

TIT-TAT

Project Report Submitted By

VIVIN V ABRAHAM

Reg. No. : AJC20MCA-2088

In Partial fulfillment for the Award of the Degree Of

**MASTER OF COMPUTER APPLICATIONS (2 Year)
(MCA)**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2020-2022

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, “**TIT-TAT**” is the bonafide work of **VIVIN V ABRAHAM (Reg. No. : AJC20MCA-2088)** in partial fulfillment of the requirements for the award of the Degree of Regular Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

Mr. Ajith G.S
Internal Guide

Ms. Grace Joseph
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner



AMAL JYOTHI COLLEGE OF ENGINEERING

Department of Computer Applications

CERTIFICATE ON PLAGIARISM CHECK

1	Name of the Scholar	VIVIN V. ABRAHAM
2	Title of the Publication	TIT-TAT
3	Name of the Guide	AJITH G.S
4	Similar Content (%) identified	15 %
5	Acceptable Maximum Limit	25%
6	Software Used	TURNITIN
8	No.of pages verified	49

*Report on plagiarism check result with % of similarity shall be attached.

Name & Signature of the Scholar:

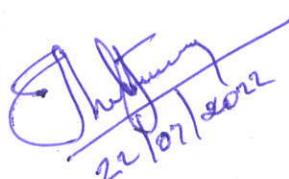
Vivin V. Abraham 

Name & Signature of the Guide:

AJITH G.S 

Name & Signature of the Head of the Department:

Head of the Department
Dept. of Computer Applications
Amal Jyothi College of Engineering
Koovalappally P. O., Kanjirapally - 686 518
Kerala, India


22/07/2022



DECLARATION

I hereby declare that the project report “**TIT-TAT**” is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Regular Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

Date: 25/07/2022

VIVIN V ABRAHAM

KANJIRAPPALLY

Reg. No. : AJC20MCA-2088

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Rev. Fr. Dr. Rubin Thottupurathu Jose** and **Ms. Grace Joseph** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Mr. Ajith G.S** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

VIVIN V ABRAHAM

ABSTRACT

The real power of this project is not just bought and sell something, but in the formation of stronger relationships with buyer and seller to delivering of a high level of service and support, which in turn improves organization services and its goodwill. The **TIT-TAT** Project is a software application which avoids buying and selling of used useful products on the basis of real money. This website keeps the data in a centralized way which is available to all the users at the same time. It manages historical data in database.

New thing that included in this project is that the customers can buy or sell products without any transaction of money in any means. And also, the customer's product is verified and evaluated and by an evaluator based on the details they provided. Customers can set their demanded product based on the value of their product with needs and specifications.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	CURRENT SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	SYSTEM PROPOSED	6
2.5	PROPOSED SYSTEM BENEFITS	7
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	10
3.1.3	BEHAVIORAL FEASIBILITY	10
3.2	DESCRIPTION OF THE SYSTEM	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	11
3.3.1	DJANGO FRAMEWORK	11
3.3.2	GOOGLE FIRESTORE	12
4	SYSTEM DESIGN	13
4.1	INTRODUCTION	14
4.2	UML DIAGRAM	14
4.2.1	USE CASE DIAGRAM	15
4.2.2	SEQUENCE DIAGRAM	17
4.2.3	STATE CHART DIAGRAM	19
4.2.4	ACTIVITY DIAGRAM	19
4.2.5	CLASS DIAGRAM	21
4.2.6	OBJECT DIAGRAM	22
4.2.7	COMPONENT DIAGRAM	23
4.2.8	DEPLOYMENT DIAGRAM	24
4.3	DESIGN OF USER INTERFACES	25

4.4	DATABASE DESIGN	31
5	SYSTEM TESTING	35
5.1	INTRODUCTION	36
5.2	TEST PLAN	37
5.2.1	UNIT TESTING	38
5.2.2	INTEGRATION TESTING	47
5.2.3	VALIDATION TESTING	47
5.2.4	USER ACCEPTANCE TESTING	48
6	IMPLEMENTATION	49
6.1	INTRODUCTION	50
6.2	IMPLEMENTATION METHODS	50
6.2.1	TRAINING OF USERS	51
6.2.2	APPLICATION SOFTWARE TRAINING	51
6.2.3	SYSTEM MAINTENANCE	51
6.3	HOSTING	52
7	CONCLUSION & FUTURE SCOPE	55
7.1	CONCLUSION	56
7.2	FUTURE SCOPE	56
8	BIBLIOGRAPHY	57
9	APPENDIX	59
9.1	SAMPLE CODE	60
9.2	SCREEN SHOTS	73
9.3	PLAGIARISM REPORT	82

List of Abbreviation

- IDE - Integrated Development Environment
- HTML - Hyper Text Markup Language.
- CSS - Cascading Style Sheet
- UML - Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The true power of this initiative is not in the purchase and sale of goods but in the development of better ties between buyer and seller to order to provide a high level of service and support, which increases the organization's services and goodwill. The TIT-TAT Project is a software application that eliminates the use of real money in the purchasing and selling of secondhand usable things. This website maintains the data in a central location that is simultaneously accessible to all users. It oversees the database's historical data.

New thing that included in this project is that the customers can buy or sell products without any transaction of money in any means. And also, the customer's product is verified and evaluated and by an evaluator based on the details they provided. Customers can set their demanded product based on the value of their product with needs and specifications.

1.2 PROJECT SPECIFICATION

The proposed system is a website in which user can bought and sell used useful products online. And also, that the product is clearly verified and evaluated by an evaluator thus the product gets real value. Due this, customer not loss the value of the product by any means and can buy a product similar to the evaluated value. After the evaluation the value gets turned to coins.

The system includes 4 modules. They are:

1. Admin Module

Admin must have a login into this system. Admin has access to every detail stored in the system. He can also eliminate fraud users, he can approve the product, and publish them. And he can approve the advertisements and receive the charges for the advertising

2. Evaluator Module

Evaluator have a login into this system. He has done the overall verification and evaluation of the available of products in this system. Based on the evaluation only the product gets displayed for selling purpose. After the evaluation the value gets turned to coins.

3. Customer Module

Customer can register and share the details of the products he wants to sell with some major specification and also mention what product he needs with major specification with similar value of the customer's product. Customer can update, remove the product details and also search product based on categories and product name.

4. Ads Customer Module

According to specific plans, the ads customer can register and exchange the specifics of adverts. And Ads Customer should pay the payment then only his advertisement gets published.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

The process of doing a system analysis involves acquiring and analyzing data, diagnosing issues, and using the information to suggest system changes. Between system users and system developers, it is a problem-solving activity that necessitates extensive communication. In any system development process, system analysis or study is a crucial step. In-depth research and analysis are done on the system. System analysts act as interrogators and delve deeply into how the current system functions. The input to the system is identified, and the system as a whole is viewed. The results of the various processes are identified. Understanding the issue, identifying the pertinent and important variables, evaluating and synthesizing the many elements, and selecting the best or, at the very least, most acceptable course of action are all part of system analysis.

It is necessary to thoroughly examine the process using a variety of methods, including questionnaires and interviews. To reach a conclusion, it is necessary to carefully examine the data gathered by various sources. An understanding of the system's operation serves as the conclusion. The existing system is what we refer to as. Problem areas are now found after a comprehensive examination of the current system. The designer now acts as a problem-solver and makes an effort to resolve the issues the business is currently experiencing. The solutions are offered as suggestions. The best suggestion is then chosen after being analytically compared to the current system. The user is presented with suggestion in order to receive their approval. The idea is examined in response to user requests, and necessary revisions are made. The user's satisfaction with the proposal marks the conclusion of this cycle.

In order to do additional research on the system, preliminary investigation involves acquiring and interpreting data. A problem-solving process known as preliminary research necessitates close coordination between system users and developers. Many feasibility studies are conducted by it. From these studies, it is possible to determine an approximate estimate of the system activities, from which it is possible to make a decision regarding the methods to use for an efficient system study and analysis.

2.2 CURRENT SYSTEM

The current system does not completely eliminate the need for cash when purchasing and selling goods. Customers may register and sell their products solely in exchange for money, and only another client who has sufficient funds may purchase the product. Each client can set up their own profile to buy or sell goods. The shortcomings of the current system are addressed by the suggested system.

The current system must be altered in order to include new data, increase its efficiency, and make it more adaptable and secure. Customers can buy or sell goods using the new method even if they don't have any money.

2.3 DRAWBACKS OF EXISTING SYSTEM

- No proper online management of system
- Cannot valued properly
- It is difficult to search particular product with his proper need.
- If no cash no product can buy if customer have a useless product.

2.4 SYSTEM PROPOSED

The suggested approach is designed to eliminate all the drawbacks of the current one. In order to increase the growth of brought and sold products, it is vital to have a system that is more user-friendly and user-attractive. In our proposed system there is admin who can view all the customers and products. Evaluation and verification of the product and customer is done by the evaluator. It allows customers to show case their product with proper specification and demand a product based on the value of the product. Users of this proposed system are admin, evaluator, customer and Ads customer. The software application which avoids buying and selling of used useful products on the basis of real money. This application centrally stores the data so that it is simultaneously accessible to all users. Using a database to manage historical data is relatively simple. The distributors can utilise this application with no special training. They have simple access to the products and can purchase them for no cost. The process of entering sales and purchase data for a product into databases is extremely simple. The Owner makes money from the advertisements that are displayed on the website.

2.5 PROPOSED SYSTEM BENEFITS

The system is very easy to create and put into operation. The system works in practically any set up and calls for very little in the way of system resources. It has the following attributes:

➤ **Enhanced security: -**

Unauthorized access must be prevented in order for data to stay secure. Data protection means that they are shielded against various types of erasure. Security, integrity, privacy, and confidentiality are the four connected problems that make up the system security issue. Security is maintained by requiring a username and password to log in. Since we use secured databases to maintain the papers, it will also ensure data security.

➤ **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

➤ **Better service: -**

The product will avoid the burden of hard copy storage. For performing the same activity, we can also save time and resources. The data can be kept for a longer time without losing any information.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is conducted to determine whether the project, once completion, will serve the organization's goal for the amount of work, effort, and time invested in it. A feasibility study enables the developer to predict the project's usefulness and potential future. A system proposal's workability, which includes the influence on the organisation, capacity to satisfy user needs, and efficient use of resources, is the basis for a feasibility study. As a result, before a new application is accepted for development, it often undergoes a feasibility assessment.

The document outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility study, including its technical, economic, and operational viabilities. It has the following characteristics: -

3.1.1 Economical Feasibility

Cost and benefit analyses are required to support the emerging system. criteria to make sure that focus is placed on the project that will yield the best results the earliest. The price that would be involved in developing a new system is one of the variables.

