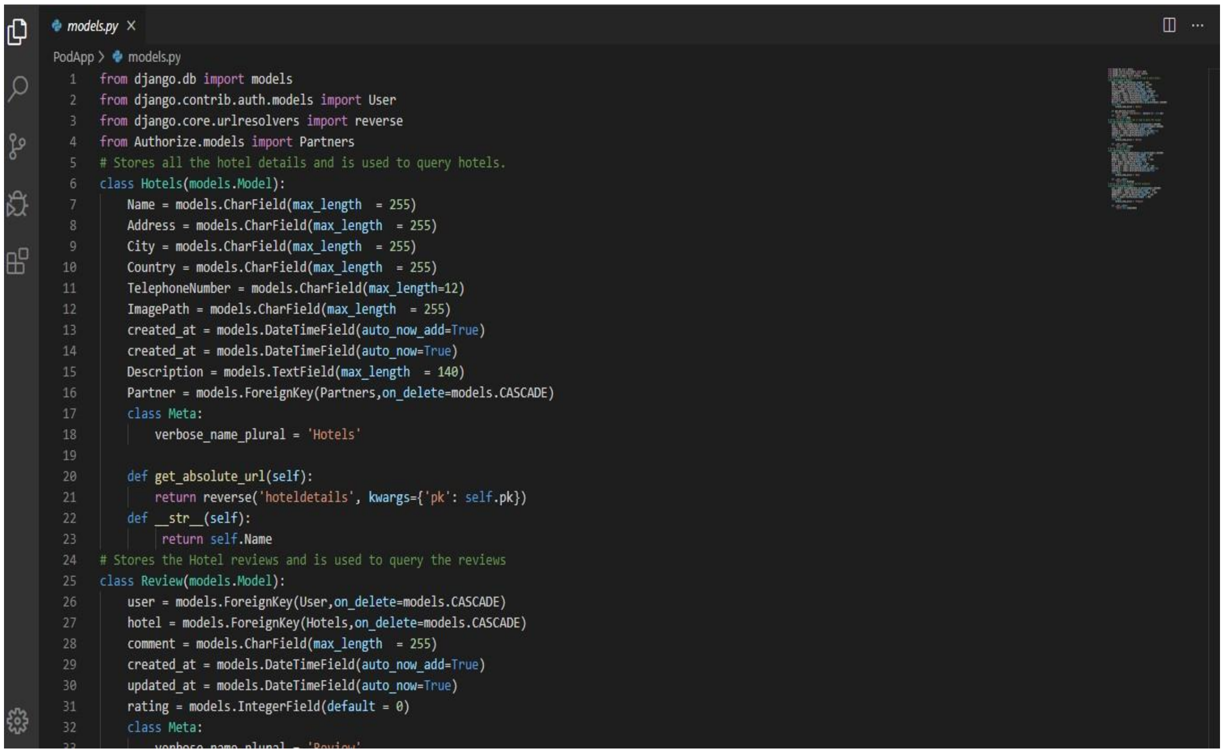
```python
from django.db import models
from django.contrib.auth.models import User
from django.core.urlresolvers import reverse
from HotelApp.models import Hotels
# Stating the Photo model , and giving it a file field to allow photo uploads.
class Photo(models.Model):

    path = models.FileField()
    hotel = models.ForeignKey(Hotels)
    uploaded_at = models.DateTimeField(auto_now_add=True)
    class Meta:
        verbose_name_plural = 'Photo'

    def __str__(self):
        return self.hotel.Name

```

```
ManagePods
  migrations
  templates \ Manage...
    Delete.html
    home.html
    hoteldash.html
    managehotel.html
    PartnerCharts.html
    photoDash.html
    roomDash.html
    uploadf.html
    viewbookings.html
    yourhotels.html
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  tests.py
  urls.py
  views.py
media
PodApp
Reservations
```

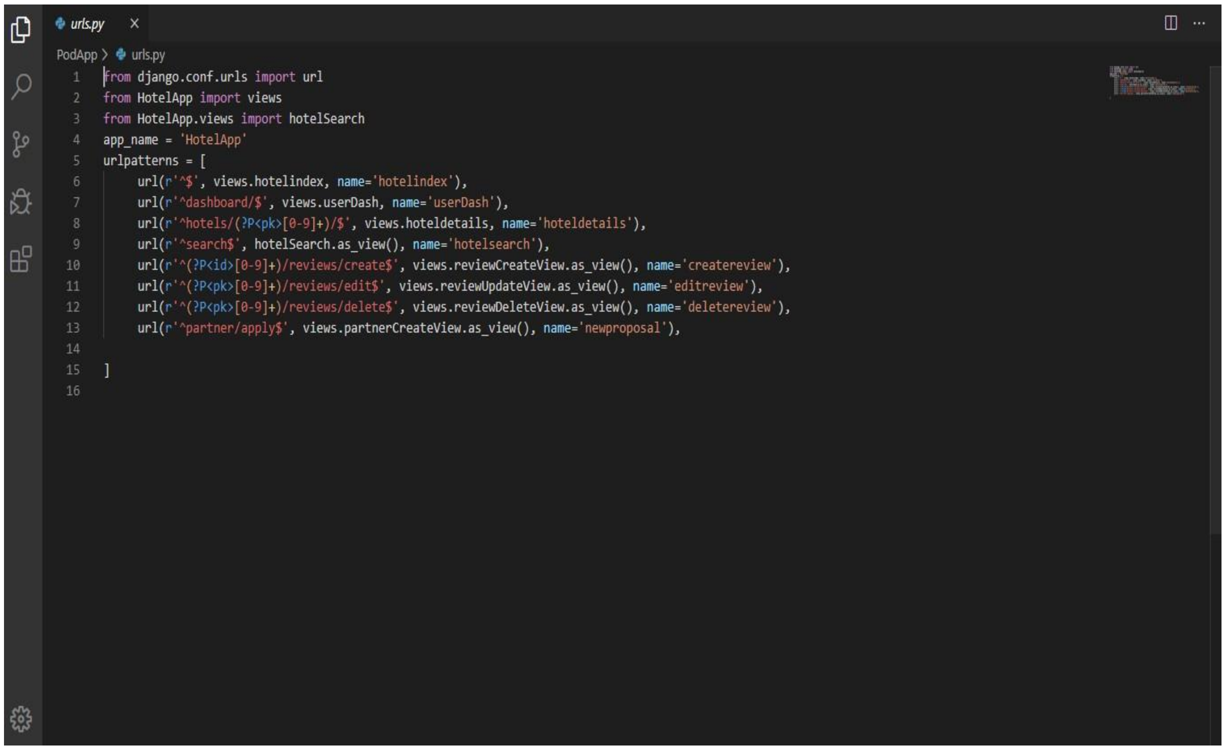
```python
# Display a hotel dashboard enabling the partner to manage their hotels including add rooms
def showdashboard(request,pk):
    thehotel = Hotels.objects.get(id = pk)
    context = {'Hotel': thehotel}
    return render(request,'ManageHotels/hoteldash.html',context)


def managehotel(request,pk):
    thehotel = Hotels.objects.get(id = pk)
    context = {'Hotel': thehotel}
    return render(request,'ManageHotels/managehotel.html',context)

def showRoomsDash(request,pk):
    thehotel = Hotels.objects.get(id = pk)
    rooms = Room.objects.filter(hotel=thehotel)
    context = {'Hotel': thehotel,'rooms':rooms}
    return render(request,'ManageHotels/roomDash.html',context)

def showphotodash(request,id):
    thehotel = Hotels.objects.get(id = id)
    photos = Photo.objects.filter(hotel =thehotel)
    thumbnail = photos.first()
    context = {'Hotel': thehotel,'Photos':photos,'Thumbnail':thumbnail}
    return render(request,'ManageHotels/photoDash.html',context)
def editThumbnail(request,id):
    if request.method == 'POST' and request.FILES['thumb']:
        thumb = request.FILES['thumb']
        fs = FileSystemStorage()
        filename = fs.save(thumb.name,thumb)
        name = thumb.name
        thehotel = Hotels.objects.get(id = id)
        photos = Photo.objects.filter(hotel =thehotel)
```

