RentIt

Project Report Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Engineering
in
Information Technology

Submitted by

Chetan Kumavat : (Roll No. 19UITE9006)

Vishal Khandelwal : (Roll No. 19UITE9020)

Under the Supervision of

Abhisek Gour
Assistant Professor

Department of Computer Science and Engineering
MBM University, Jodhpur

June, 2022

RentIt

Project Report Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Engineering
in
Information Technology

Submitted by

Chetan Kumavat : (Roll No. 19UITE9006)

Vishal Khandelwal : (Roll No. 19UITE9020)

Under the Supervision of

Abhisek Gour
Assistant Professor

Department of Computer Science and Engineering
MBM University, Jodhpur

June, 2022

This page was intentionally left blank.

Department of Computer Science & Engineering
MBM University
Ratanada, Jodhpur, Rajasthan, India –342011

CERTIFICATE

This is to certify that the work contained in this report entitled **"RentIt**" is submitted by the group members Mr. Chetan Kumavat (Roll No.: 19UITE9006), Mr. Vishal Khandelwal (Roll No.: 19UITE9020), to the Department of Computer Science & Engineering, MBM University, Jodhpur, for the partial fulfillment of the requirements for the degree of **Bachelor of Engineering** in **Information Technology**.

They have carried out their work under my supervision. This work has not been submitted elsewhere for the award of any other degree or diploma.

The project work in our opinion has reached the standard fulfilling of the requirements for the degree of Bachelor of Engineering in Information Technology in accordance with the regulations of the Institute.

Abhisek Gour
Assistant Professor
(Supervisor)
Dept. of Computer Science & Engg.
MBM University, Jodhpur

Prof. N.C. Barwar
(Head)
Dept. of Computer Science & Engg.
MBM University, Jodhpur

This page was intentionally left blank.

DECLARATION

We, ***Chetan Kumavat*** and ***Vishal Khandelwal***, hereby declare that this project titled **"RentIt"** is a record of original work done by us under the supervision and guidance of ***Asst. Prof. Abhisek Gour***.

We, further certify that this work has not formed the basis for the award of the Degree/Diploma/Associateship/Fellowship or similar recognition to any candidate of any university, and no part of this report is reproduced as it is from any other source without appropriate reference and permission.

SIGNATURE OF STUDENT

(Chetan Kumavat)
8th Semester, IT
Enroll. - 17R/06331
Roll No. - 19UITE9006

SIGNATURE OF STUDENT

(Vishal Khandelwal)
8th Semester, IT
Enroll. - 18R/33043
Roll No. - 19UITE9020

This page was intentionally left blank.

ACKNOWLEDGEMENT

We, Chetan Kumavat and Vishal Khandelwal were indebted to **Dr. Nemi Chand Barwar**, Head of Department, Computer Science & Engineering, MBM University, Jodhpur for the support and availability of facilities by the department.

We wish to express our deepest sense of gratitude to our seminar guide **Abhisek Gour**, **Assistant Professor**, Dept. of Computer Science & Engg., for his able guidance and useful suggestions, that helped us in completing the project work, on time. His guidance, encouragement, suggestion, and constructive criticism have contributed immensely to the evolution of our ideas on the report.

We must acknowledge with heartfelt gratitude the authors of the references and other literature referred to in this project.

Lastly, yet most importantly, we would like to express our heartfelt thanks to our family, friends, and peers for their blessings, wishes, and support for the successful completion of this report.

This page was intentionally left blank.

ABSTRACT

Technology plays a massive role in our everyday lives, from the easiest of applications to the most revolutionary inventions. A web developer has created every website or web page that we come across. Web development, also known as website development, refers to the tasks associated with creating, building, and maintaining websites and web applications that run online on a browser.

A web application is an application program that is usually stored on a remote server, and users can access it through the use of Software known as a web browser. In general, a web application can contain online shops (or we can also tell them e-commerce shops), webmail, calculators, social media platforms, etc. There is also some kind of web application that usually requires a special kind of web browser to access them.

Web Framework is a software framework that is designed to support the development of web applications including web services, web resources, and web APIs. We have two functions of frameworks — a) the one to work on the server side, that helps to set up app logic on the server (backend) or b) to work on the client-side (front end).

This project report discusses our major project which is RentIt – A Rental WebApp. We have developed this project by integrating front-end languages and frameworks with back-end languages and frameworks with the objective that people can use the rented product instead of purchasing a new product whenever they are needed it for a short duration of time at a cost lower than the cost of a new product.

This page was intentionally left blank.

Contents

This page was intentionally left blank.

List of Figures

This page was intentionally left blank.

Chapter 1

INTRODUCTION

Technology plays a massive role in our everyday lives, from the easiest of applications to the most revolutionary inventions. A web developer has created every website or web page that we come across.

1.1 What is Web Development

Web development, also known as website development, refers to the tasks associated with creating, building, and maintaining websites and web applications that run online on a browser. It may, however, also include web design, web programming, and database management. Web development is closely related to the job of designing the features and functionality of apps (web design). The term development is usually reserved for the actual construction of these things (that is to say, the programming of sites).

Fig 1.1: Web Development

1.2 Why is Web Development Important

The Internet isn't going anywhere. In fact, it's become a portal and primary method of research, connection, education, and entertainment in the world.

As of 2021, there were 4.66 billion global Internet users — more than half the world's population. Given the rapidly-increasing number of Internet users, it's no surprise that web development is a rapidly expanding industry. Between now and 2030, the employment of web developers is expected to grow by 13%, much faster than most other technology careers.

1.3 History

The idea of the internet had existed in some form for at least half a century before it finally became a common household utility in the 1990s. Conceived in the 1980s, the World Wide Web gained significant traction with the introduction of the Mosaic browser in 1993. Shortly thereafter, businesses began recognizing the web's commercial potential, as network infrastructure grew to accommodate what would prove to be a massive influx of online activity. Then the tech bubble grew and burst, the survivors of which (Google, Amazon, and the like) went from being key tech influencers to veritable corporate giants within about a decade.

In 1989, Tim Berners-Lee (then a fellow at the CERN Laboratory in Europe) outlined his concept of a computer platform that could facilitate collaboration among researchers who are based in different parts of the world. This led to the invention of the Hypertext Markup Language (HTML) in 1990.

The web advanced a great deal in the years following the tech crash of 2000–2001. During this time, the government started to play an increasingly influential role in the web, while concurrently, strong tech companies emerged from the ashes of the big collapse to set the new course for digital commerce and culture. And as this newer and more solid foundation was laid, the internet increasingly became the main channel for telecommunications in the modern age.