Some significant financial queries raised during the initial probe include the following:

- These expenses carry out a comprehensive system investigation.
- Costs associated with the hardware and software.
- Benefits in the form of decreased expenditures or fewer expensive errors.

There are no manual costs associated with the suggested system because it was developed as part of a project. Additionally, the system's ability to develop economically is indicated by the fact that all resources are already available.

The project's cost, was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical Feasibility

The system needs to be assessed first from a technical standpoint. An overview design of the system's requirements in terms of input, output, programmes, and procedures must serve as the foundation for the assessment of this viability. The inquiry must next advise the kind of equipment, necessary procedure for constructing the system, and means of operating the system once it has been designed after having identified an outline system.

Technical issues raised during the investigation are:

Does the existing technology sufficient for the suggested one?

- Is the suggested technology compatible with the current technology?
- Can the system grow if it is improved?

The project should be designed in such a way that the required performance and functionality are met while still adhering to the limits. The research makes use of cryptographic methods and a high resolution scanning gadget. Because newer versions of the same software still work with earlier versions, the system can continue be used even though the technology may become outdated after a while. The project is therefore only subject to a few limitations. Technically speaking, the project can be developed because the system was created utilising Python for the front end and a Google Firestore server for the back end. Technically speaking, the project can be developed because the system was created utilising Python for the front end and a Google Firestore server for the back end. The project is technically viable because the system was created with Google Firestore in the back end and Python in the front end. Additionally, the system's Intel i5 core processor, 8GB of RAM, and 512 GB SSD hard drive provided good performance.

3.1.3 Behavioral Feasibility

The following inquiries are part of the suggested system:

- Is there enough assistance for the users?
- Will the suggested system harm anyone?

The project would be advantageous because, when created and implemented, it would achieve the goals. The project is deemed to be behaviorally feasible after carefully weighing all behavioral factors.

3.2 DESCRIPTION OF THE SYSTEM

3.2.1 Hardware Specification

Processor - Core i5 from Intel

RAM - 8 GB

Hard drive - 512 GB SSD

3.2.2 Software Specification

Front End - HTML, CSS, python

Backend - Google firestore

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, JQuery, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 Django Framework

A Python-based web framework called Django enables you to easily build effective online apps. Django is characterized as a "batteries-included" framework since it provides the built-in capability for everything, including the Django Admin Interface and the default database, SQLlite3. A similar collection of components is typically necessary when creating a website: a mechanism for managing user authentication (signing up, signing in, and signing out), a management panel for your website, forms, a method for uploading files, etc. Django offers ready-made parts that can be used for rapid development.

Why Use the Django Framework?

- Very good scalability and very good documentation.
- Django is used by well-known MNCs and businesses including Instagram, Disqus, Spotify, Youtube, Bitbucket, and Dropbox..
- Django is the simplest Framework to comprehend, has rapid development, and fully integrated Batteries.
- The last but certainly not least reason to learn Django is Python, a programming language with a sizable library and features like web scraping, machine learning, image processing,

scientific computing, etc. All of this may be combined with a web application to perform a number of complex tasks.

3.3.2 Google Firestore

- Designed for great performance, automatic scaling, and simple application development, Firestore is a NoSQL document database. Despite sharing many of the same capabilities as conventional databases, the Firestore interface is unique because it is a NoSQL database and so describes connections between data items differently.
- A NoSQL, document-oriented database is Firestore. There are no tables or rows, in contrast to a SQL database. Instead, you keep your data as documents that are grouped together into collections.
- A maintenance-free, serverless document database that can easily scale to handle any demand.
- Direct database connectivity will speed up the creation of mobile, web, and IoT applications.
- Real-time application development is made simple by built-in offline mode and live synchronisation.
- Data protection measures that are completely adjustable ensure that the data is always secure.
- Seamless integration of Google Cloud services like BigQuery and Cloud Functions with Firebase.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

For each technical system or product, design is the initial step in the development phase. Design involves creativity. Effective systems depend on good design. In order to specify a process or a system in sufficient depth to enable its physical embodiment, the term "design" is defined as "the act of applying numerous approaches and concepts." To specify a device, method, or system in sufficient depth to allow for its physical embodiment, it may be characterised as the process of applying numerous techniques and principles. Software design is used regardless of the development paradigm that is employed and forms the technical core of the software engineering process. A system or product's architectural details are developed as part of the system design. This programme, like any systematic technique, has undergone the best design phase possible, fine refining all efficiency, performance, and accuracy levels. A user-oriented document is transitioned during the design phase into a document for programmers or database staff. Logical and physical design are the two stages of system development.

4.2 UML DIAGRAM

Specifying, visualising, creating, and documenting the artefacts of software systems is all done using the standard language known as UML. The Object Management Group (OMG) was responsible for developing UML, and in January 1997, a draught of the UML 1.0 definition was presented to the OMG.

Unified Modeling Language, or UML, is its name. Unlike other popular programming languages like C++, Java, COBOL, etc., UML has some unique features. Software blueprints are created using the graphical language UML. UML is a general-purpose visual modelling language that is used to design, specify, build, and document software systems. Even while UML is typically used to represent software systems, it is not constrained by this definition. In addition, it is applied to modelling non-software systems. As an illustration, consider the manufacturing facility's process flow. Although UML is not a programming language, tools that employ UML diagrams can produce code in a number of different languages. Object oriented analysis and design are directly related to UML. UML has been standardised and is now an OMG standard. An entire

UML diagram representing a system is created using all the elements and linkages. The most crucial aspect of the entire process is how the UML diagram looks. To finish it, all the additional components are used. These nine diagrams are included in UML.

- Use case diagram
- Sequence diagram
- State chart diagram
- Activity diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

4.2.1 USECASE DIAGRAM

A use case diagram is a visual representation of how a system's components interact. Use cases are an approach in system analysis that help to find, explain, and arrange system needs. The term "system" in this context refers to something being created or run, such as a website for mail-order product sales and services. In UML (Unified Modeling Language), a common notation for the modelling of actual objects and systems, use case diagrams are used.

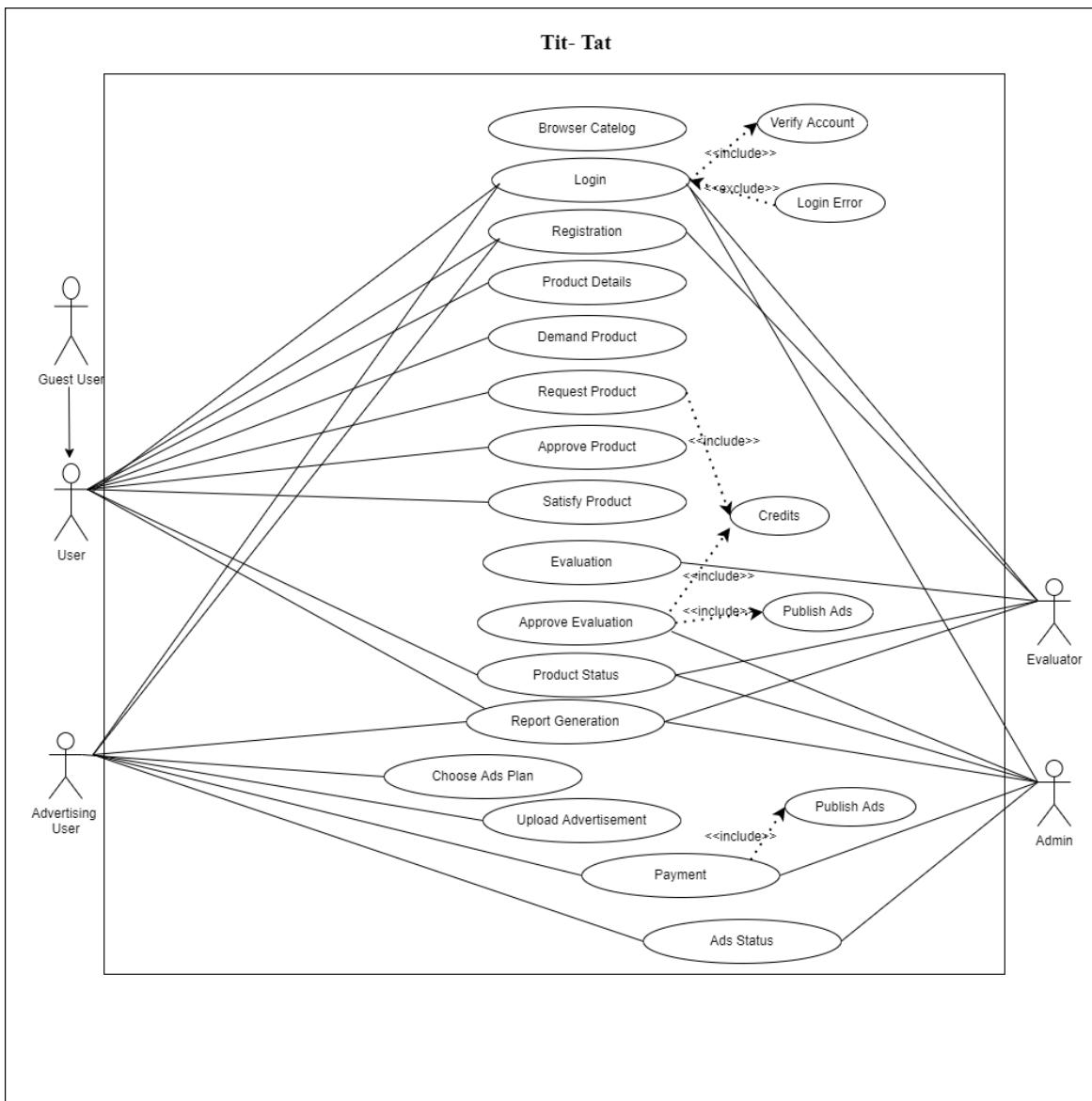
The planning of overall requirements, validating a hardware design, testing and debugging a software product in development, producing an online help reference, or carrying out a customer-service-focused activity are some examples of system objectives. Items ordering, catalogue updates, payment processing, and customer relations are a few examples of use cases in a product sales system. Four elements make up a use case diagram.

- The border, It describes the system of interest in terms of its environment.
- The actors are often people who are a part of the system and are classified by the functions they play.
- The actors within and around the system perform the use cases, which are the specialised roles.
- The connections and interdependencies between the actors and use cases.

To depict a system's functional needs, use case diagrams are created. To create a successful use case diagram after identifying the aforementioned things, we must follow the principles below.