Fig5.8 Pod module files and code

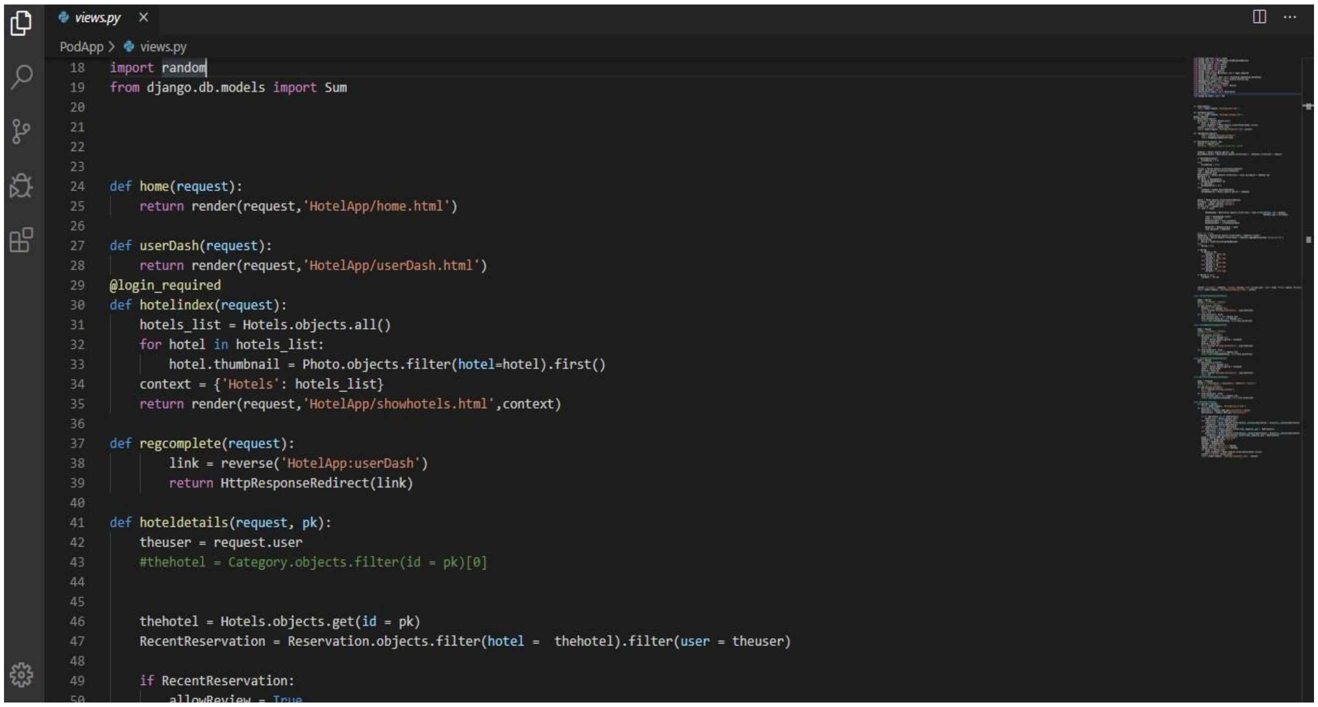
5.6 Pod App Module Code



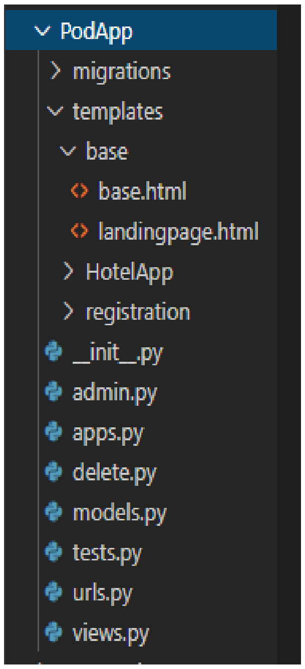
```python
from django.db import models
from django.contrib.auth.models import User
from django.core.urlresolvers import reverse
from Authorize.models import Partners
# Stores all the hotel details and is used to query hotels.
class Hotels(models.Model):
    Name = models.CharField(max_length  = 255)
    Address = models.CharField(max_length  = 255)
    City = models.CharField(max_length  = 255)
    Country = models.CharField(max_length  = 255)
    TelephoneNumber = models.CharField(max_length=12)
    ImagePath = models.CharField(max_length  = 255)
    created_at = models.DateTimeField(auto_now_add=True)
    created_at = models.DateTimeField(auto_now=True)
    Description = models.TextField(max_length  = 140)
    Partner = models.ForeignKey(Partners,on_delete=models.CASCADE)
    class Meta:
        verbose_name_plural = 'Hotels'

    def get_absolute_url(self):
        return reverse('hoteldetails', kwargs={'pk': self.pk})
    def __str__(self):
         return self.Name
# Stores the Hotel reviews and is used to query the reviews
class Review(models.Model):
    user = models.ForeignKey(User,on_delete=models.CASCADE)
    hotel = models.ForeignKey(Hotels,on_delete=models.CASCADE)
    comment = models.CharField(max_length  = 255)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    rating = models.IntegerField(default = 0)
    class Meta:
        verbose_name_plural = 'Review'
```

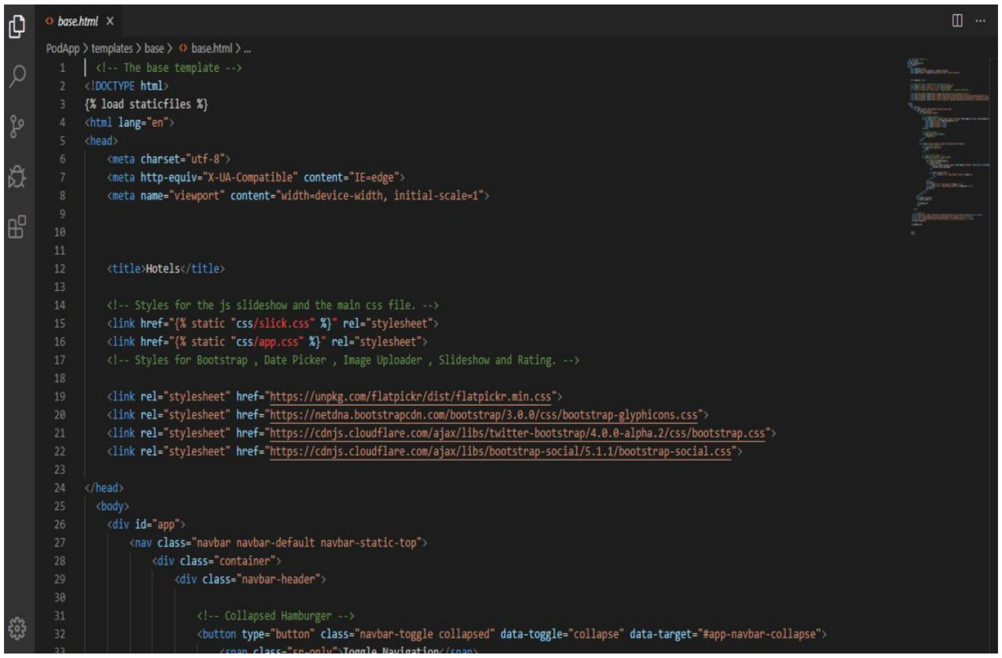


```python
from django.conf.urls import url
from HotelApp import views
from HotelApp.views import hotelSearch
app_name = 'HotelApp'
urlpatterns = [
     url(r'^$', views.hotelindex, name='hotelindex'),
     url(r'^dashboard/$', views.userDash, name='userDash'),
     url(r'^hotels/(?P<pk>[0-9]+)/$', views.hoteldetails, name='hoteldetails'),
     url(r'^search$', hotelSearch.as_view(), name='hotelsearch'),
     url(r'^(?P<id>[0-9]+)/reviews/create$', views.reviewCreateView.as_view(), name='createreview'),
     url(r'^(?P<pk>[0-9]+)/reviews/edit$', views.reviewUpdateView.as_view(), name='editreview'),
     url(r'^(?P<pk>[0-9]+)/reviews/delete$', views.reviewDeleteView.as_view(), name='deletereview'),
     url(r'^partner/apply$', views.partnerCreateView.as_view(), name='newproposal'),

]
```



```python
import random
from django.db.models import Sum




def home(request):
    return render(request,'HotelApp/home.html')

def userDash(request):
    return render(request,'HotelApp/userDash.html')
@login_required
def hotelindex(request):
    hotels_list = Hotels.objects.all()
    for hotel in hotels_list:
        hotel.thumbnail = Photo.objects.filter(hotel=hotel).first()
    context = {'Hotels': hotels_list}
    return render(request,'HotelApp/showhotels.html',context)

def regcomplete(request):
        link = reverse('HotelApp:userDash')
        return HttpResponseRedirect(link)

def hoteldetails(request, pk):
    theuser = request.user
    #thehotel = Category.objects.filter(id = pk)[0]


    thehotel = Hotels.objects.get(id = pk)
    RecentReservation = Reservation.objects.filter(hotel =  thehotel).filter(user = theuser)

    if RecentReservation:
        allowReview = True
```