As hardware improvements cultivated broader networks and greater bandwidth, web development responded by enabling designers with an array of multimedia to incorporate into the growing and diversifying art of web presentation. Cascading Style Sheets afforded web design with new ways to organize and display content.

Yet still, with all of these revolutions and progressions in web development – the basic interface and structure of the web page have maintained its integrity and balance of form and function.

1.4 What is Web App

A web application is an application program that is usually stored on a remote server, and users can access it through the use of Software known as a web browser. A web application can be developed for several uses, which can be used by anyone like it can be used as an individual or as a whole organization for several reasons.

In general, a web application can contain online shops (or we can also say e-commerce shops), webmail, calculators, social media platforms, etc. There is also some kind of web application that usually requires a special kind of web browser to access them. We cannot access those kinds of web applications by using regular web- browsers. However, most of the web applications available on the internet can be accessed using a standard web browser.

If we talk about the web application in general, a web application usually uses a combination of server-side scripts such as PHP, and ASP, for handling the information/ data storage and retrieval of the data.

Some of them also use client-side scripts such as JavaScript, and HTML to represent the data/information in front of the users, and some of the web applications are also using both server-side and client-side at the same time.

It allows the users to communicate with the organization or companies by using the online form, online forums, shopping carts, content management system, and much more.

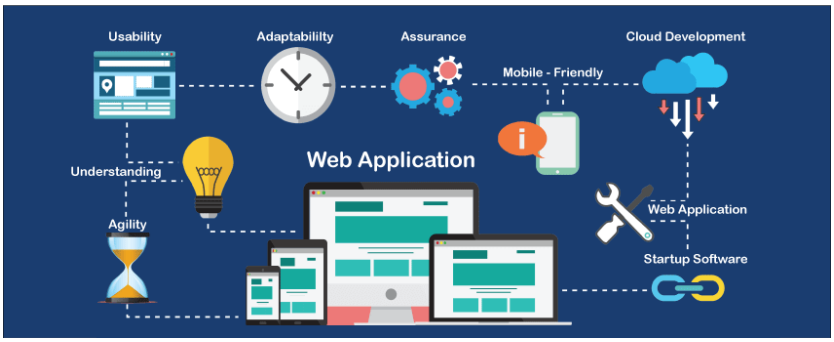

Fig 1.2: Web Application

1.5 Flow of Web App

Let's understand what the flow of the typical web application looks like:

- In general, a user sends a request to the web server using web browsers such as Google Chrome, Microsoft Edge, Firefox, etc over the internet.
- Then, the request is forwarded to the appropriate web application server by the web server.
- Web application server performs the requested operations/ tasks like processing the database, querying the databases; produces the result of the requested data.
- The obtained result is sent to the web server by the web application server along with the requested data/information or processed data.
- The web server responds to the user with the requested or processed data/information and provides the result to the user's screen.

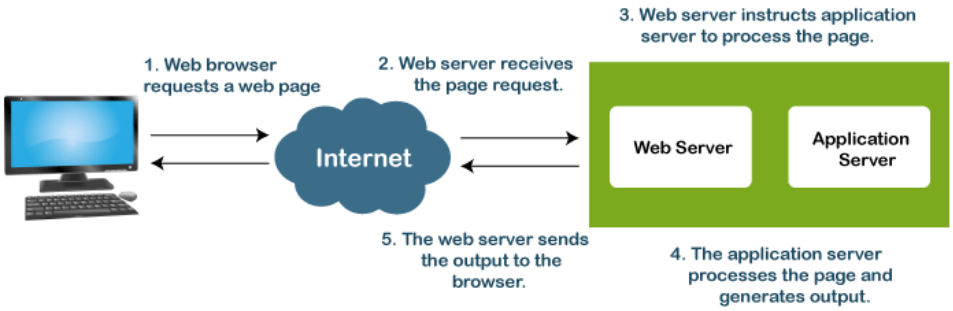

Fig 1.3: Flow of Web Application

1.6 What is Web Framework

Web Framework is a software framework that is designed to support the development of web applications including web services, web resources, and web APIs. In simple words, web frameworks are a piece of software that offers a way to create and run web applications. Thus, you don’t need to code on your own and look for probable miscalculations and faults.

In earlier days of web app development, web frameworks were introduced as a means to end hand-coding of applications where just the developer of a particular app could change it. That was long ago, now we have web-specific languages and the trouble with changing an app's structure is resolved because of the arrival of general performance. Now, depending upon your task you may choose one web framework that fulfills all your requirements or converges multiple frameworks.

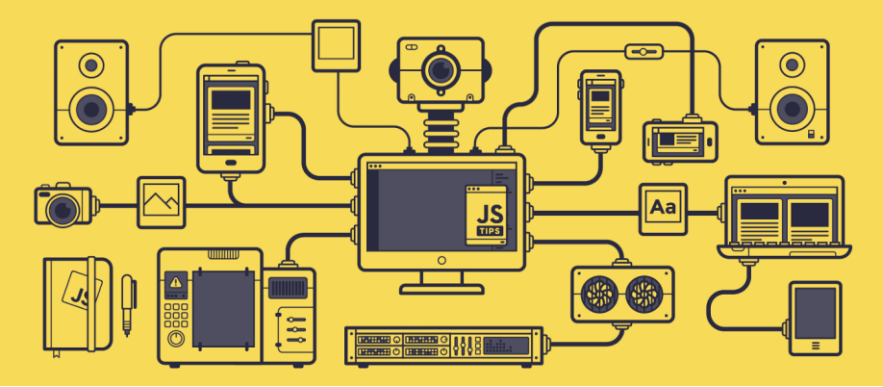

Fig 1.4: Working of Web Framework

1.7 Types of Web Frameworks

As web standards began to advance, the app logic shifted towards the client-ensuring smarter communication between the user and the web application. With logic on the client-side, they (client) can react swiftly to user input. This makes web apps more responsive and easily navigatable on any device. Thus, we have two functions of frameworks — a) the one to work on the server side, that helps to set up app logic on the server (backend) or b) to work on the client-side (front end).

The front-end frameworks mostly manage the external part of the website, i.e. what the user sees when they open the application. The back-end manages the internal part. Let's take a deeper look.