- A use case's naming plays a crucial role. The name should be selected so that it can specify the functionalities carried out.
- Give actors a name that fits them.
- In the diagram, make obvious all links and dependencies.
- Since the primary goal of the diagram is to define the needs, avoid attempting to include all possible relationships.
- To further explain some crucial issues, take notes wherever necessary

USECASE DIAGRAM



4.2.2 SEQUENCE DIAGRAM

The interactions between items are simply shown in a sequential order, or the order in which they occur, in a sequence diagram. The words "sequence diagram" can also apply to event diagrams or event scenarios. Sequence diagrams show the actions that the system's components take and the sequence in which they occur. These diagrams are frequently used by businesspeople and software developers to document and comprehend requirements for new and current systems.

Sequence Diagram Notations –

- i. **Actors** – A type of role that interacts with the system and its objects is represented by an actor in a UML diagram. An actor is always outside the purview of the system that the UML diagram is intended to represent. To portray a variety of roles, including those of human users and other external subjects, we employ actors. Stick person notation is used to represent an actor in a UML diagram. A sequence diagram can have several actors.
- ii. **Lifelines** – A named piece that shows a specific participant in a sequence diagram is called a lifeline. In essence, a lifeline represents each incident in a sequence diagram. The lifeline components in a sequence diagram are at the top.
- iii. **Messages** – The use of messages illustrates communication between objects. On the lifeline, the messages are displayed in order. Arrows are how we depict messages. A sequence diagram is built around lifelines and messages.

The following categories serve as a general breakdown of messages:

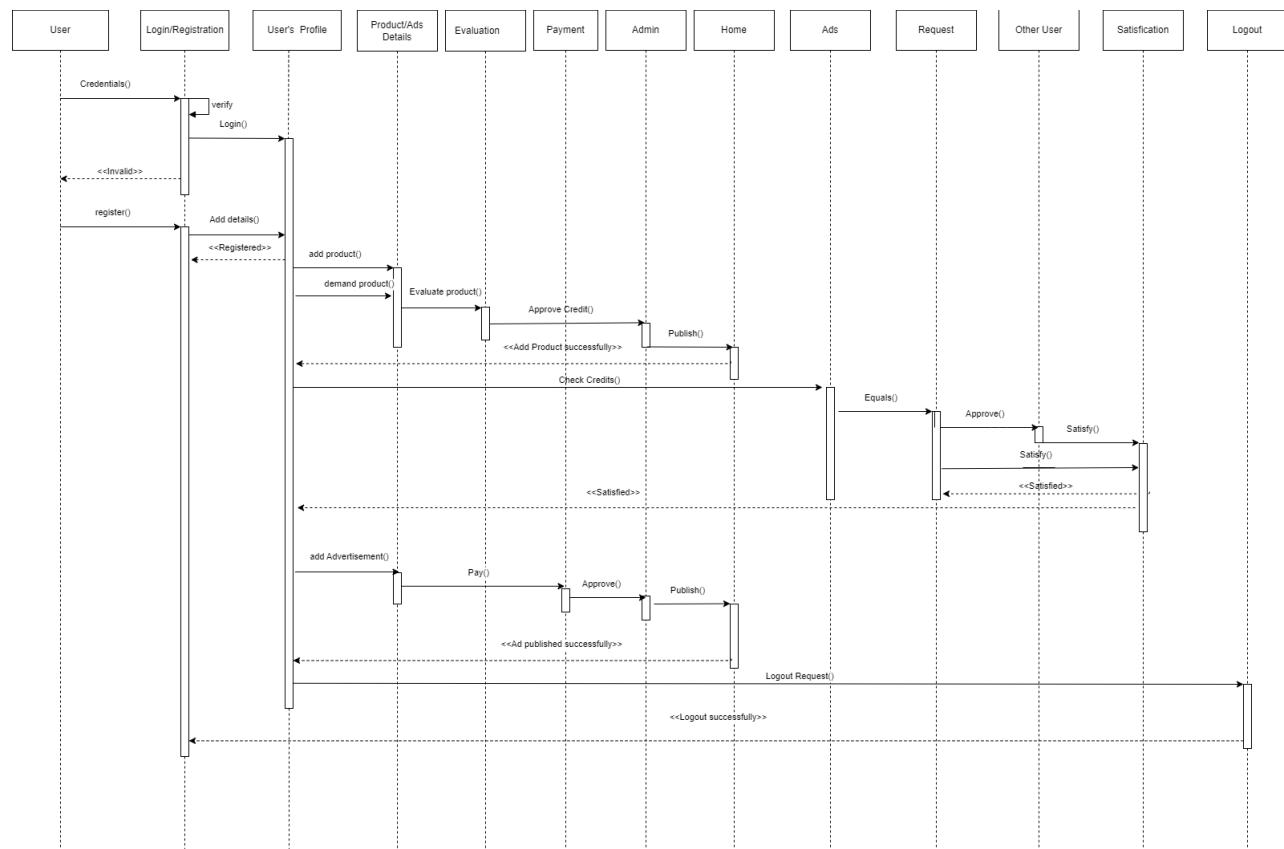
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

iv. Guards –In UML, guards are used to model circumstances. They are employed when we need to impose a restriction on the flow of messages under the guise of a fulfilment of a condition. Guards are crucial in informing software developers of the limitations imposed by a system or specific procedure.

Sequence diagram applications –

- Utilized to model and visualize the reasoning behind a complex function, operation, or process.
- They are also employed to display the specifics of UML use case diagrams.
- Used to comprehend the precise operation of present or future systems.
- Consider the movement of tasks and communications among system's many elements

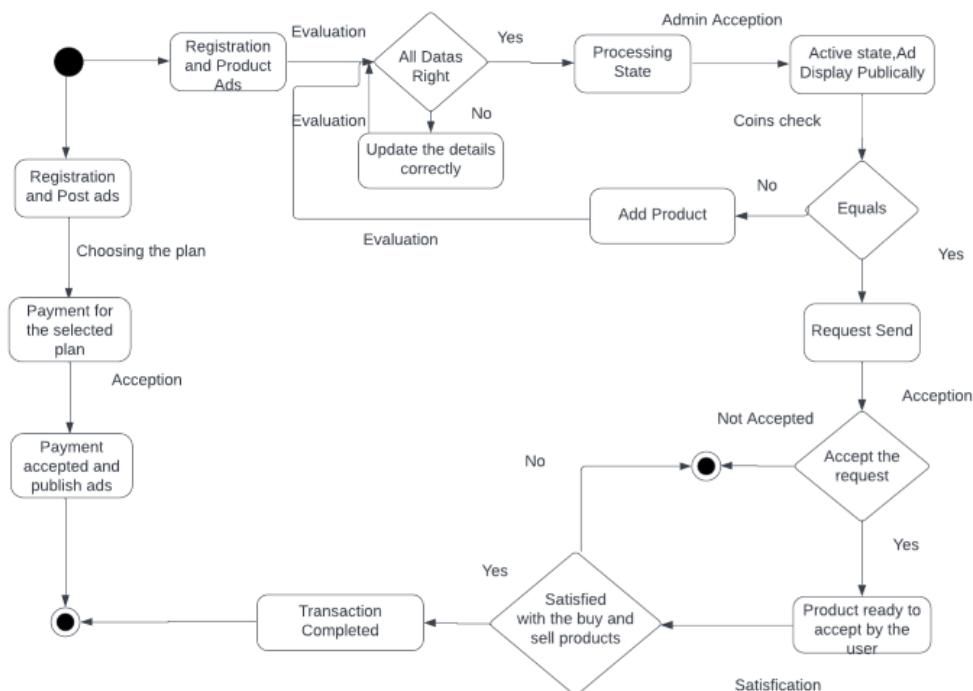
SEQUENCE DIAGRAM



4.2.3 STATE CHART DIAGRAM

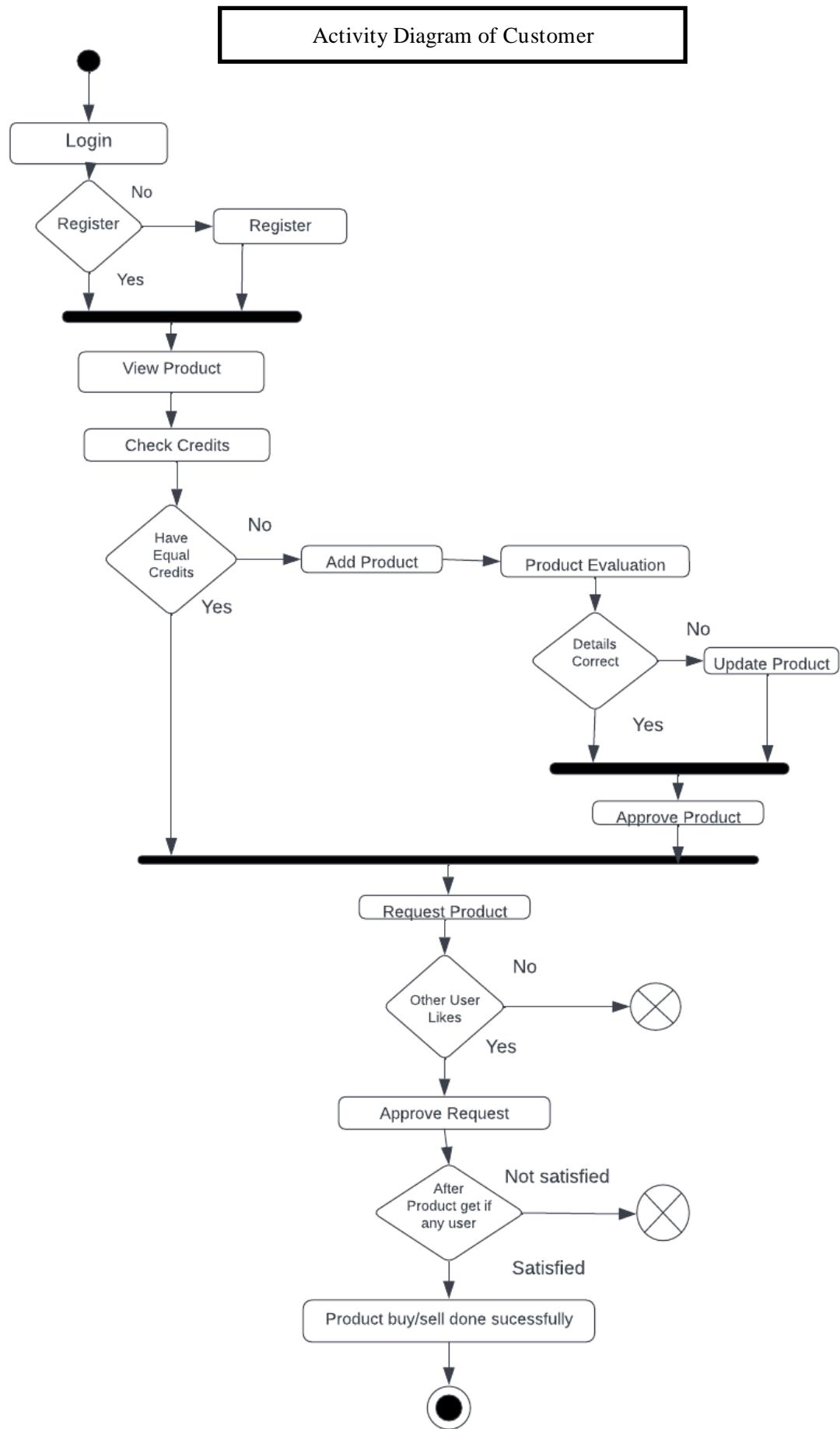
The behaviour of a software system is depicted using state diagrams. The behaviour of a class, a subsystem, a package, or even an entire system can be modelled using UML state machine diagrams. State Transition Diagram or a Statechart are other names for it. Statechart diagrams give us an effective approach to represent the communications or interactions that take place between external entities and a system. These illustrations are employed to represent the event-based system. An event is used to control the status of an item. The many states of an entity within the application system are described using statechart diagrams.