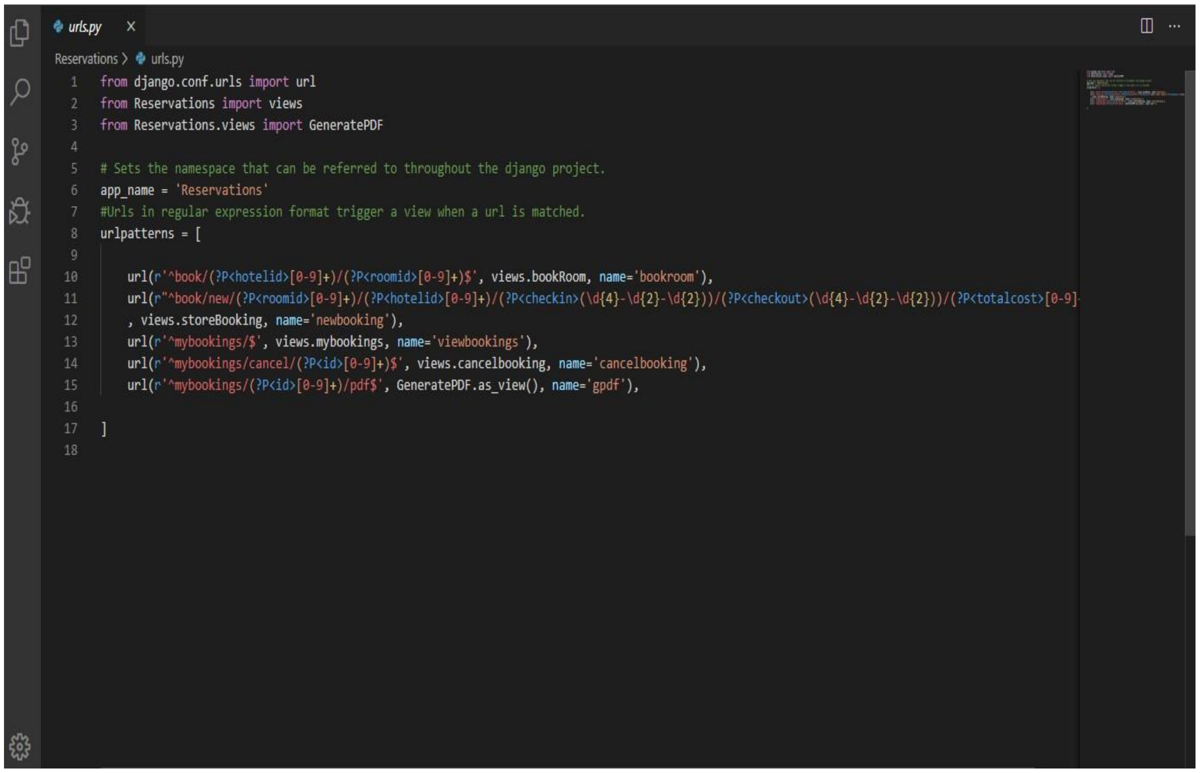




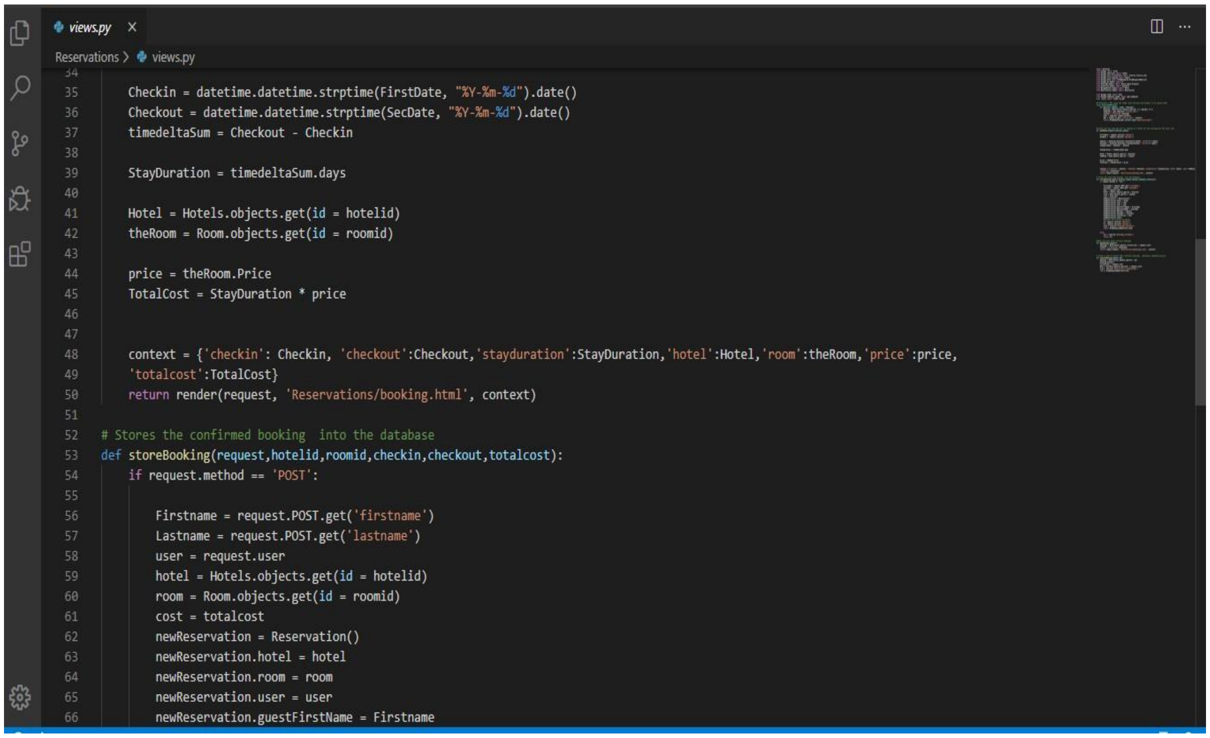
```html
<!-- The base template -->
<!DOCTYPE html>
{% load staticfiles %}
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">



    <title>Hotels</title>

    <!-- Styles for the js slideshow and the main css file. -->
    <link href="{% static "css/slick.css" %}" rel="stylesheet">
    <link href="{% static "css/app.css" %}" rel="stylesheet">
    <!-- Styles for Bootstrap , Date Picker , Image Uploader , Slideshow and Rating. -->

    <link rel="stylesheet" href="https://unpkg.com/flatpickr/dist/flatpickr.min.css">
    <link rel="stylesheet" href="https://netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap-glyphicons.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha.2/css/bootstrap.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-social/5.1.1/bootstrap-social.css">

</head>
  <body>
    <div id="app">
        <nav class="navbar navbar-default navbar-static-top">
            <div class="container">
                <div class="navbar-header">

                    <!-- Collapsed Hamburger -->
                    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#app-navbar-collapse">
                        <span class="sr-only">Toggle Navigation</span>
```

Fig5.9 Pod App module files and code

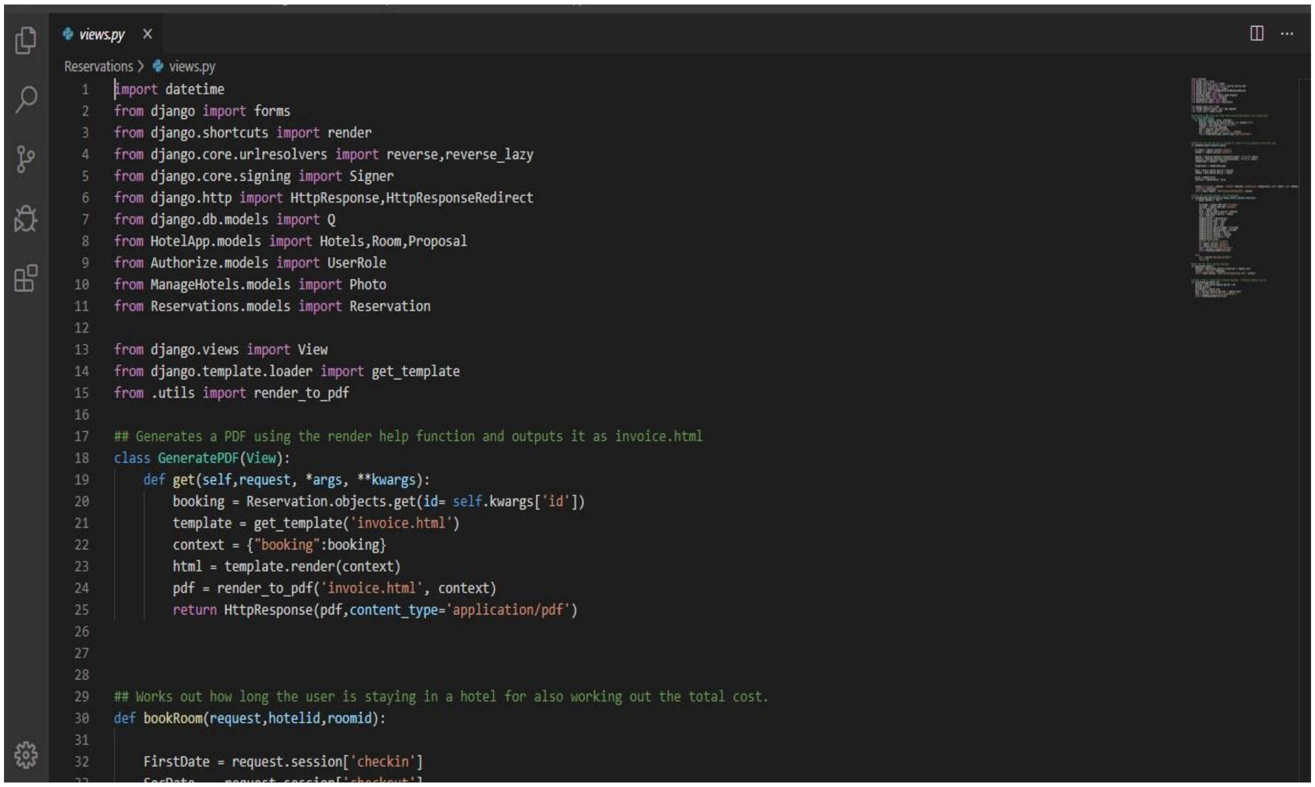
5.7 Reservation Module code



```python
from django.conf.urls import url
from Reservations import views
from Reservations.views import GeneratePDF

# Sets the namespace that can be referred to throughout the django project.
app_name = 'Reservations'
#Urls in regular expression format trigger a view when a url is matched.
urlpatterns = [

    url(r'^book/(?P<hotelid>[0-9]+)/(?P<roomid>[0-9]+)$', views.bookRoom, name='bookroom'),
    url(r"^book/new/(?P<roomid>[0-9]+)/(?P<hotelid>[0-9]+)/(?P<checkin>(\d{4}-\d{2}-\d{2}))/(?P<checkout>(\d{4}-\d{2}-\d{2}))/(?P<totalcost>[0-9]-
    , views.storeBooking, name='newbooking'),
    url(r'^mybookings/$', views.mybookings, name='viewbookings'),
    url(r'^mybookings/cancel/(?P<id>[0-9]+)$', views.cancelbooking, name='cancelbooking'),
    url(r'^mybookings/(?P<id>[0-9]+)/pdf$', GeneratePDF.as_view(), name='gpdf'),

]