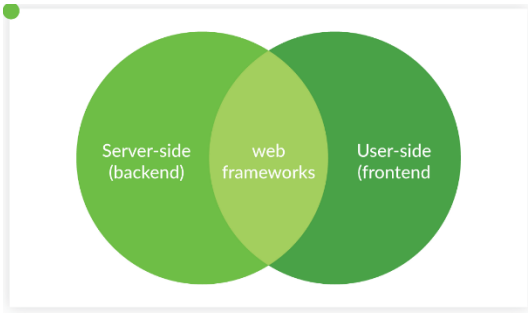

Fig 1.5: Types of Web Frameworks

1.7.1 Server-side Frameworks

The rules and architecture of the server-side frameworks permit you to create simple pages, landings, and forms of different kinds. To create a web application with a well-developed interface you need a wider range of functionality. Server-side frameworks handle HTTP requests, database control, and management, URL mapping, etc. These frameworks can improve security and form the output data- simplifying the development process. Some of the top server-side frameworks are –

- NET (C#)
- Django (Python)
- Ruby on Rails (Ruby)
- Express (JavaScript/Node.JS)
- Symfony (PHP)

1.7.2 Client-side Frameworks

Client-side frameworks don't take care of the business logic like the server-side ones. They function inside the browser. Therefore, you can enhance and implement new user interfaces. Several animated features can be created with frontend and single-page applications. Every client-side framework varies in functionality and use. Here are some client-side frameworks for comparison's sake; all of whom use JavaScript as their programming language –

- Angular
- Ember.JS

- Vue.JS
- React.JS

1.8 Web App Framework - Architecture

Most web frameworks depend on the MVC (Model-View-Controller) architecture. The reason why this pattern is preferred lies in its rational design that separates the app logic from the interface and forms the three essential parts that are represented in the architecture's name — MVC (Model-View-Controller).

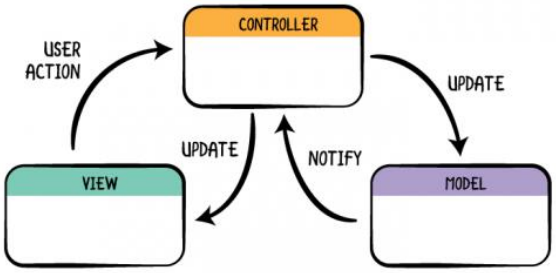

Fig 1.6: MVC Pattern

1.8.1 Model

The Model comprises all the data, business logic layers, guidelines, and functions. The Model, upon getting user input data from the Controller, tells the way an updated interface should be displayed directly to the View.

1.8.2 View

The View is for the graphical representation of the data like graphs or charts etc. It is the app's front-end. The View gets the user input and communicates the same to the Controller for examination and then updates and reconstructs itself according to the Model's instructions, or the Controller's in case the modification requirement is minimum.

1.8.3 Controller

The Controller translates the input data into the scope of commands of the previous ones. It is the midway between the Model and the View. It gets the user input from the View; after processing it, the Controller notifies the Model (or View) of the changes required.

This page was intentionally left blank.

Chapter 2

TECHNOLOGY USED

2.1 System Requirements

- **Hardware Requirements –**
  - ❖ Processor : Intel ® Core i3
  - ❖ Installed Memory : 2 GB RAM
  - ❖ Processor Type : 64-bit Processor
  - ❖ Cache Memory : 512 KB
- **Software Requirements –**
  - ❖ OS : Windows XP/7/8/8.1/10/11 or Ubuntu/Debian or MacOS (32-bit/64-bit)
  - ❖ CPU : 4^{th} Generation
  - ❖ Monitor : 1280 x 720 Screen Resolution, IPS Display

2.2 Technology Used

2.2.1 Notepad++ (Code Editor)

Notepad++ is a free, open-source text and source code editor for use with Microsoft Windows. Written in the C++ programming language, Notepad++ prides itself in paring down on unnecessary features and streamlining processes to create a light and efficient text notepad program. In practical terms, this means high speed and an accessible, user-friendly interface.

Its features are as follows:-

- Edit text files up to 2 GB in size (the maximum size in Windows Notepad is 58 MB).
- Edit multiple files, organized in tabs (tabbed editing).
- Line numbering.
- Syntax highlighting for over 70 programming languages, including HTML and Windows batch files.
- Advanced find and replace, with support for regular expressions.
- Create and edit text files for different operating systems, including macOS and Linux.
- Split-screen for editing and viewing multiple files at once, or multiple parts of the same file.
- Macros for recording a sequence of editing commands to be executed repeatedly.
- Support for lines to have bookmarks.
- A plugin system for adding features to the software.

And many more…

2.2.2 Languages & Frameworks Used

In this project, we have used Python (Back-End Language), Django (Web Framework), Twilio (Provides Communication APIs), HTML, CSS, & JS (Front-End Languages), Bootstrap (HTML, CSS, & JS Framework), and SQLite (Database). We have integrated both front-end and back-end technologies to develop our major project.

- **HTML –**

  HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. "Hypertext" refers to links that connect web pages, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet

and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, , , <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. However, the convention and recommended practice are to write tags in lowercase.

- **CSS –**

  Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

  CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, and CSS3. However, CSS4 has never become an official version.

  CSS (Cascading Style Sheets) is used to style and layout web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

- **JavaScript –**

JavaScript (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is easy to learn and also a powerful scripting language, widely used for controlling web page behavior.

Contrary to popular misconception, JavaScript is not "Interpreted Java". In a nutshell, JavaScript is a dynamic scripting language supporting prototype-based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such as if statements, for and while loops, and switch and try ... catch blocks function the same as in these languages (or nearly so).

- **Python –**

  Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components.

  Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

  Its features are as follows:-

  - **Easy-to-learn:**

    Python has few keywords, a simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

  - **Easy-to-read:**

Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain:**

  Python's source code is fairly easy-to-maintain.

- **Broad standard library:**

  Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode:**

  Python has support for an interactive mode that allows interactive testing and debugging of snippets of code.

- **Portable:**

  Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Extendable:**

  You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases:**

  Python provides interfaces to all major commercial databases.

- **GUI Programming:**

  Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable:**

  Python provides a better structure and support for large programs than shell scripting.

And many more…

- **Django –**

Django is a Python-based free and open-source web application framework for the backend development of web applications and websites.

It follows the Model View Template (MVT) architectural pattern. It separates the code into three distinct parts – Model, View, and Templates. The developers just have to code what should be displayed to the user and Django will take care of all the background details.

Maintained by the Django Software Foundation, the primary goal of Django is to ease the creation of complex, database-driven websites.

The Python web framework Django emphasizes the following core areas:

- Reusability
- Pluggability of components
- Less code
- Low coupling
- Rapid development

Why Django For Web Development?

- **KISS and DRF compliant:**

  The 'Don't Repeat Yourself' and 'Keep It Short and Simple' philosophy is closely followed by the Django web framework. The simplicity and reusability of elements are the main cornerstones of Python, and Django effortlessly carries it onward.

- **Batteries-included:**

  Django framework is batteries-included, meaning it comes inbuilt with a lot of tools and utilities that add to the building of complex web applications.