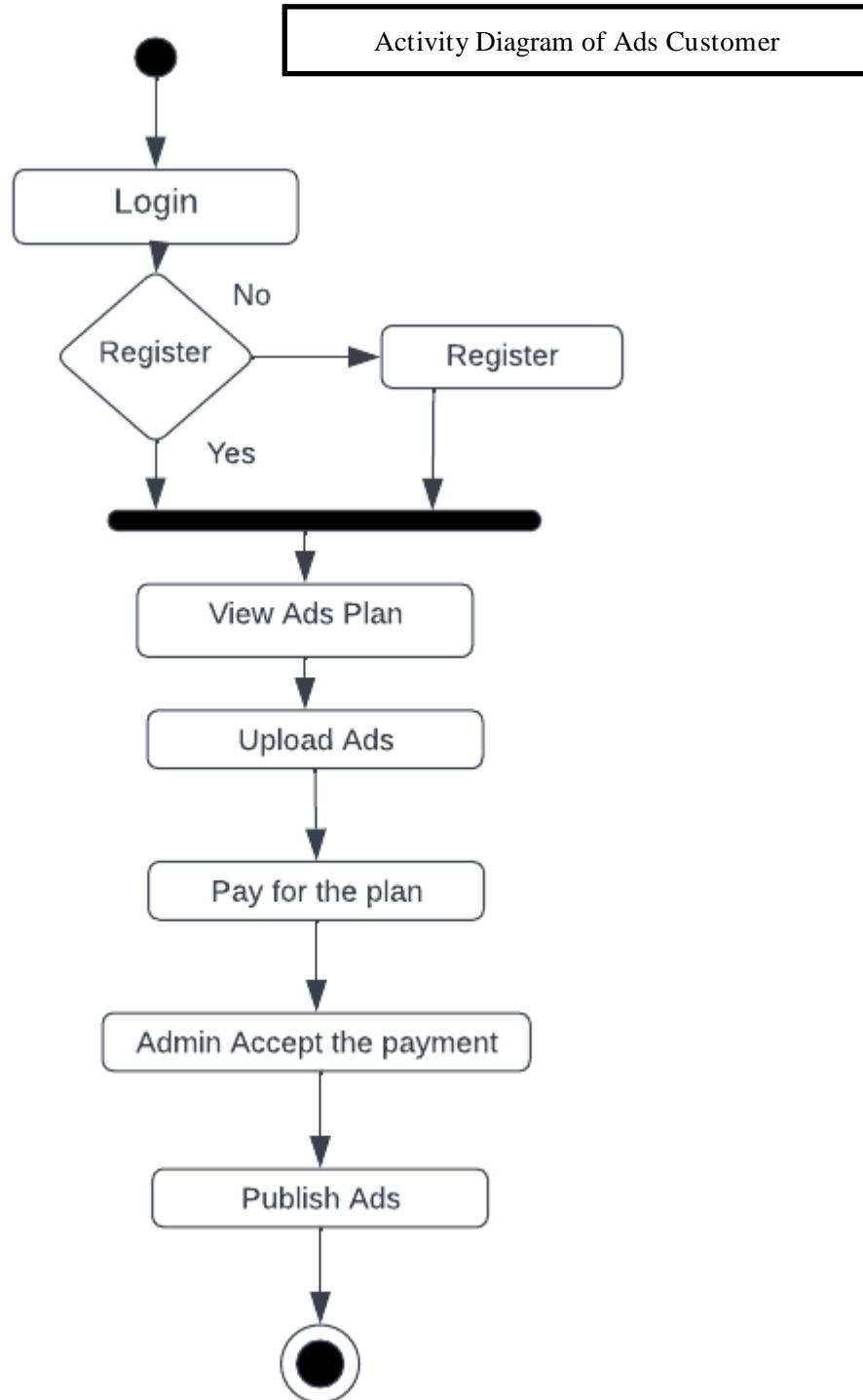
State- Chart Diagram



4.2.4 ACTIVITY DIAGRAM

Using several levels of abstraction, activity diagrams show how activities are organised to create a service. It is also appropriate for simulating how a set of related use cases operate together to reflect business workflows.

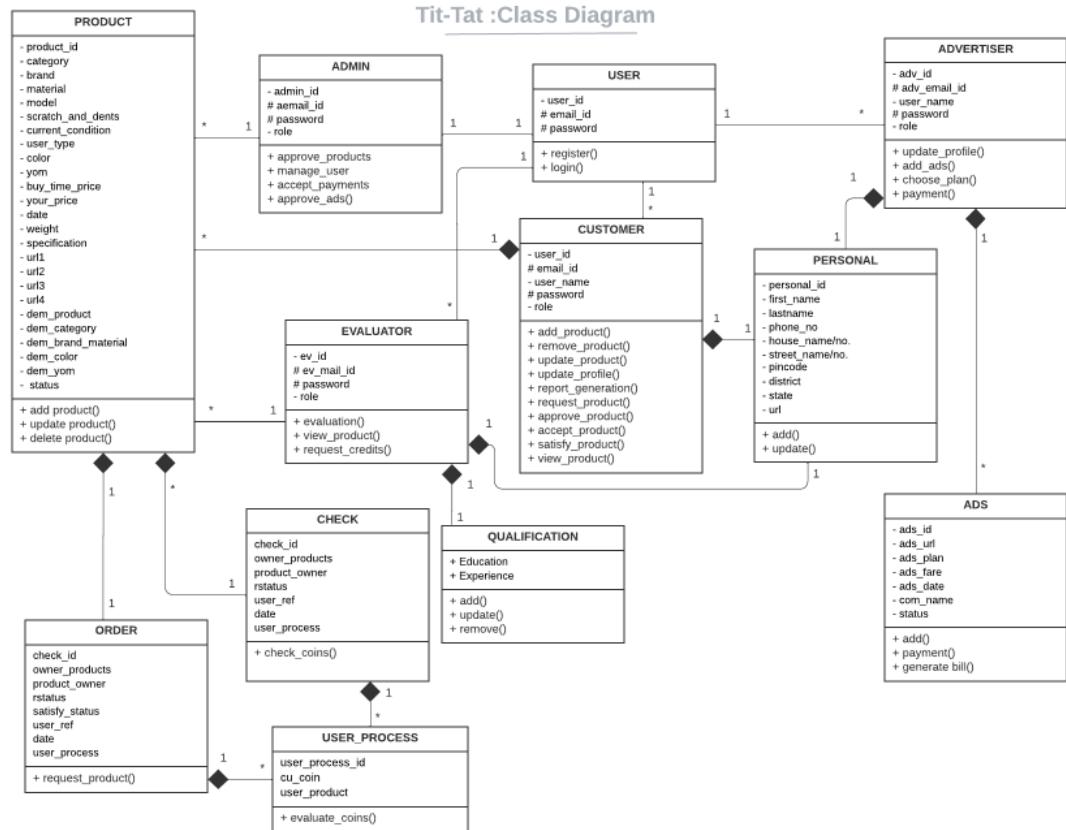




4.2.5 CLASS DIAGRAM

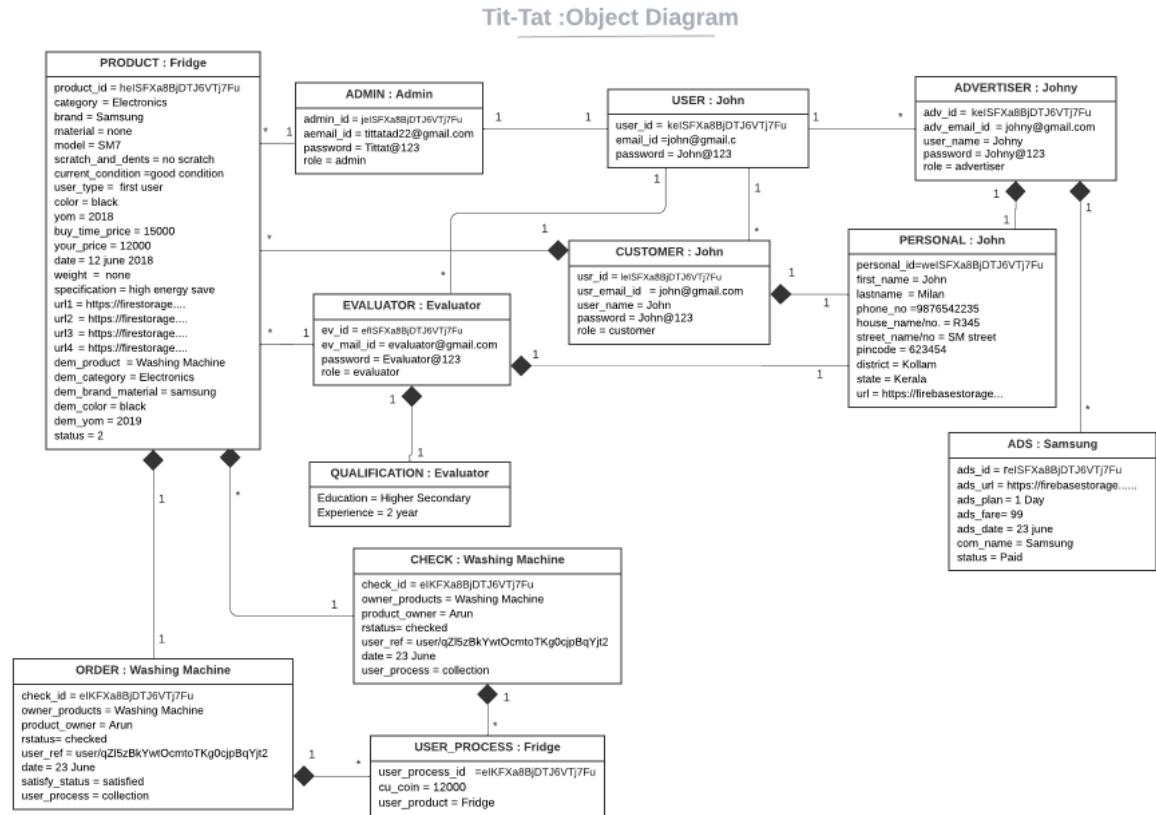
A class diagram is a static diagram. It represents a static view of an application. Class diagrams are used to create executable code for software applications as well as for visualising, explaining, and documenting various system components. Class diagrams outline a class's characteristics, methods, and restrictions on the system. Because they are

the only UML diagrams that can be directly transferred to object-oriented languages, class diagrams are extensively utilised in the modelling of object oriented systems. A class diagram displays a number of classes, interfaces, affiliations, collaborations, and constraints. It is also referred to as a structural diagram.