```



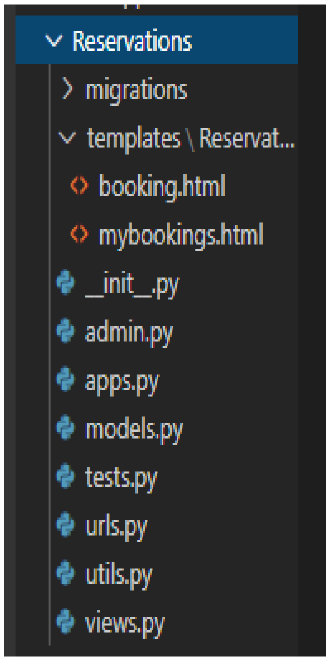
```python

        Checkin = datetime.datetime.strptime(FirstDate, "%Y-%m-%d").date()
        Checkout = datetime.datetime.strptime(SecDate, "%Y-%m-%d").date()
        timedeltaSum = Checkout - Checkin

        StayDuration = timedeltaSum.days

        Hotel = Hotels.objects.get(id = hotelid)
        theRoom = Room.objects.get(id = roomid)

        price = theRoom.Price
        TotalCost = StayDuration * price


        context = {'checkin': Checkin, 'checkout':Checkout,'stayduration':StayDuration,'hotel':Hotel,'room':theRoom,'price':price,
        'totalcost':TotalCost}
        return render(request, 'Reservations/booking.html', context)

# Stores the confirmed booking  into the database
def storeBooking(request,hotelid,roomid,checkin,checkout,totalcost):
    if request.method == 'POST':

        Firstname = request.POST.get('firstname')
        Lastname = request.POST.get('lastname')
        user = request.user
        hotel = Hotels.objects.get(id = hotelid)
        room = Room.objects.get(id = roomid)
        cost = totalcost
        newReservation = Reservation()
        newReservation.hotel = hotel
        newReservation.room = room
        newReservation.user = user
        newReservation.guestFirstName = Firstname
```



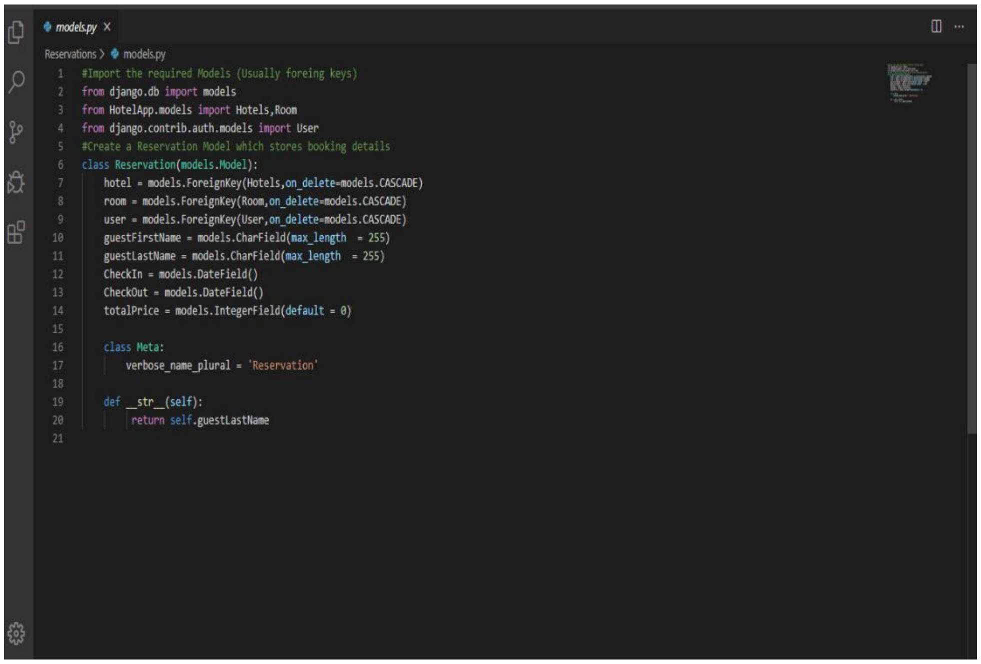
views.py

Reservations > views.py

```python
import datetime
from django import forms
from django.shortcuts import render
from django.core.urlresolvers import reverse,reverse_lazy
from django.core.signing import Signer
from django.http import HttpResponse,HttpResponseRedirect
from django.db.models import Q
from HotelApp.models import Hotels,Room,Proposal
from Authorize.models import UserRole
from ManageHotels.models import Photo
from Reservations.models import Reservation

from django.views import View
from django.template.loader import get_template
from .utils import render_to_pdf

## Generates a PDF using the render help function and outputs it as invoice.html
class GeneratePDF(View):
    def get(self,request, *args, **kwargs):
        booking = Reservation.objects.get(id= self.kwargs['id'])
        template = get_template('invoice.html')
        context = {"booking":booking}
        html = template.render(context)
        pdf = render_to_pdf('invoice.html', context)
        return HttpResponse(pdf,content_type='application/pdf')



## Works out how long the user is staying in a hotel for also working out the total cost.
def bookRoom(request,hotelid,roomid):

    FirstDate = request.session['checkin']
```



- Reservations
  - migrations
  - templates \ Reservat...
    - booking.html
    - mybookings.html
  - __init__.py
  - admin.py
  - apps.py
  - models.py
  - tests.py
  - urls.py
  - utils.py
  - views.py



models.py

Reservations > models.py

```python
#Import the required Models (Usually foreing keys)
from django.db import models
from HotelApp.models import Hotels,Room
from django.contrib.auth.models import User
#Create a Reservation Model which stores booking details
class Reservation(models.Model):
    hotel = models.ForeignKey(Hotels,on_delete=models.CASCADE)
    room = models.ForeignKey(Room,on_delete=models.CASCADE)
    user = models.ForeignKey(User,on_delete=models.CASCADE)
    guestFirstName = models.CharField(max_length  = 255)
    guestLastName = models.CharField(max_length  = 255)
    CheckIn = models.DateField()
    CheckOut = models.DateField()
    totalPrice = models.IntegerField(default = 0)

    class Meta:
        verbose_name_plural = 'Reservation'

    def __str__(self):
        return self.guestLastName
```