  It incorporates an admin panel, user interface, testing web browsers, code editing tools, indentation, productive user interface, and more.

- **Intelligent Templating System:**

Django template is a text document or a Python string that later gets converted to markup Python code using the Django template language (DTL). The template engine is designed to be familiar and useful, striking a balance between feature-richness and ease of use.

It's similar to HTML, and thus comfortable and easy to learn. Django templates are highly flexible allowing developers to augment the template language as per their requirements.

- **Powerful Object Relational Mapper (ORM):**

  The Django ORM provides a bridge between the relational database tables and models, keeping the hassle of maintaining and updating large databases.

  In addition, it supports all the major relational databases such as MySQL and PostgreSQL, if your organization is reluctant to shift to Django's in-built ORM.

  Django applications are perfect when you are developing data-driven content management systems, as they can handle large datasets.

- **Automatic Admin Interface:**

  Django's automatic admin interface is one of the most powerful admin panel generation tools for frameworks. It acts as an internal management tool for your business and is useful for managing content on your site.

  It reads metadata from your models to create a powerful and production-ready interface for admins. You can simply customize the interface to your requirements.

- **REST framework:**

  Django REST Framework(DRF) is a powerful toolkit for developing web APIs. Using DRF you can significantly reduce your development time and create RESTful APIs.

The Django web application acts as a REST server, for simplified serialization, handling authentications and permissions. These web APIs enhance your web application with new, unique features, adding to your object's functionalities.

❖ **Safety and Security:**

Django has one of the best security systems out there. A Django app is safe and secure, promoting good security practices in developers.

The Django framework is regularly updated with new security patches, and also protects against cybersecurity attacks such as Clickjacking attacks, CRLF injections, and timing attacks.

It offers security features for consumers such as user authentication, scanning user-uploaded content, and different user logins with defined access settings.

❖ **Testing and Debugging Support:**

Automated testing is an extremely useful debugging and testing tool for web developers. Django provides a set of tools that automate the testing process.

The test client is a Python class that acts like a dummy web browser, allowing you to test your views and interact with your Django application.

It also allows you to test the output of your asynchronous views using the test client or even write your own fully asynchronous tests for your Django project.

Thus Django is the best web framework for test-driven development.

- **Bootstrap –**

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all browsers (IE, Firefox, and Chrome) and for all sizes

of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers – Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

Why We Use Bootstrap?

- ❖ It is a Faster and Easier way for Web-Development.
- ❖ It creates Platform-independent web pages.
- ❖ It creates Responsive Web-pages.
- ❖ It designs responsive web pages for mobile devices too.
- ❖ It is a Free and open-source framework available on www.getbootstrap.com.

- **Twilio (Modern Communication API) –**

Twilio is a modern communication API Used by developers for establishing communications. Twilio can be used to send SMS, WhatsApp, Voice, Video, email, and even IoT, across the customer journey. All you need to do is integrate its API with your software.

Twilio offers complete solutions to building communication with telephony. More than a million developers and leading brands are already using Twilio to build innovative communications solutions. Twilio Communications APIs enable voice, messaging, and video conversations within the web and mobile apps. This facilitates developers to make easy communications between different apps.

How to Connect the Twilio API?

- ❖ Get the Twilio Credentials to access the Communication API
- ❖ Create a Twilio Account by going on https://twilio.com/console. You can choose a Free trial from there.
- ❖ After creating the accounts head over to the account details page and Get the API.
- ❖ Get your Free Twilio Phone number from there so that you will be able to make calls and send messages.

- ❖ Get Twilio Messaging SID, if you want to send SMS and MMS.
- ❖ Find Twilio API at RapidAPI.com or go directly to the Twilio Package Page.
- ❖ Use Your Credentials to Make Connections with Account SID and Account Token.
- ❖ Login to RapidAPI to Access your Code Snippets.
- ❖ Once you log in, customize Fields and Language and select the code that fits best to your project.

Fig 2.1: Key Areas of Twilio

- **SQLite Database –**

  SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

  SQLite source code is in the public domain and is free for everyone to use for any purpose.

  Why SQLite?

  - ❖ SQLite does not require a separate server process or system to operate (serverless).
  - ❖ SQLite comes with zero-configuration, which means no setup or administration is needed.

- A complete SQLite database is stored in a single cross-platform disk file.
- SQLite is very small and lightweight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
- SQLite is self-contained, which means no external dependencies.
- SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
- SQLite supports most of the query language features found in the SQL92 (SQL2) standard.
- SQLite is written in ANSI-C and provides a simple and easy-to-use API.
- SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

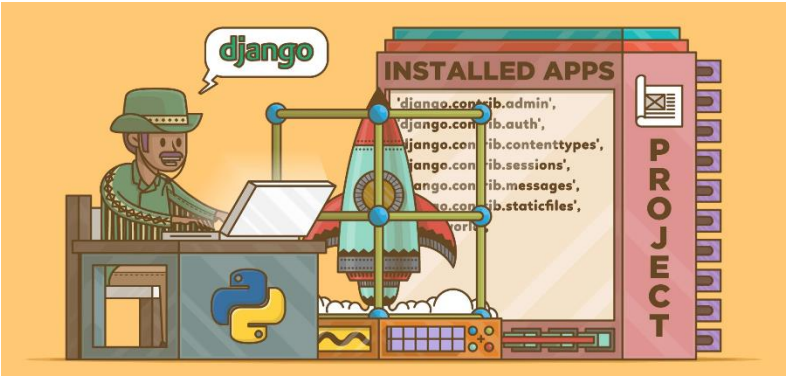

Fig 2.2: Web Development Using Django

This page was intentionally left blank.

Chapter 3

PROJECT DETAILS

Developed a RentIt – A Rental WebApp as our major project by integrating front-end technologies with back-end technologies. In this project, the front-end part was developed by Chetan, and the back-end part was developed by Vishal.

3.1 Aim of the Project

RentIt – A Rental WebApp allows users to rent their belongings as well as they can also request and use (after the request is accepted) the products from the list of already rented products. To authenticate the user by the OTP through SMS, this web app uses the Twilio service.

It is almost similar to an E-commerce website. But there is a difference in both of them depending on their business models. An E-commerce website works on the B2C (Business-To-Client) model whereas this web app works on the C2C (Client-To-Client) model.

Main Objectives:-

- Through this rental web app, users can easily use the rented items whenever they are needed instead of purchasing the new one at a lower cost than the cost of the newly purchased product.
- It tries to eliminate the stocking up problem of newly purchased items, which get stocked up in our house after a limited use by promoting the use of rented products.
- Users can earn some money by renting their unused belongings.