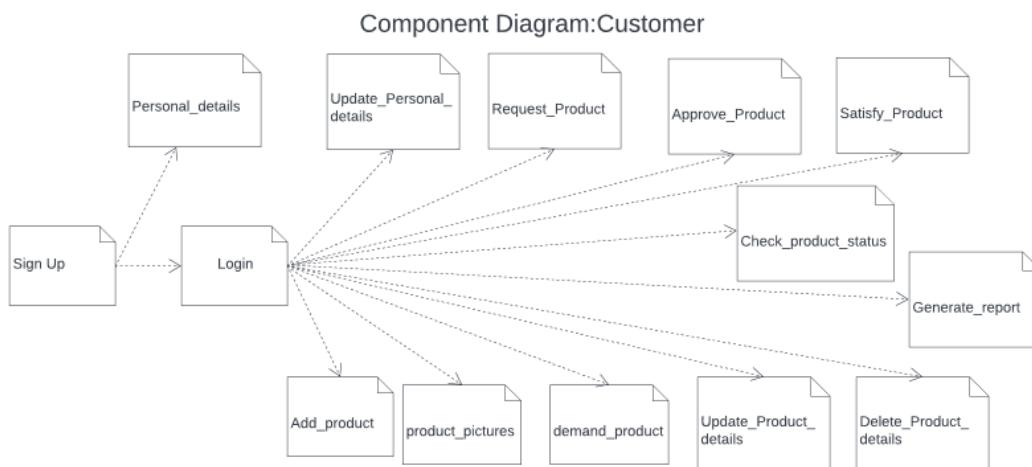
4.2.6 OBJECT DIAGRAM

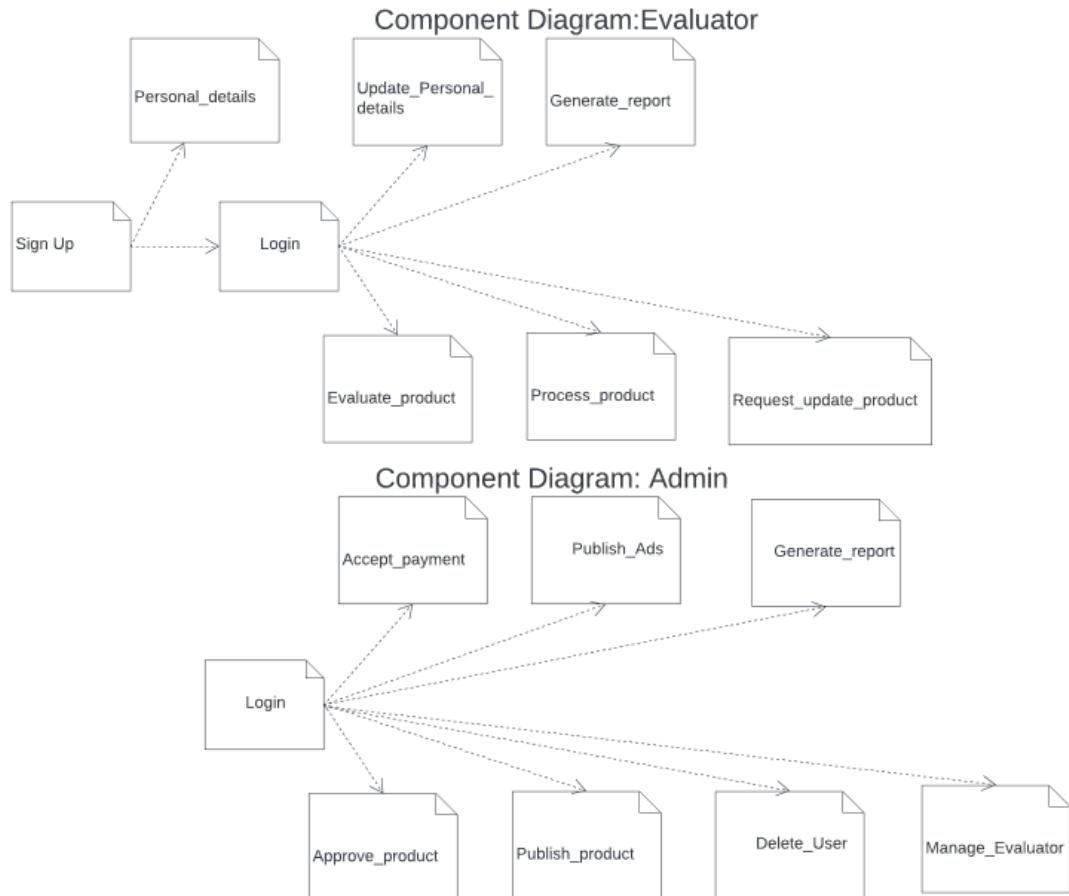
Since class diagrams are the source of object diagrams, class diagrams are a prerequisite for object diagrams. An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system. To represent a group of items and their connections as an instance, object diagrams are employed.



4.2.7 COMPONENT DIAGRAM

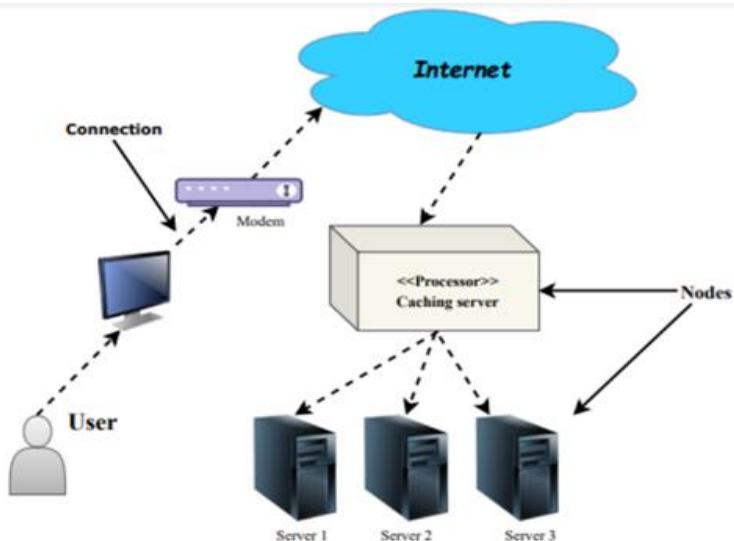
Component diagrams have different behaviours and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how a system's components are connected and arranged.





4.2.8 DEPLOYMENT DIAGRAM

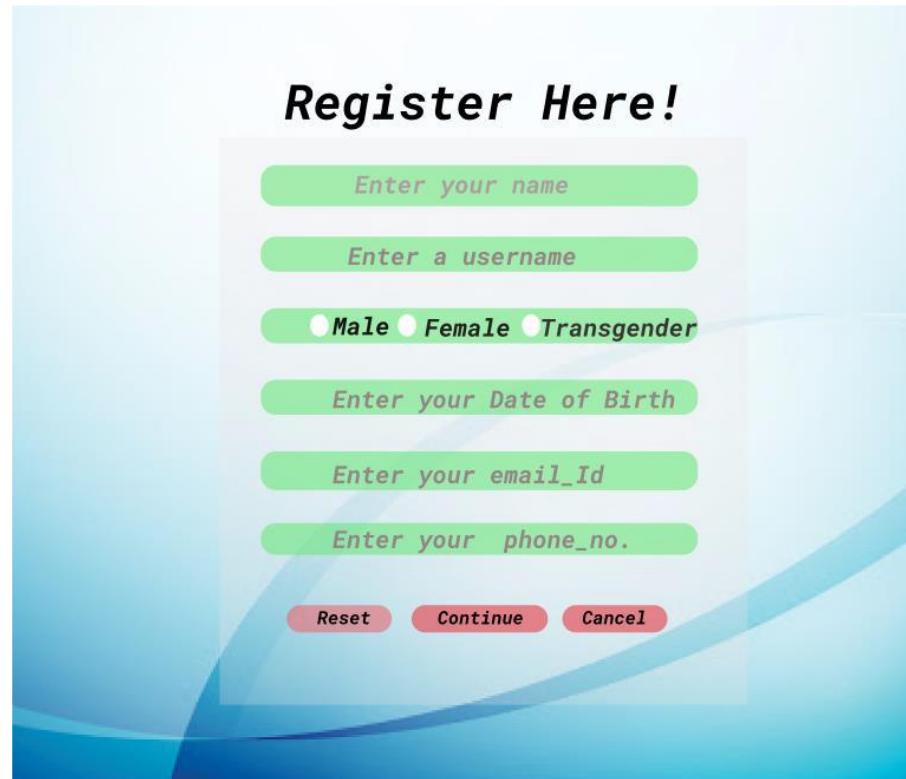
An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, which is a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system.