Fig5.10 Reservation module files and code

Chapter 6

RESULTS & DISCUSSION

User interface

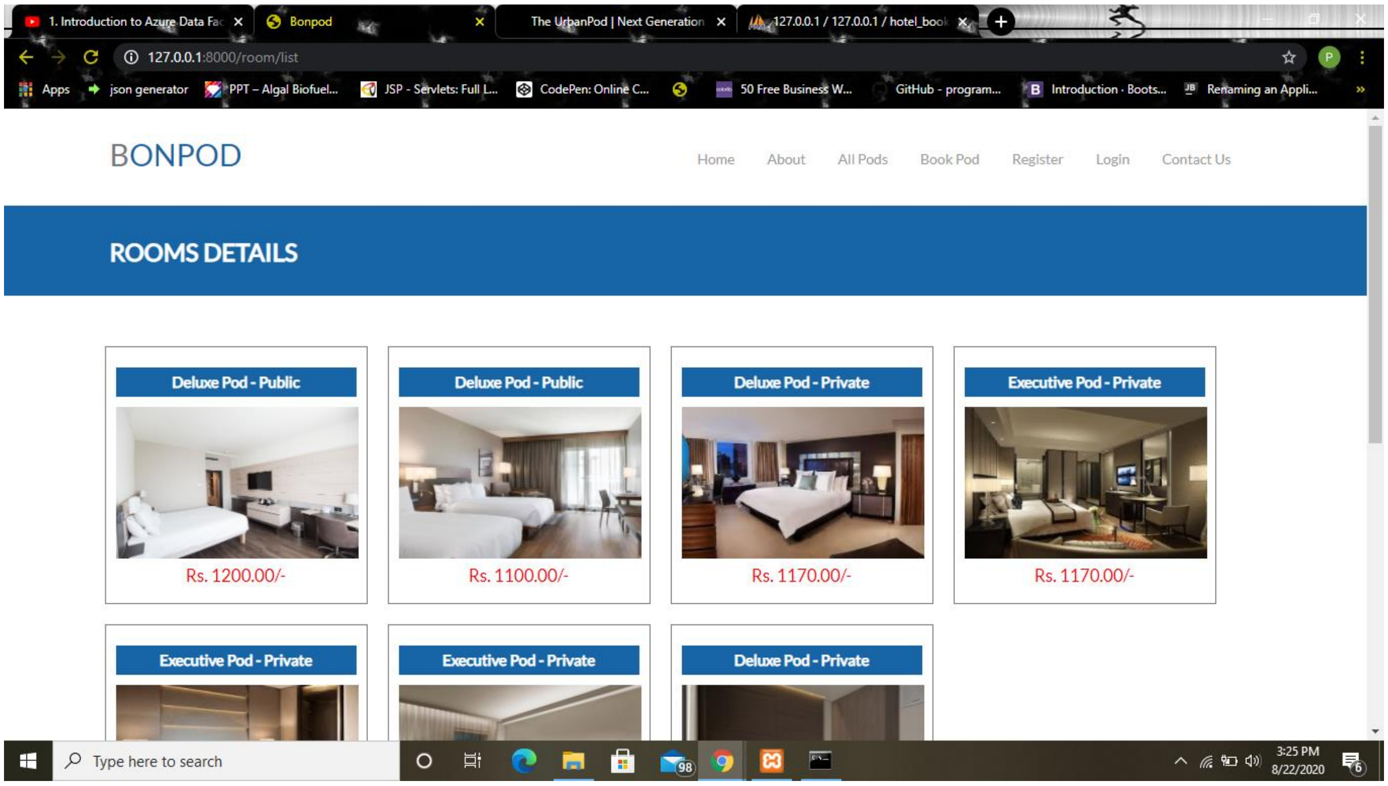


Fig 6.1 Rooms and booking webpage

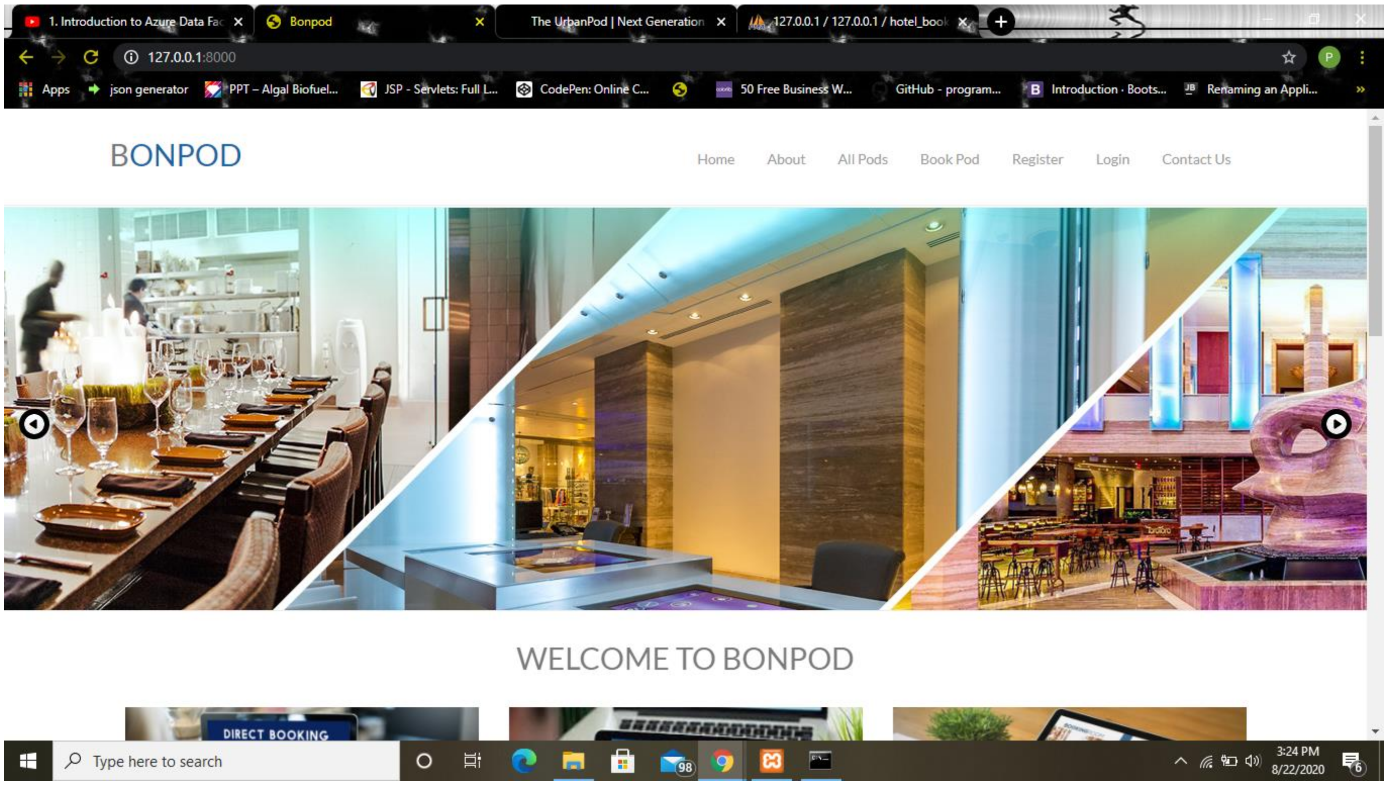
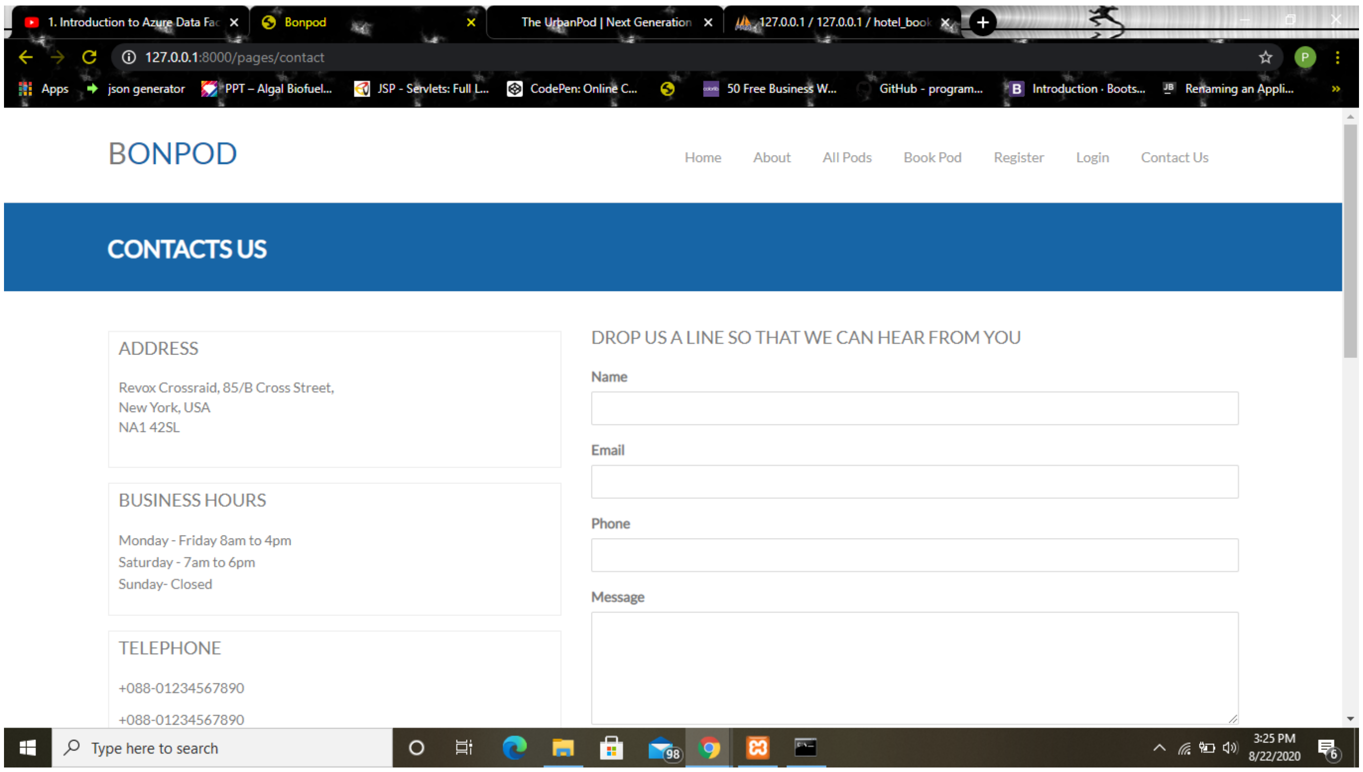