3.2 Project Stages

We can segregate the project architecture based on the tools used:

- **HTML, CSS, JS –** Used for creating web app interface (web app's layout).
- **Bootstrap –** Used for responsive, mobile-first web app design.
- **Python –** Used for back-end programming.
- **Django –** Used for creating a web app and also the integration of the front-end with the database.
- **Twilio –** Used for sending OTP to the user for account verification.
- **SQLite –** Database (for storing data).

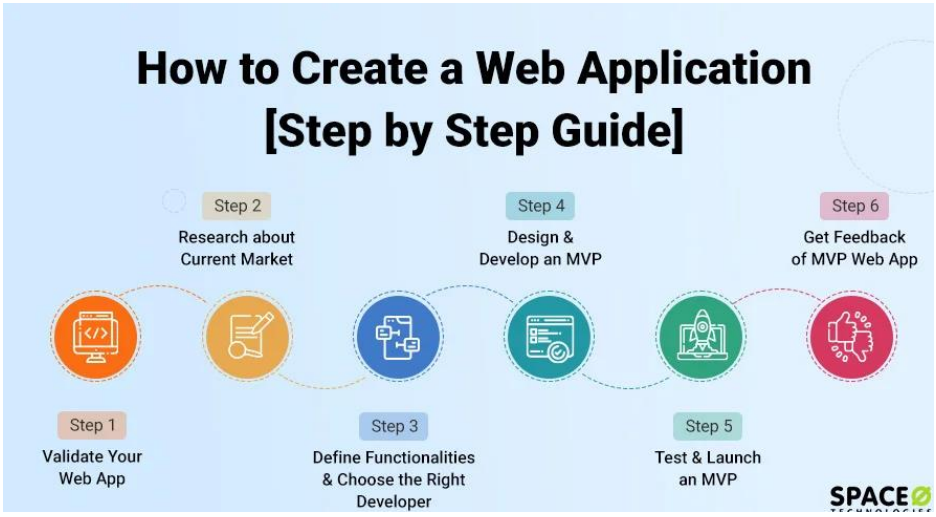

Fig 3.1: Stages of the Web App Development

3.3 Working of Project

- First of all, users need to register using a Name, E-mail ID, Mobile Number, and a secure password.
- It will generate an OTP which will be sent to the user's mobile phone.
- Then, he needs to put the OTP in the OTP box for completion of the registration process.

- Then the user will be redirected to the login page, where he needs to put his name as username and password (created during registering process) to login into the account.
- After logging into the account, the user will first see the page of posting a product where he can post a product for rent by providing all necessary details.
- If he posts the product by clicking on submit button on the posting page, then a request is sent to the admin and when the admin approves that request, the user product will be displayed on the rented products page.
- He can also go to the page where products are already rented by clicking the products option on the upper side of the page.
- There, he can select the product and request it by selecting the time period and location from the list of rented products.
- After the request is accepted, then he will get the rented product and he can use it for the period he requested for.

Now, below there are some snapshots of the project:

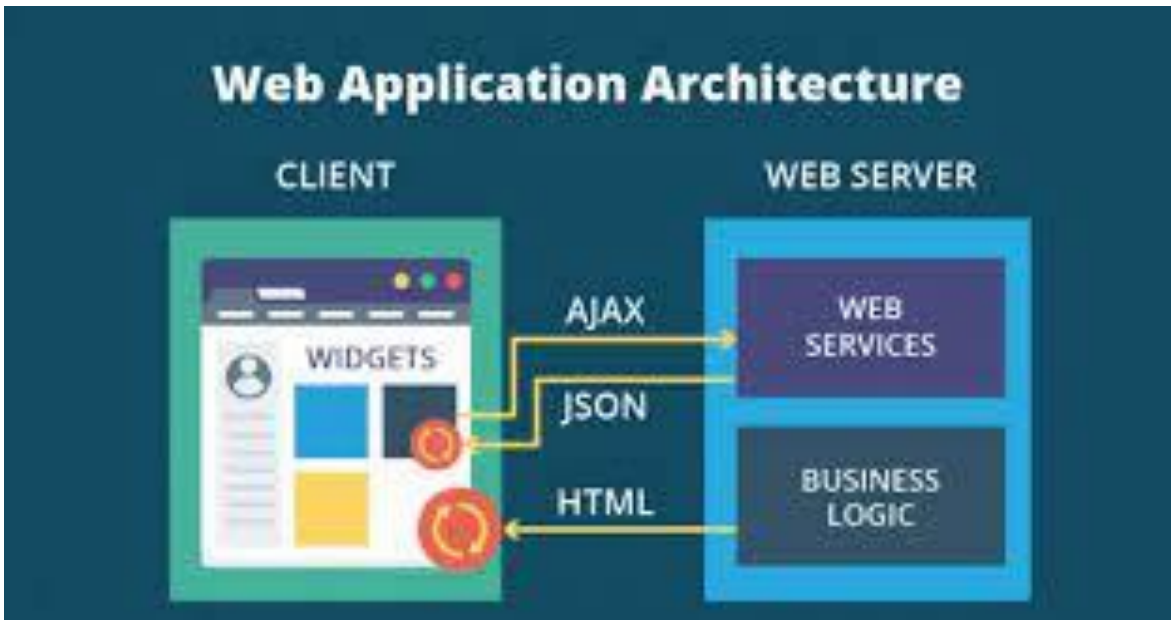

Fig 3.2: WebApp Architecture

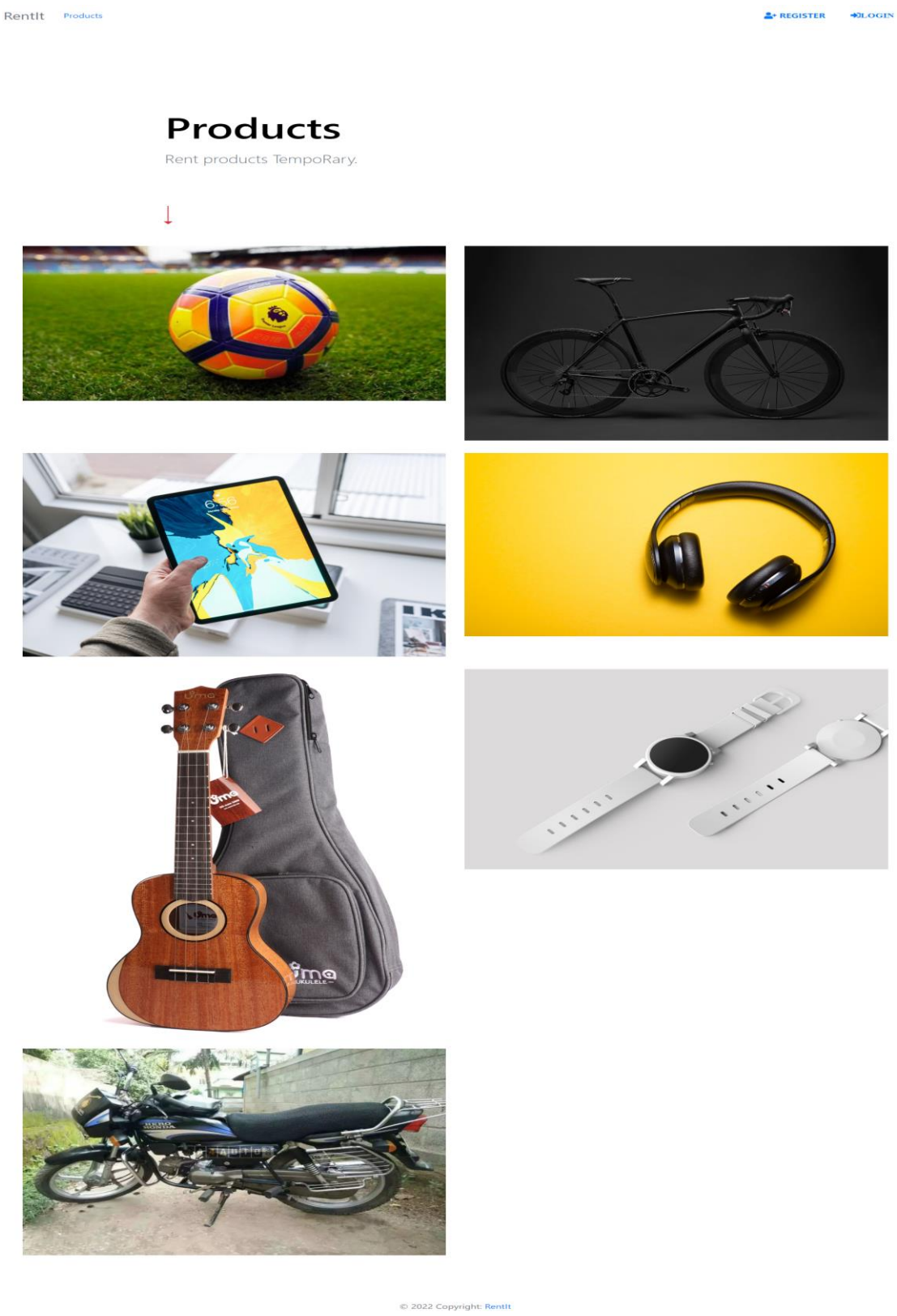

Fig 3.3: Home or Landing Page of RentIt WebApp

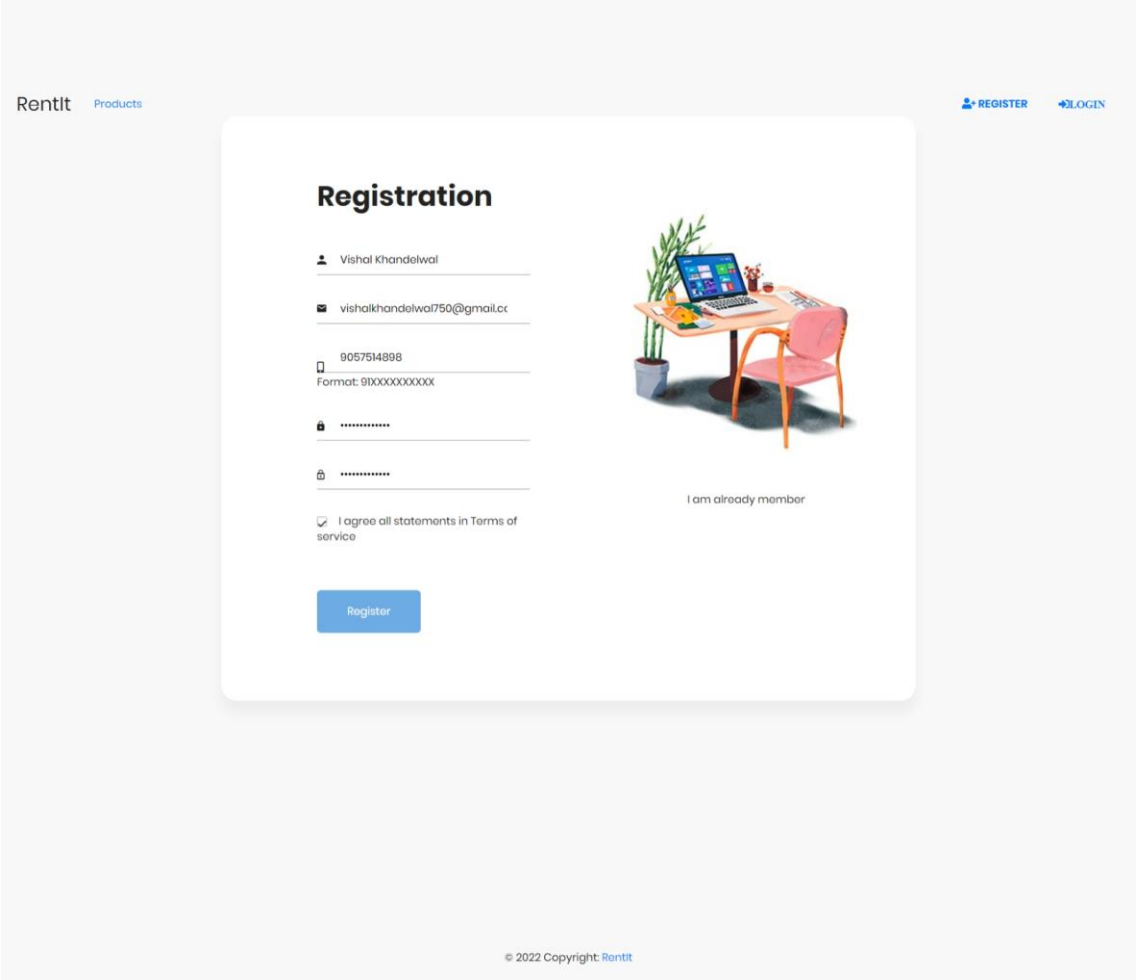

Fig 3.4: Registration of the User

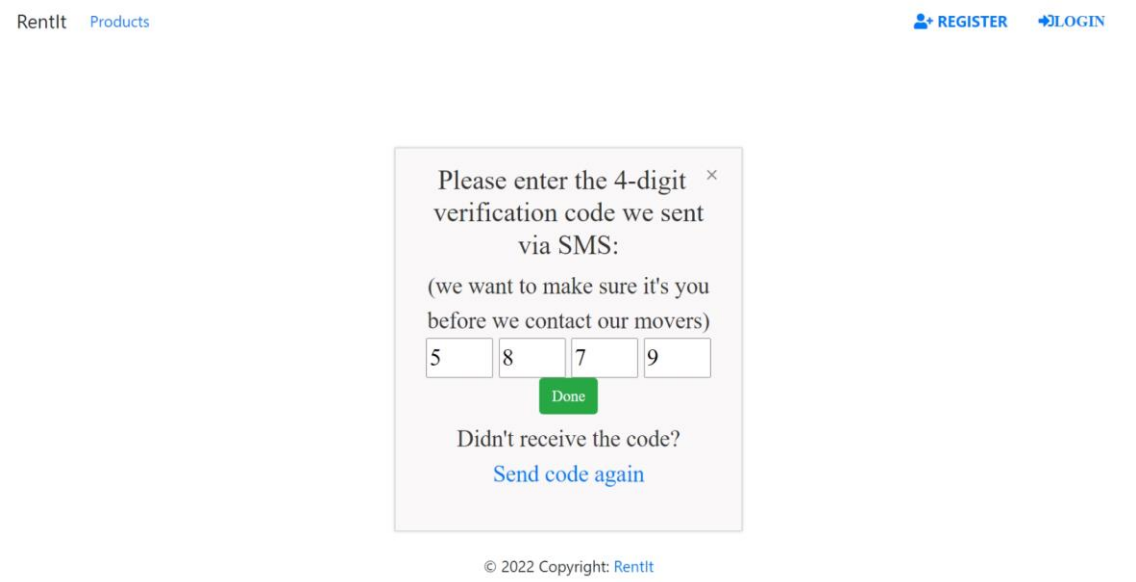

Fig 3.5: Verifying the User Through OTP

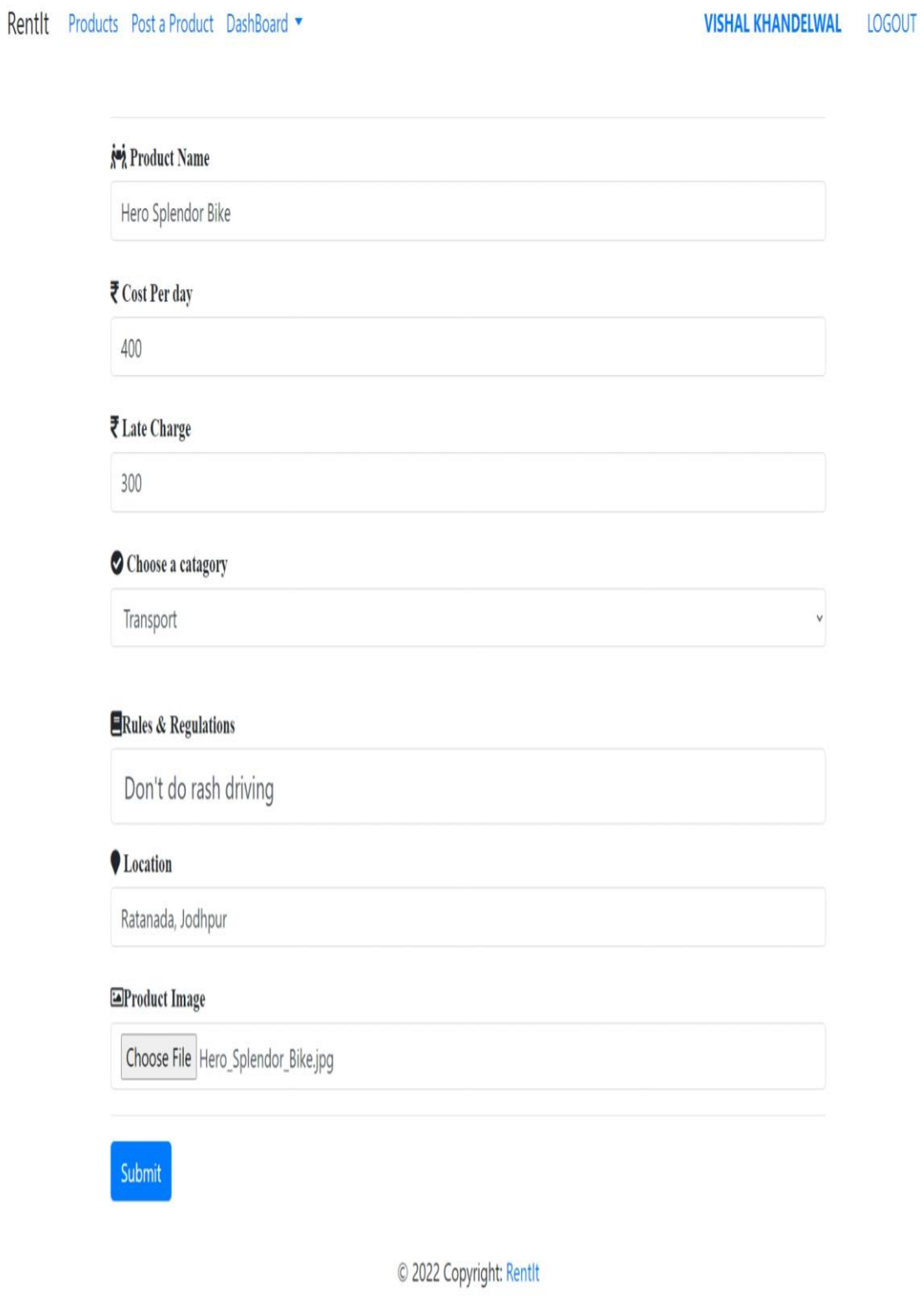

Fig 3.6: Posting A Product For Rent

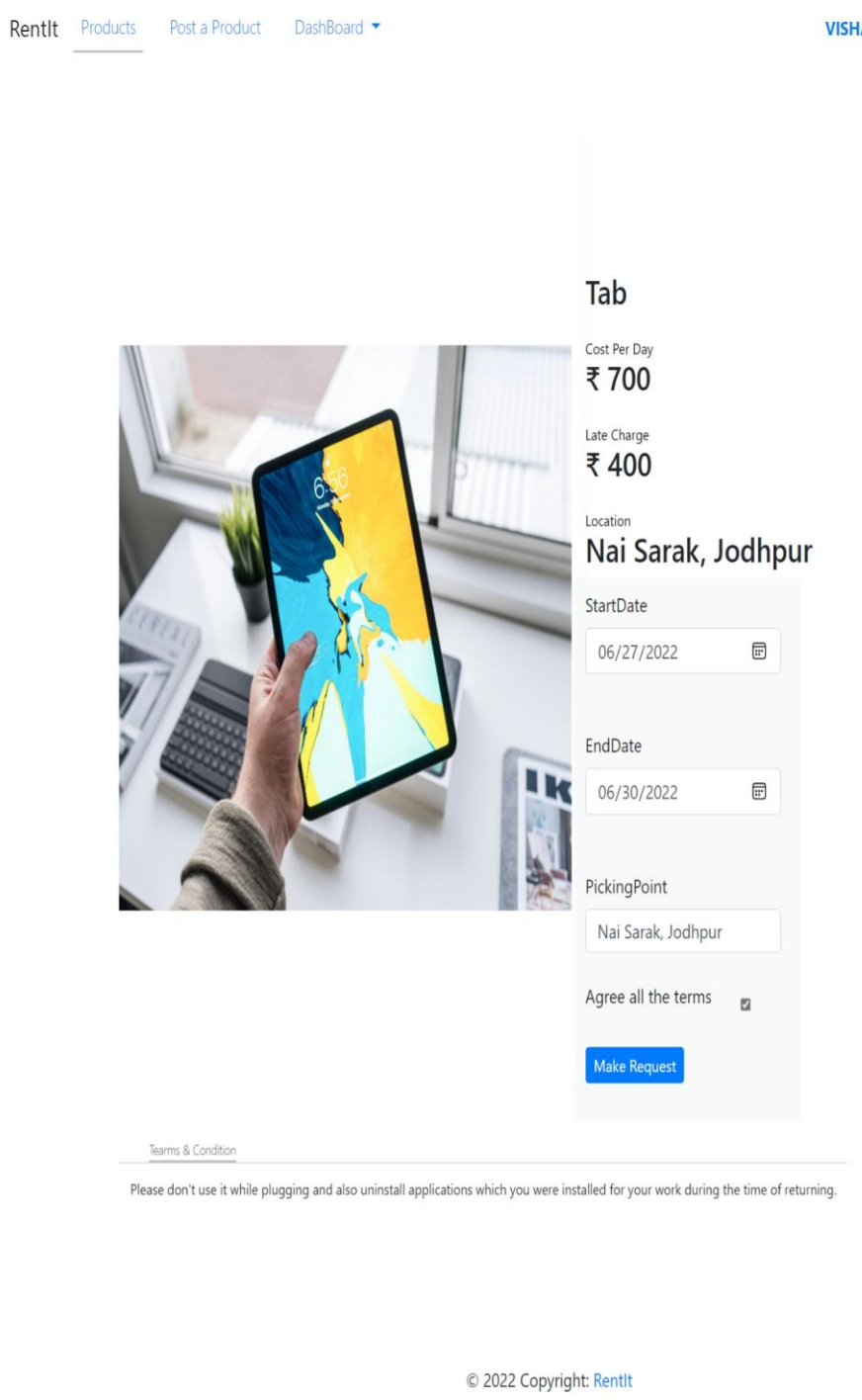

Fig 3.7: Making A Request For A Rented Product

This page was intentionally left blank.

Chapter 4

RESULTS / OUTCOMES

Throughout developing this major project we have added a lot to our existing knowledge. One new functionality we have learned about is how to send OTP to the user's phone for account verification through the integration of Twilio API. Twilio is a modern communication API Used by developers for establishing communications. Twilio can be used to send SMS, WhatsApp, Voice, Video, email, and even IoT, across the customer journey. All you need to do is integrate its API with your software.

We have also learned a website is different from a web app in many terms. The web application is designed for interaction with end users while the website contains static content. The user of the web application can read the content of the web application and also manipulate the data while the user of the website only can read the content of the website but not manipulate it. The web application site should be precompiled before deployment while the website does not need to be precompiled.