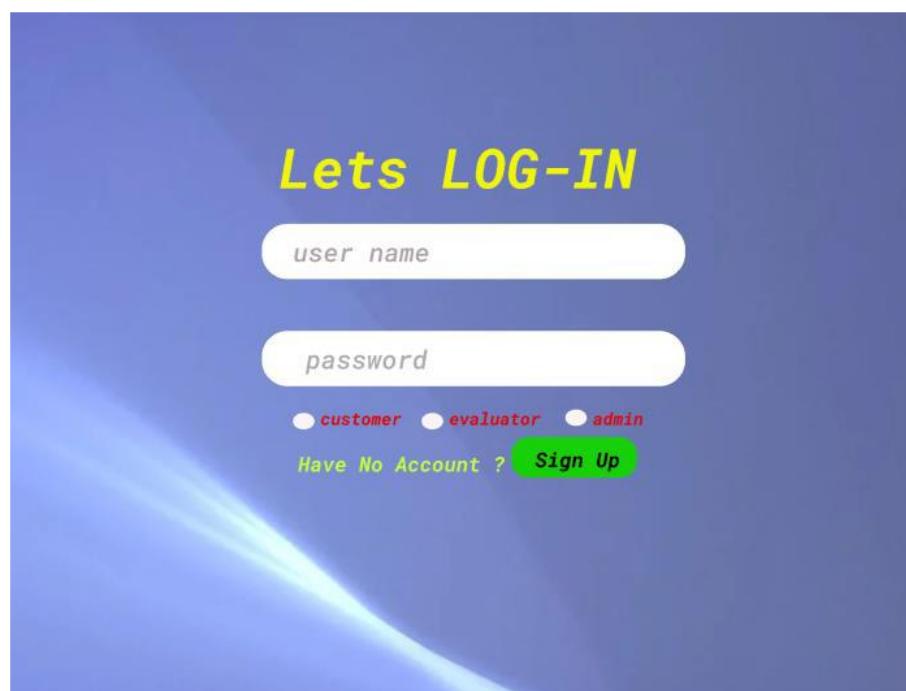
4.3 DESIGN OF USER INTERFACES USING FIGMA

4.3.1-INPUT DESIGN

Form Name: User Registration



Form Name: User Login



Form Name : Add Product

Add Product:

<i>Product Name</i>	<input type="text"/>
<i>Category Name</i>	<input type="text"/>
<i>Brand</i>	<input type="text"/>
<i>Color</i>	<input type="text"/>
<i>Year of Manufacture</i>	<input type="text"/>
<i>Specifications</i>	<input type="text"/>
<i>Condition</i>	<input type="text"/>
<i>Expected Market Value</i>	<input type="text"/>

Save Continue Clear

Form Name : Demand Product

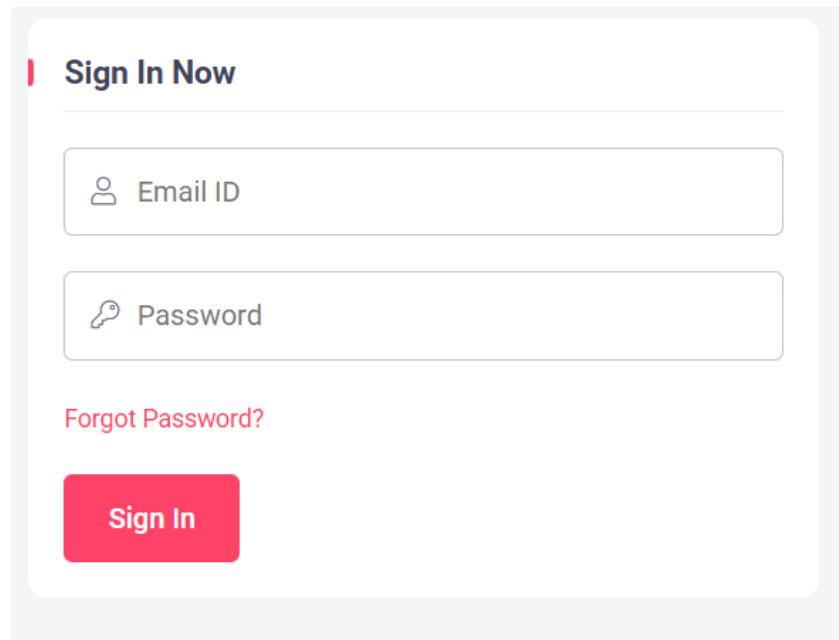
Demand Product:

<i>Product Name</i>	<input type="text"/>
<i>Category Name</i>	<input type="text"/>
<i>Brand</i>	<input type="text"/>
<i>Color</i>	<input type="text"/>
<i>Year of Manufacture</i>	<input type="text"/>
<i>Specifications</i>	<input type="text"/>
<i>Credits</i>	<input type="text"/>

Save Search Clear

4.3.2 OUTPUT DESIGN

User Login



The form is titled "Sign In Now". It contains two input fields: "Email ID" with a user icon and "Password" with a key icon. Below the fields is a link "Forgot Password?". A large red "Sign In" button is at the bottom.

User Registration



The form is titled "Contact Details". It includes a placeholder for a profile picture with a black silhouette. Below it are file upload controls ("Choose File", "No file chosen", "Upload"). The form then lists several input fields: "First Name*", "Last Name*", "Phone*", "House Name/House No.", "Street Name/Street No.", "Pincode", "Select District" (a dropdown menu), "Select State" (another dropdown menu), and a large red "Submit" button at the bottom.

Add Product

Product Details

Category :Computer & Accessories

Product

brand

Model

Scratch and Dents if any

Current Condition

User Type

Color

Buy Time Price

Year of Manufacture

Add Your Price

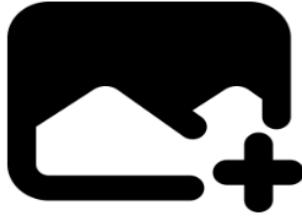
Specifications in detail

Submit

Add Product Photos

Upload and Submit 4 different Photos of your product

Category : Computer & Accessories
Product : Fridge
Front side



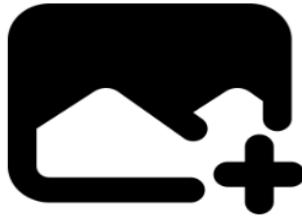
Choose File No file chosen Upload

Left side



Choose File No file chosen Upload

Right side



Choose File No file chosen Upload

back side



Choose File No file chosen Upload

[Submit](#)

Demand Product

Demand Product Details

Select Categories

Demand Product

Demand Brand/Material

Demand Color

Year of Manufacture

Submit

4.4 DATABASE DESIGN

An organized system with the capacity to retain information, a database enables users to quickly and effectively access stored data. Any database must be protected because the data is its main goal.

There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly as possible meet these criteria. This process, known as information level design, is carried out independently of all DBMSs.

The second phase consists of translating this information-level design into a design for the particular DBMS that will be used to implement the system in question. The properties of the particular DBMS that will be utilised are the focus of this process, referred to as Physical Level Design. Alongside the system design, a database design is created. The following two main goals are to be accomplished by the way the data is organised in the database.

- Integrity of the data
- Independency of the data

4.4.1 NoSQL database

Unlike relational databases, which store data in tables, NoSQL databases store data in documents. NoSQL database types include those that are exclusively document-based, key-value stores, wide-column, and graph-based. NoSQL databases are built from the ground up to suit a rising number of modern organisations to store and analyse huge volumes of data at scale.

NoSQL database technology stores data as JSON documents rather than the columns and rows that relational databases use to accomplish so. NoSQL, or "not only SQL," actually means "not SQL," not "no SQL." This suggests that data storage and retrieval in a NoSQL JSON database may essentially "use no SQL." Or, for the best of both worlds, you can mix JSON's flexibility with SQL's power. As a result, NoSQL databases are made to be flexible, scalable, and rapid to satisfy the data management requirements of modern businesses. Here, a NoSQL document database called Google Firestore is used.

The NoSQL data architecture of Cloud Firestore allows us to store data in documents with field mappings for values. A container used to store the papers is called a collection. These containers are how our data is organised, and they are how searches are made. Documents may

handle a wide range of data types, from simple characters and numbers to complex nested structures. We might also create sub-collections inside of documents and create a hierarchical data structure that grows along with our database. The Firestore data architecture is flexible enough to support any data structure that works best for our software. Also persuasive, useful, and adaptable is the query in Cloud Firestore.

DATABASE DESIGN

Collection Name: users

Field	Data type	Description
user_id	string	Unique identification
email_id	string	Must contain '@' and '.'
user_name	string	Rememberable name
password	string	Can be string, integer, and special characters
role	string	Role of user
personal	collection	Details of User
products	collection	Details of Products

Collection Name: personal

Field	Data type	Description
personal_id	string	Unique identification
first_name	string	The user's first name
last name	string	The user's last name
phone_no	number	Only 10 numbers aloud
house_name/no.	string	House name or number
street_name/no.	string	Street name or number
pincode	number	Pincode of the address
district	string	District of the customer
state	string	state of the customer
url	String	URL of profile picture

Collection Name: product

Field	Data type	Description
product_id	string	Unique identification
category	string	Category of the product
brand	string	Brand of the product
material	string	Material which is used in the product
model	string	Model of the product
scratch_and_dents	string	Scratch and dents if any
current_condition	string	The current condition of the product
user_type	string	Type of the user last used
color	string	Color of the product
yom	number	Year of manufacture of the product
buy_time_price	number	Buy time price of the product
your_price	number	Current expected price
date	timestamp	Ad posted date
weight	number	Weight of the product
specification	string	Product specifications
url1	string	URL of product's front-side photo
url2	string	URL of product's right-side photo
url3	string	URL of product's left-side photo
url4	string	URL of product's back-side photo
dem_product	string	Demanded product name
dem_category	string	Demanded product category
dem_brand_material	string	Demanded product brand or material
dem_color	string	Demanded product color
dem_yom	number	Demanded product year of manufacture
status	number	Status of the product

Collection Name: check

Field	Data type	Description
check_id	string	Unique identification
owner_products	reference	Owner's product
product_owner	reference	Product owner
rstatus	status	Requested status
user_ref	reference	Requested user
date	timestamp	The date which the request happens
user_process	collection	Details of product check

Collection Name: user_process

Field	Data type	Description
user_process_id	string	Unique identification
cu_coin	number	User product value
user_product	reference	User product details

Collection Name: order

Field	Data type	Description
order_id	string	Unique identification
owner_products	reference	Owner's product
product_owner	reference	Product owner
rstatus	string	Requested status
satisfy_status	number	Satisfaction status of the product
user_ref	reference	Requested user
date	timestamp	The date on which the request happens
user_process	collection	Details of product check

Collection Name: advertiser

Field	Data type	Description
adv_id	string	Unique identification
adv_email_id	string	Must contain '@' and '.'
user_name	string	Rememberable name
password	string	Can be string,integer, and special characters
role	string	Role of user
personal	collection	Details of user
advertisement	collection	Details of advertisement

Collection Name: ads

Field	Data type	Description
ads_id	string	Unique identification
ads_url	string	URL of advertisement
ads_plan	string	Plan chosen for advertise
ads_fare	number	Charge of cost for publishing ads
ads_date	timestamp	The date on which the request happens
status	string	Advertisement status

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the practice of carefully executing software in order to determine whether it behaves as intended. Verification and validation are phrases that are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to ensure that it complies with all relevant specifications. Software testing is merely one type of verification; other methods include reviews, analyses, inspections, and walkthroughs. Validation is the process of making sure what has been specified is what the user truly desired.

Static and dynamic analysis are additional practises that are frequently related to software testing. Software's source code is examined via static analysis, which collects metrics and looks for errors without actually running the code. In order to give data like execution traces, timing profiles, and test coverage information, dynamic analysis examines how software behaves while it is running.