Fig 6.2 Home page of Bonpod



BONPOD

Home About All Pods Book Pod Register Login Contact Us

CONTACTS US

ADDRESS

Revox Crossraid, 85/B Cross Street,
New York, USA
NA1 42SL

BUSINESS HOURS

Monday - Friday 8am to 4pm
Saturday - 7am to 6pm
Sunday- Closed

TELEPHONE

+088-01234567890

+088-01234567890

DROP US A LINE SO THAT WE CAN HEAR FROM YOU

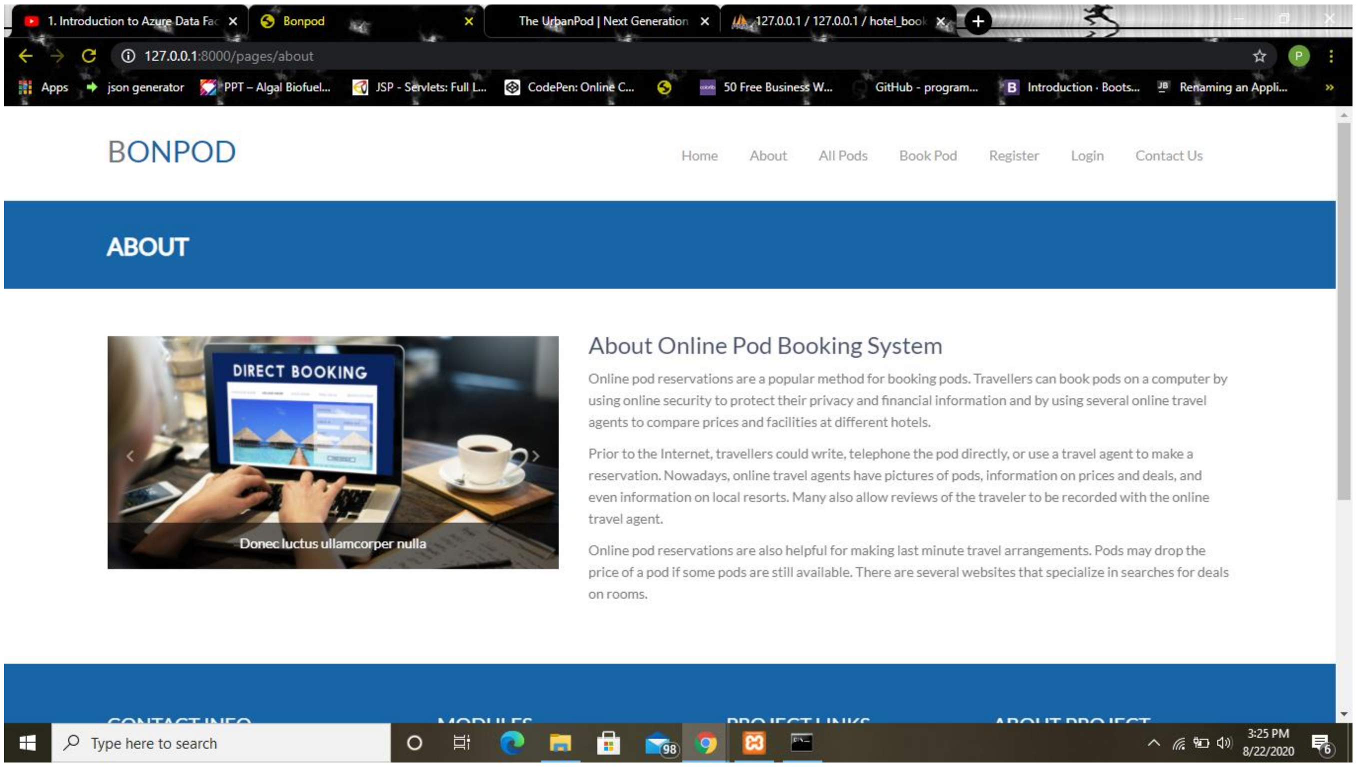
Name

Email

Phone

Message

Fig6.3 Contact us page



BONPOD

Home About All Pods Book Pod Register Login Contact Us

ABOUT

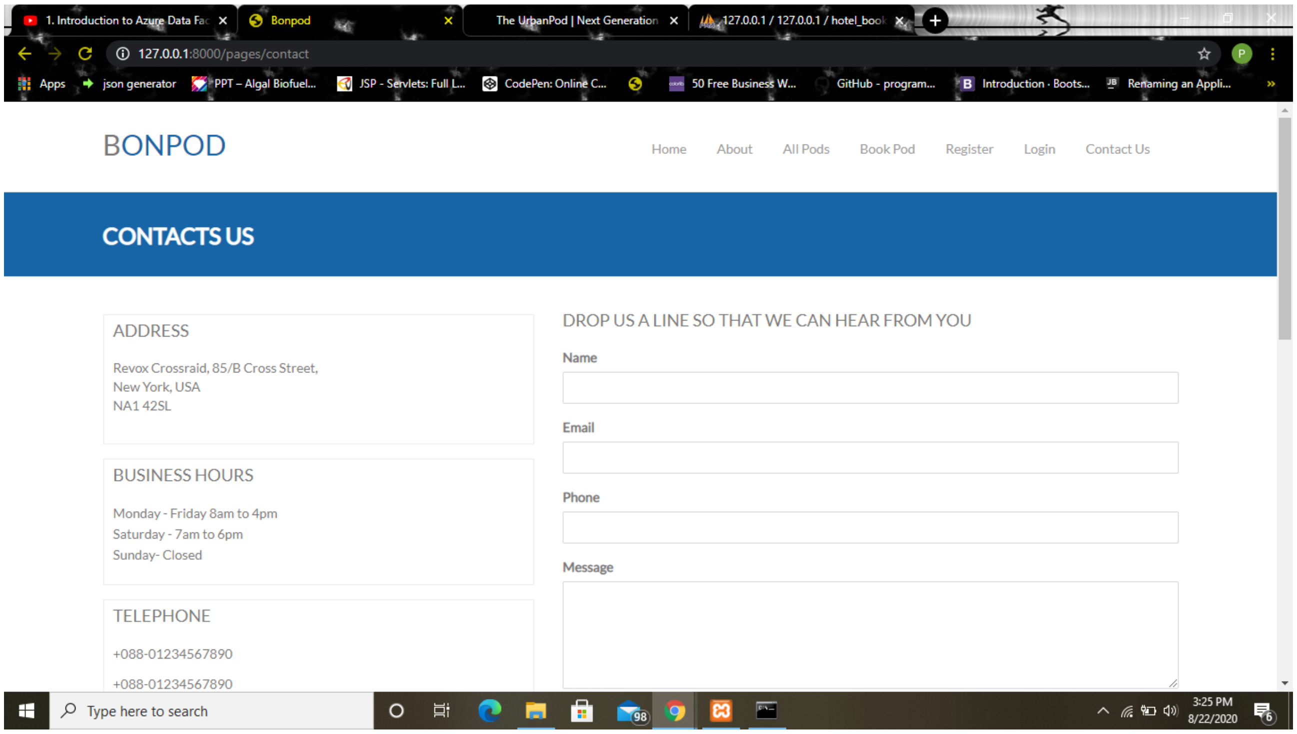
About Online Pod Booking System

Online pod reservations are a popular method for booking pods. Travellers can book pods on a computer by using online security to protect their privacy and financial information and by using several online travel agents to compare prices and facilities at different hotels.

Prior to the Internet, travellers could write, telephone the pod directly, or use a travel agent to make a reservation. Nowadays, online travel agents have pictures of pods, information on prices and deals, and even information on local resorts. Many also allow reviews of the traveler to be recorded with the online travel agent.

Online pod reservations are also helpful for making last minute travel arrangements. Pods may drop the price of a pod if some pods are still available. There are several websites that specialize in searches for deals on rooms.