Integration is complex for web applications because of its complex functionality while integration is simpler for the website. Web applications need authentication, as they offer a much broader scope of options than websites while Authentication is not obligatory for informational websites. The user may ask to register to get a regular update or to access additional options. These features were not available for unregistered website visitors.

Finally, due to the development of this project, we understand we should give more focus on using rented items instead of buying a new product when the need for that product is for a limited duration. This will in turn reduce waste from our environment and also we'll save our money and invest that money in other things.

This page was intentionally left blank.

Chapter 5

CONCLUSION & FUTURE WORK

5.1 Conclusion

In a nutshell, users using this rental web app can solve their problem of purchasing new products whenever they are needed on any occasion and can use the rented products. It provides an experience similar to the e-commerce website to the users. It promotes the use of rented products to limit the stocking up of newly purchased items used for a limited duration.

5.2 Future Work

Needless to say, the technical aspects of the work we've done are not flawless and could be improved. In our existing web app, we can improvise to make it more advanced, easy to use, and efficient. Like, we can incorporate a payment gateway to do the payment, products recommendation based on their location, delivery service, and a chatting page for communication between clients. With acquiring more knowledge of concepts in the future, we can also incorporate live tracking of the delivery of the rented product.

This page was intentionally left blank.

References

[1] Web Development, Study Guide: Geeks For Geeks, https://www.geeksforgeeks.org/web-development/.

[2] HTML Tutorial, Study Guide: W3 Schools, https://www.w3schools.com/html/.

[3] CSS Tutorial, Study Guide: W3 Schools, https://www.w3schools.com/css/default.asp.

[4] JavaScript Tutorial, Study Guide: W3 Schools, https://www.w3schools.com/js/default.asp.

[5] What are web frameworks and why do you need them?, Study Guide: Intelegain Technologies, https://intelegain-technologies.medium.com/what-are-web-frameworks-and-why-you-need-them-c4e8806bd0fb.

[6] Notepad++, Study Guide: Computer Hope, https://www.computerhope.com/jargon/n/notepad-plus-plus.htm.

[7] Python – Overview, Study Guide: Tutorials Point, https://www.tutorialspoint.com/python/python_overview.htm.

[8] Why Django for Web Development?, Study Guide: Monocubed, https://www.monocubed.com/blog/django-web-framework/.

[9] Bootstrap, Study Guide: Geeks For Geeks, https://www.geeksforgeeks.org/bootstrap/.

[10] SQLite - Overview, Study Guide: Tutorials Point, https://www.tutorialspoint.com/sqlite/sqlite_overview.htm.

[11] What is Twilio and How Does it Work?, Study Guide: Tekki Web Solutions (TWS), https://www.tekkiwebsolutions.com/blog/what-is-twilio/.