A series of activities known as testing can be organised in advance and carried out in a methodical manner. Beginning with the module level, testing progresses to the integration of the full computer-based system. Testing is necessary for the system testing objectives to be successful, and there are many rules that can be used as testing objectives. These are:

- A programme is tested by being run with the goal of identifying any errors.
- A strong chance of discovering an undetected error characterises an excellent test scenario.
- Any test that finds an error that has not yet been found is successful.

Software bugs would be found if testing was successfully completed in accordance with the aforementioned objectives. Additionally, testing shows that the programme appears to function as intended and that the performance criteria appear to have been satisfied.

There are three methods for programme testing.

- For accuracy
- For effective implementation
- For the complexity of computing

A programme must perform exactly as it was intended to in order to pass a correctness test. This is considerably trickier than it might seem, especially with big programmes.

5.2 TEST PLAN

A test plan suggests a sequence of desirable steps to be taken in order to complete various testing methodologies. The Test Plan serves as a guide for the course of action. A computer programme, its documentation, and associated data structures are all made by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure that it fulfils the purpose for which it was intended. An independent test group (ITG) was created to address the inherent issues with allowing the builder to test what they had created. The exact goals of the test should be articulated in quantifiable terms. The test strategy should include information on the mean time to failure, cost to detect and correct problems, remaining defect density or frequency of occurrence, and test work hours per regression test.

The testing levels include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

Unit testing concentrates verification work on the software component or module, which is the smallest unit of software design. Important control pathways are evaluated to find faults within the module's border using the component level design description as a guide. Unit testing's disclosed breadth and test's relative complexity. Unit testing can be done in parallel for numerous components and is white-box focused. To verify that information enters and exits the software unit under test properly, the modular interface is put to the test. To make sure that data temporarily stored retains its integrity during each step of an algorithm's execution, the local data structure is inspected.

Before starting any other test, data flow tests across a module interface must be completed. All other tests are meaningless if data cannot enter and exit properly. The unit test must include the selective testing of execution pathways. Error circumstances must be expected in good design, and methods for handling errors must be put up so that they can be cleanly terminated or rerouted when they do occur. The penultimate phase in the unit testing process is boundary testing. At its limits, software frequently fails.

The Sell-Soft System carried out unit testing by treating each module as a distinct object and testing each one using a variety of test inputs. A few errors in the modules' internal logic were discovered and fixed. Each module is checked and run after it has been coded. To guarantee that every module functions properly and produces the desired outcome, all extraneous code was deleted.

5.2.1.1

Test Case 1

Project Name: Tit-Tat					
Signin Test Case					
Test Case ID: Test_1		Test Designed By: Vivin V Abraham			
Test Priority(Low/Medium/High): H igh		Test Designed Date: 20-07-2022			
Module Name: Customer		Test Executed By: Asst. Prof. Ajith G S			
Test Title : Signin the user with email and password		Test Execution Date: 20-07-2022			
Description: Test the Signin Page					
Pre-Condition : User has valid email id and password					
Step	Test Step	Test Data	Expecte dResult	Actual Result	Status(P ass/Fail)
1	Navigation to Signin Page		Signin Page should be displayed	Signin page displayed	Pass
2	Provide Valid Email Id	Email: anu123@gmail.com	User should be able login	User Signedin and went to home page	Pass
3	Provide Valid Password	Password: Anu@123456	and navigat e to home		

4	Click on Sign Up button		page		
5	Provide Invalid Email Id or password	Email Id: anu123@gmail.com Password: Anu@12345	User shouldn't login And redirect back to home page.	User Signed in and went to home page	Pass
7	Click on Sign In button				
Post-Condition: User logged in and successfully navigate to user home page					

Code:

```

from django.test import LiveServerTestCase
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

class Hosttest(LiveServerTestCase):
    def testform(self):
        selenium = webdriver.Chrome()
        #Choose your url to visit
        selenium.get('http://127.0.0.1:8000/account/signin')
        #find the elements you need to submit form
        player_name = selenium.find_element_by_name('email')
        player_pass = selenium.find_element_by_name('pass')

```

```

submit = selenium.find_element_by_name('button')

#populate the form with data

player_name.send_keys('anu123@gmail.com')

player_pass.send_keys('Anu@12345')

#submit form

submit.send_keys(Keys.RETURN)

#check result; page source looks at entire html document

assert 'Anu' in selenium.page_source

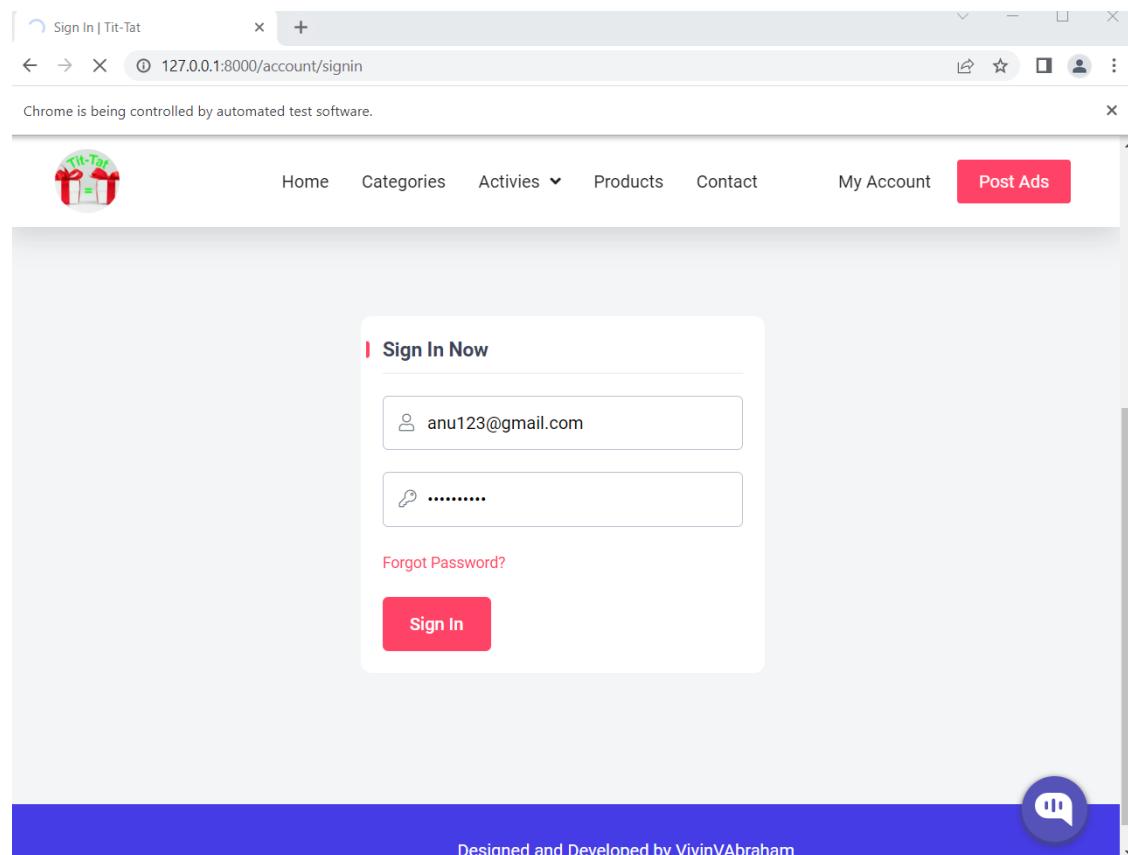
```

```

tests.py U x {} ServiceAccountKey.json categories.html M eval_history_bn.html M about.html M faq.html M form_template.html
tests.py > Hosttest > testform
7     selenium = webdriver.Chrome()
8     #Choose your url to visit
9     selenium.get('http://127.0.0.1:8000/account/signin')
10    #find the elements you need to submit form
11    player_name = selenium.find_element_by_name('email')
12    player_pass = selenium.find_element_by_name('pass')
13
14
15    submit = selenium.find_element_by_name('button')
16
17    #populate the form with data
18    player_name.send_keys('anu123@gmail.com')
19    player_pass.send_keys('Anu@123456')
20
21
22    #submit form
23    submit.send_keys(Keys.RETURN)
24
25    #check result; page source looks at entire html document
26    assert 'Anu' in selenium.page_source

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL
> TERMINAL
DevTools listening on ws://127.0.0.1:2551/devtools/browser/93ef9c49-5775-4139-aebe-3cee8195eccb
[25536:8800:0720/000418.691:ERROR:device_event_log_impl.cc(214)] [00:04:18.691] USB: usb_device_handle_win.cc:1048 Failed to read descriptor connection: A device attached to the system is not functioning. (0x1F)
[25536:8800:0720/000418.692:ERROR:device_event_log_impl.cc(214)] [00:04:18.693] USB: usb_device_handle_win.cc:1048 Failed to read descriptor connection: A device attached to the system is not functioning. (0x1F)
.
-----
Ran 1 test in 39.672s
OK
Destroying test database for alias 'default'...
hp@LAPTOP-U2SEQKP4 MINGW64 ~/project/Tit (master)
$ 

```



```

tests.py U {} ServiceAccountKey.json categories.html M eval_history_bn.html M about.html M faq.html M
	tests.py > Hosttest > testform
    7  selenium = webdriver.Chrome()
    8  #Choose your url to visit
    9  selenium.get('http://127.0.0.1:8000/account/signin')
   10  #find the elements you need to submit form
   11  player_name = selenium.find_element_by_name('email')
   12  player_pass = selenium.find_element_by_name('pass')
   13
   14
   15  submit = selenium.find_element_by_name('button')
   16
   17  #populate the form with data
   18  player_name.send_keys('anu123@gmail.com')
   19  player_pass.send_keys('Anu@12345')
   20
   21
   22  #submit form
   23  submit.send_keys(Keys.RETURN)
   24
   25  #check result; page source looks at entire html document
   26  assert 'Anu' in selenium.page_source

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL
TERMINAL
FAIL: testform (tests.Hosttest)
-----
Traceback (most recent call last):
  File "C:/Users/hp/project/Tit/tests.py", line 26, in testform
    assert 'Anu' in selenium.page_source
AssertionError

-----
Ran 1 test in 7.707s

FAILED (failures=1)
Destroying test database for alias 'default'...
hp@LAPTOP-U2SEQKP4 MINGW64 ~/project/Tit (master)
$ []