Fig 6.4 About us page



BONPOD

Home About All Pods Book Pod Register Login Contact Us

CONTACTS US

ADDRESS

Revox Crossraid, 85/B Cross Street,
New York, USA
NA1 42SL

BUSINESS HOURS

Monday - Friday 8am to 4pm
Saturday - 7am to 6pm
Sunday- Closed

TELEPHONE

+088-01234567890
+088-01234567890

DROP US A LINE SO THAT WE CAN HEAR FROM YOU

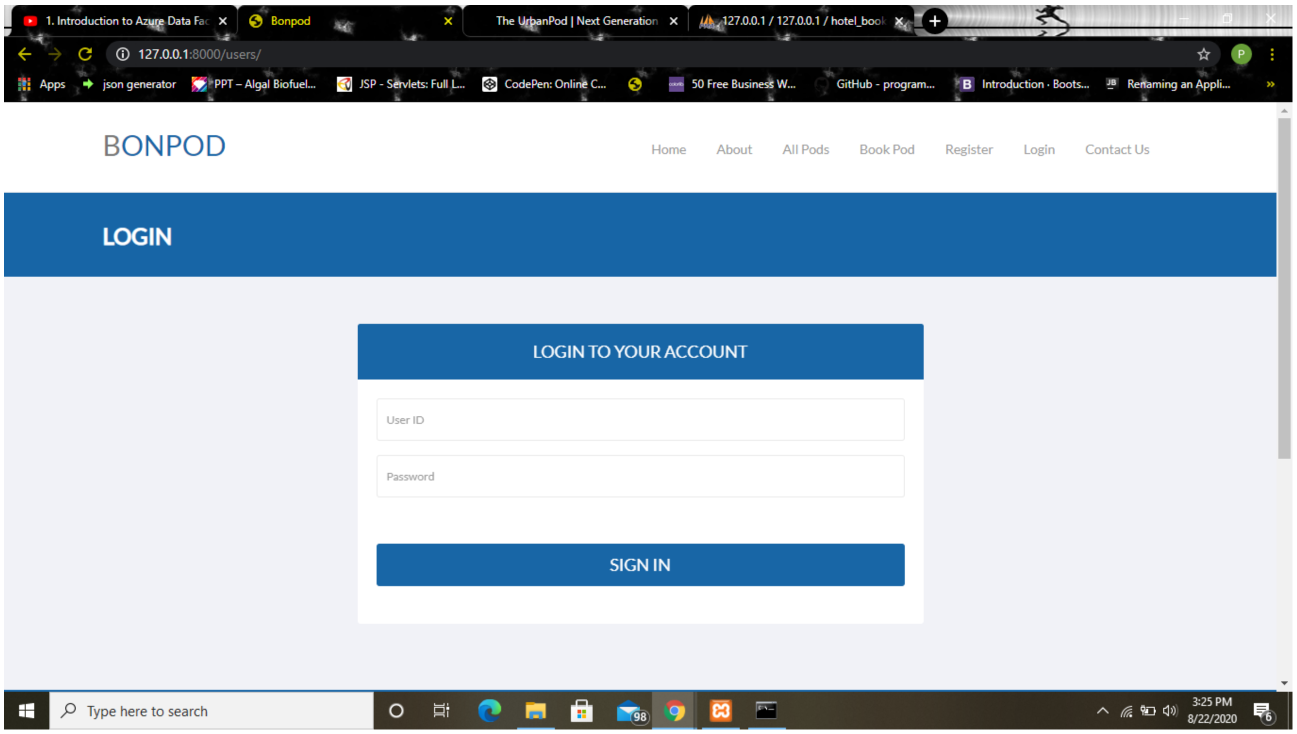
Name

Email

Phone

Message

Fig 6.5 signup page



BONPOD

Home About All Pods Book Pod Register Login Contact Us

LOGIN

LOGIN TO YOUR ACCOUNT

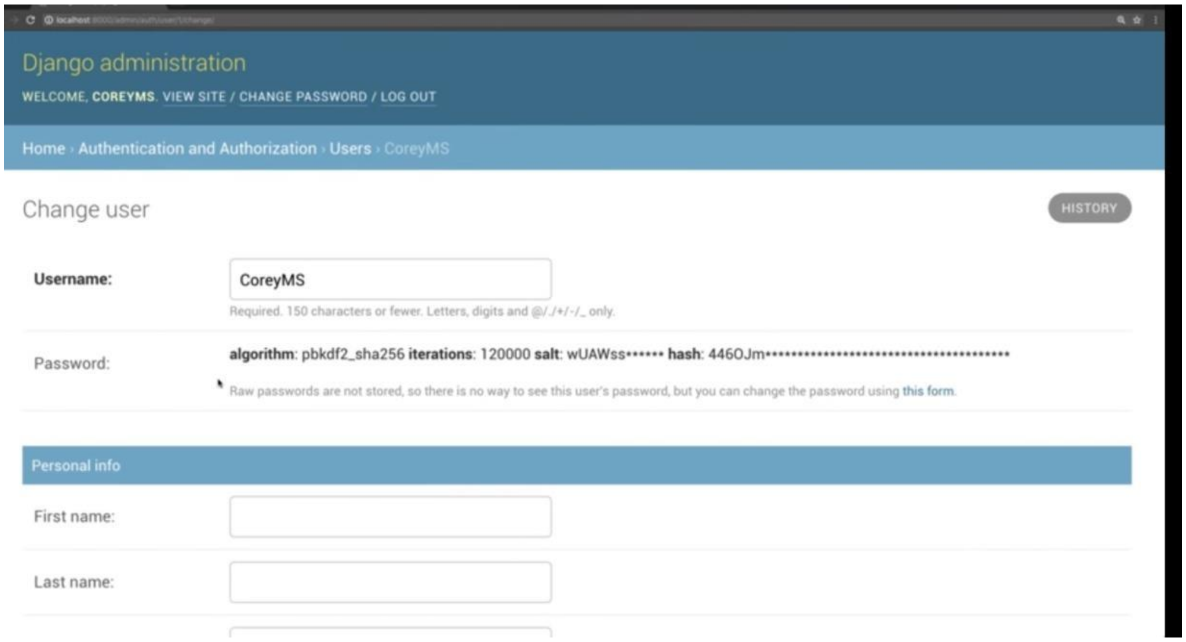
User ID

Password

SIGN IN

Fig 6.6 Sign in page

Admin part


Django administration

WELCOME, COREYMS. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Authentication and Authorization › Users › CoreyMS

Change user

HISTORY

Username: CoreyMS

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password: **algorithm**: pbkdf2_sha256 **iterations**: 120000 **salt**: wUAWss****** **hash**: 446OJm**************************************

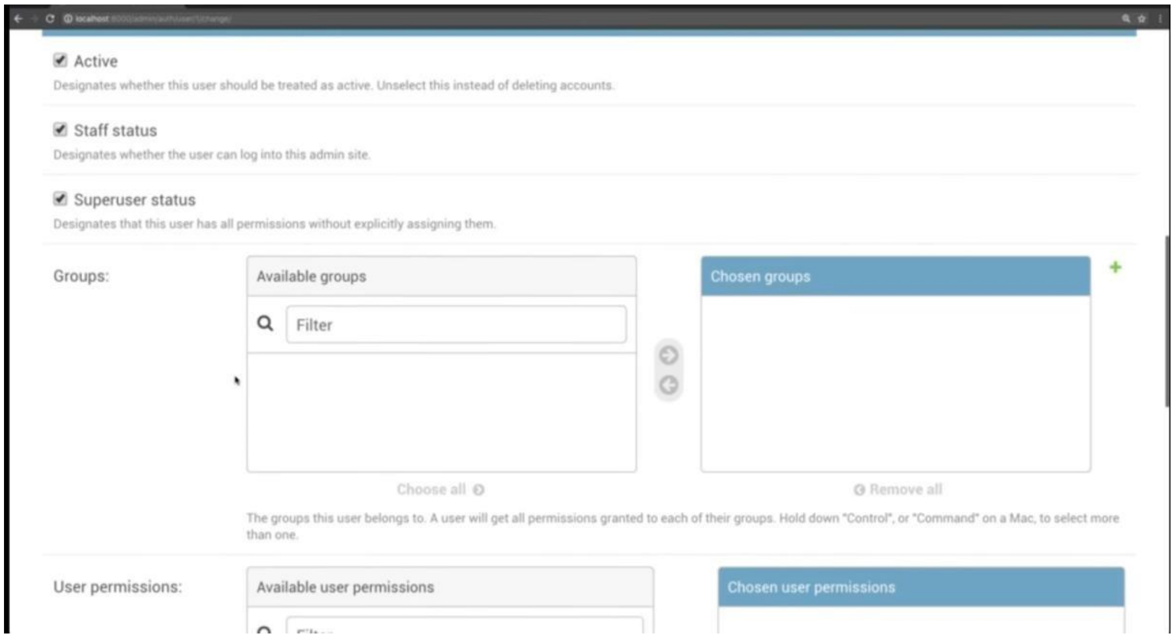
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name:

Last name:

Fig 6.7 Changing user details page


☑ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☑ Staff status

Designates whether the user can log into this admin site.

☑ Superuser status

Designates that this user has all permissions without explicitly assigning them.

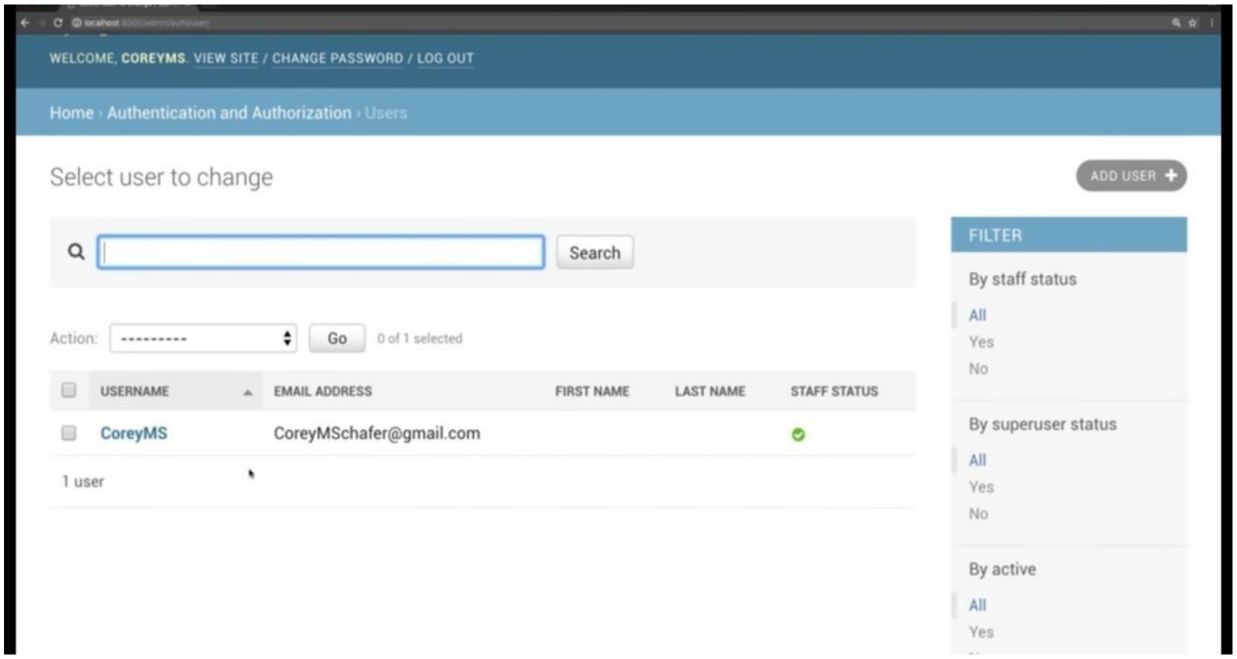
Groups: Available groups | Filter | Chosen groups

Choose all | Remove all

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions: Available user permissions | Chosen user permissions

Fig 6.8 Status Check page


WELCOME, COREYMS. VIEW SITE / CHANGE PASSWORD / LOG OUT

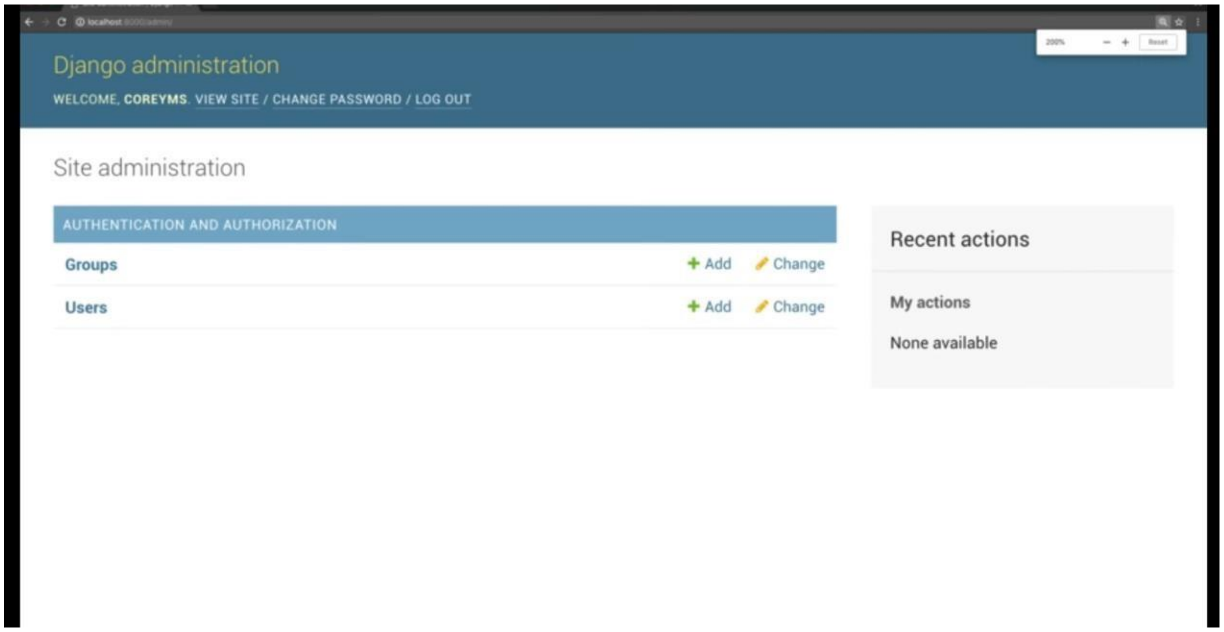
Home › Authentication and Authorization › Users

Select user to change

	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
	CoreyMS	CoreyMSchafer@gmail.com			✓

1 user

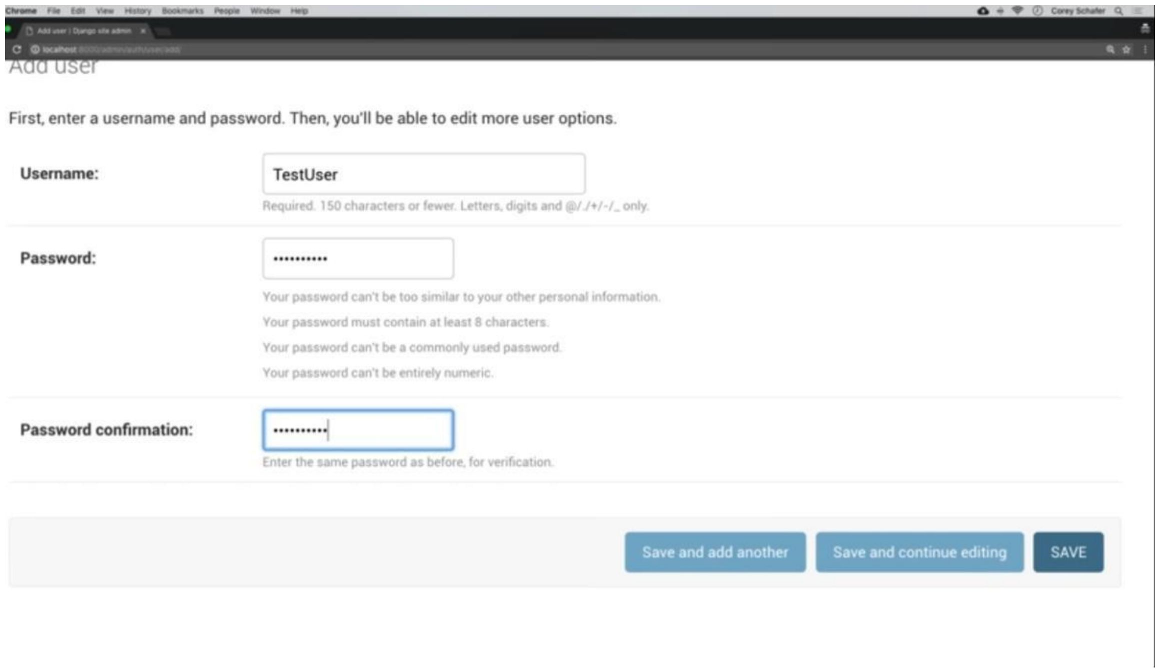
Fig 6.9 No. of Users page


Django administration

WELCOME, COREYMS. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

Fig 6.10 Groups and user page


Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username:	TestUser Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.
Password:	•••••••••• Your password can't be too similar to your other personal information. Your password must contain at least 8 characters. Your password can't be a commonly used password. Your password can't be entirely numeric.
Password confirmation:	•••••••••• Enter the same password as before, for verification.

Fig 6.11 Add user page

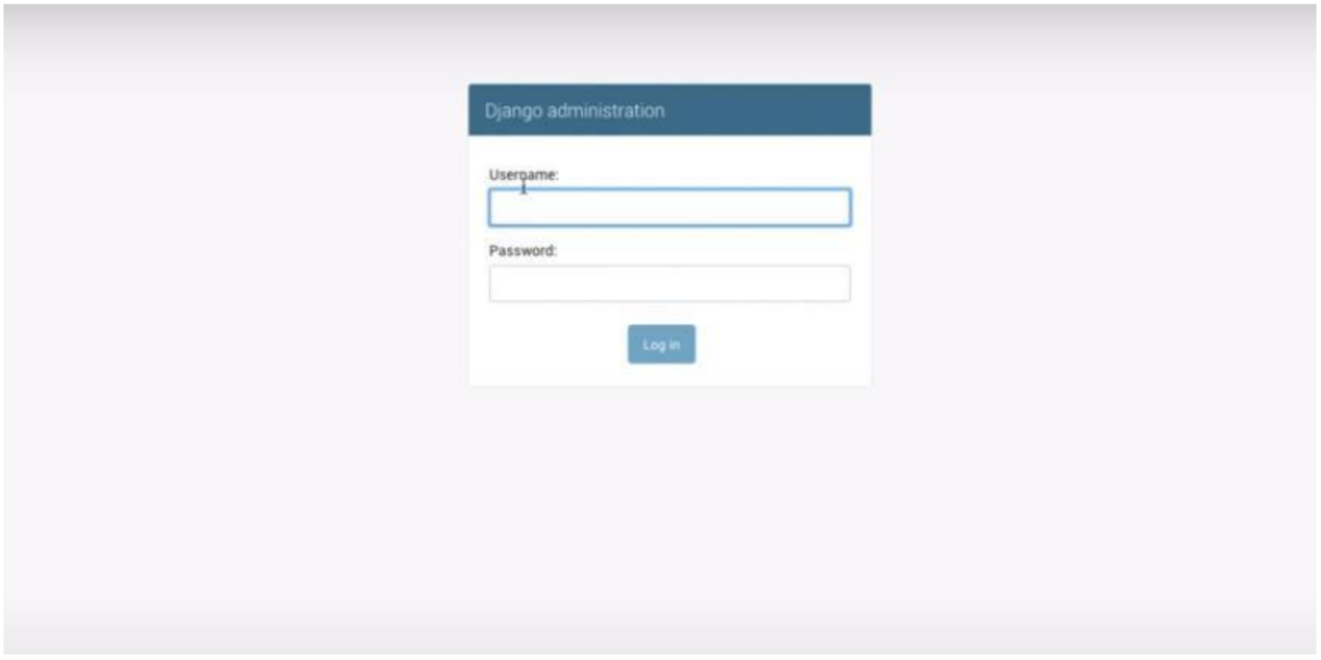


Fig 6.12 Admin login page

CONCLUSION

Through this project, we have successfully implemented a complete Bonpod website using HTML, CSS and JavaScript for the front end and Django framework for the backend and admin part and have estimated its techniques. Hence, these can be implemented to successfully log in the website and book a desired pod at any time. Also, we are getting the desired and accurate result while booking the pod and it is successfully implementated.

REFERENCES

[1] Mitra, N., "SOAP Version 1.2 Part(): Primer", W3C Recommendation, http://www.w3,org/TR/soap12-part(),2003

[2] UDDI Consortium, :UDDI Specification", http://www.uddi.org/specification.html,2000

[3] Christensen, E., Cubera, F., Meredith, G., Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note 15 March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[4] Cabral L., Domingue J., Motta E., Payne T., Hakimpour F., "Approaches to Semantic Web Services: An overview and compairasions", Lecture Notes in Computer Science, Vol.3053, pp.225-239,2004.

[5] Antoniou, G and Van Harmetial, F., "A Django web primer", Cambridge, MA:The MIT Press,2004.

[6] R. Romune and L. Lee, "The single version of HTML Being activity development" https://html.spec.whatwg.org/multipages/

[7] F. Seimon , P. Header and Lee M., "Single version of DOM specification being activity developed." https://dom.spec.whatwg.org/

[8] Lausen H. Polleres A.,Romaand., "Web Service Modeling ontology (WSMO)" http://www.w3.org/Submission/WSMO/,2005.