```

Test Case 2

Project Name: Tit-Tat					
Signup and Signin Test Case					
Test Case ID: Test_2		Test Designed By: Vivin V Abraham			
Test Priority(Low/Medium/High): High		Test Designed Date: 20-07-2022			
Module Name: Customer		Test Executed By : Vivin V Abraham			
Test Title : Signup and Signin		Test Execution Date: 20-07-2022			
Description: Sign up with username,email id,password and Signin with email and password					
Pre-Condition : The user has an Email ID,which is no one till used to Signup,A username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigation to Signup Page		Signup Page should be displayed	Signup page displayed	Pass
2	Provide valid Username	Username: Aravind	User should directed to Signin page	User directed to Signup page	Pass
3	Provide valid Email Id	Email: aravind2021b@mca.ajce.in			
4	Provide valid Password	Password: Aravind@123			
5	Click on Sign Up button				
6	Navigation to Signin Page		Signin Page should be displayed	Signin page displayed	Pass
7	Provide valid Email Id	Email: aravind2021b@mca.ajce.in			

			User should login redirect back to home page.	User Signed and went to home page	Pass
8	Provide valid Password	Password: Aravind@123			
9	Click on Sign In button				
10	Navigation to Signup Page		Signup Page should be displayed	Signup page displayed	Pass
11	Provide valid Username	Username: Aravind	User shouldn't direct to Signin page	User didn't directed to Signup page	Pass
12	Provide valid Email Id	Email: aravind2021b@mca.ajce.in			
13	Provide valid Password	Password: Aravind@123			
14	Click on Sign Up button				

Post-Condition: User Sign Up to the website with his own credentials but if the email is already in the database we cannot Sign Up to the account but we can Sign In with that credentials

Code:

```

from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import unittest

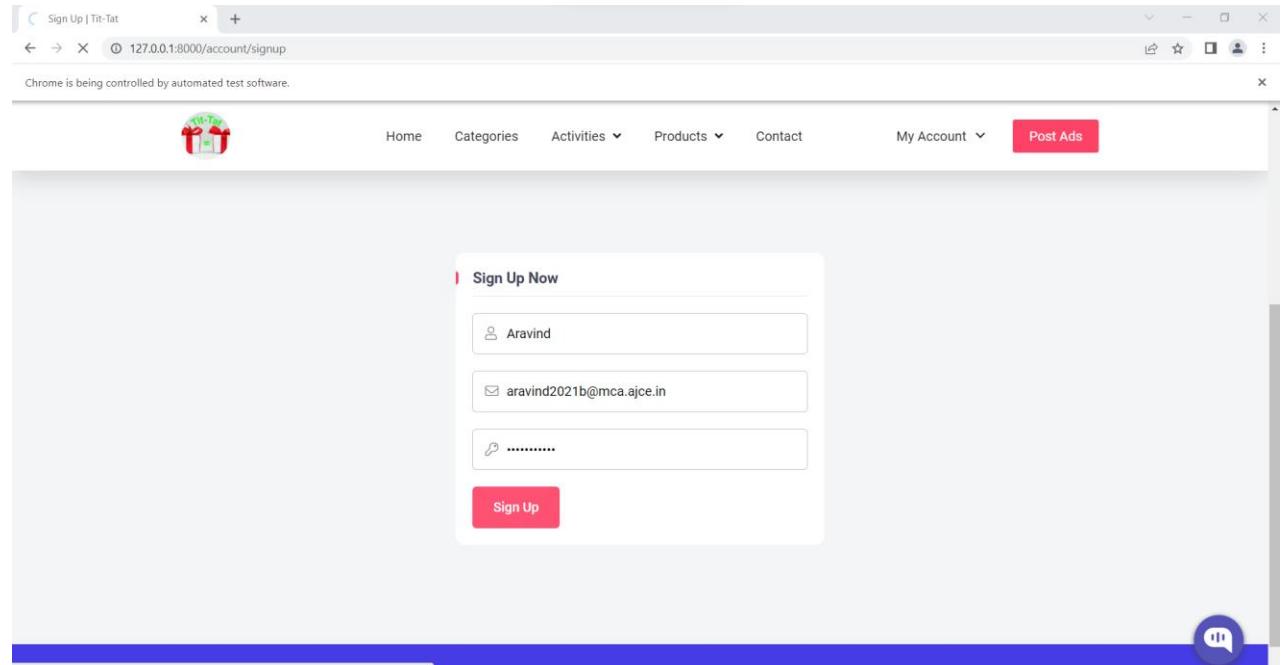
class TestExample(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        cls.selenium = webdriver.Chrome()

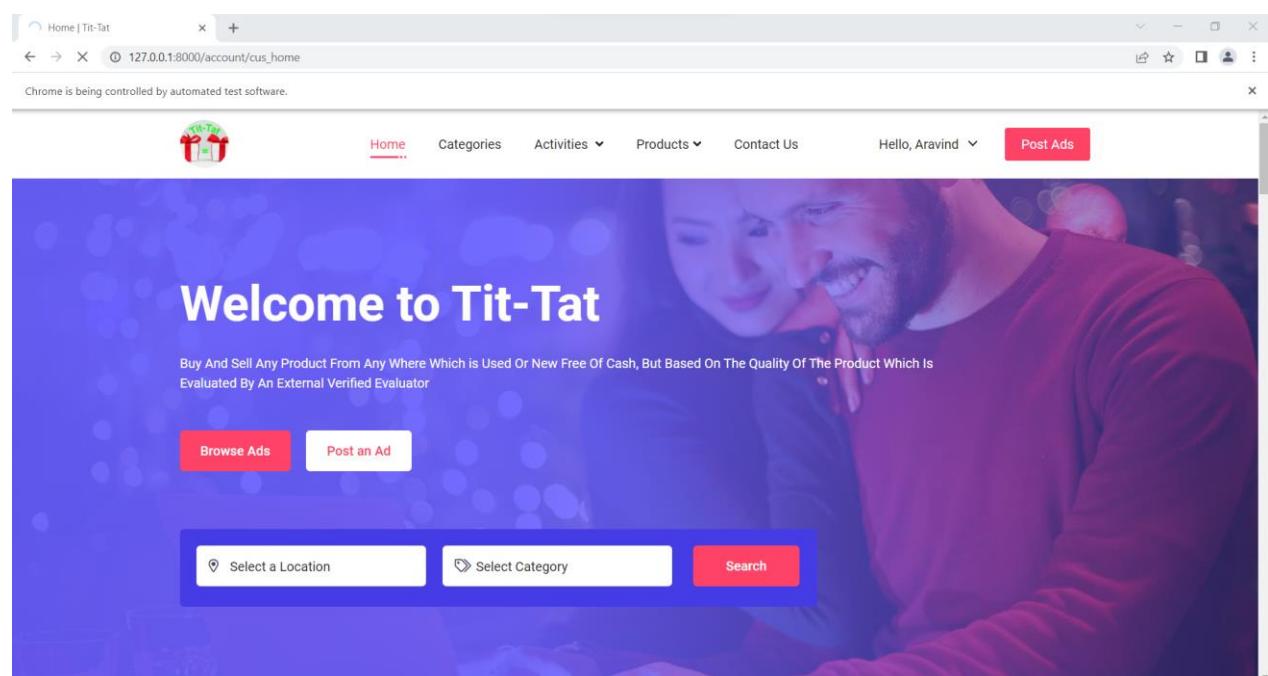
```

```
def test_testform2(self):  
    self.selenium.get('http://127.0.0.1:8000/account/signup')  
  
    player_name = self.selenium.find_element('name','name')  
  
    player_email = self.selenium.find_element('name','email')  
  
    player_password= self.selenium.find_element('name','pass')  
  
    submit= self.selenium.find_element('name','but1')  
  
    player_name.send_keys('Aravind')  
  
    player_email.send_keys('aravind2021b@mca.ajce.in')  
  
    player_password.send_keys('Aravind@123')  
  
    submit.send_keys(Keys.RETURN)  
  
    player_email = self.selenium.find_element('name','email')  
  
    player_password= self.selenium.find_element('name','pass')  
  
    submit= self.selenium.find_element('name','button')  
  
    player_email.send_keys('aravind2021b@mca.ajce.in')  
  
    player_password.send_keys('Aravind@123')  
  
    submit.send_keys(Keys.RETURN)  
  
    self.selenium.get('http://127.0.0.1:8000/account/cus_home')
```

First Sign Up



First Outcome



Second Output



The screenshot shows a browser window with the URL <http://127.0.0.1:8000/account/signup/>. The page displays a form with fields for name, email, and password. Below the form, there is a success message: "Success! Your account has been created successfully. You can now log in with the credentials you provided." At the bottom of the page, there is a link to "View my profile".

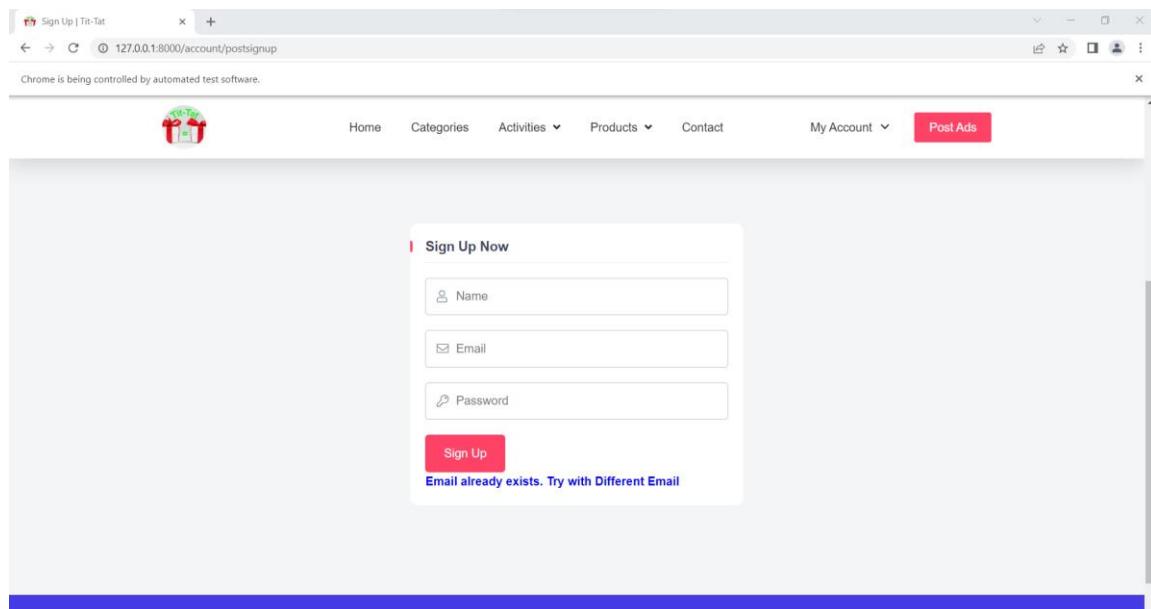
```
tests.py M categories.html eval_history_bn.html about.html 6 faq.html form_template.html categories_f.html product-d > 4 product Aa ab * No results ↑ ↓ ≡ ×
```

```
tests.py > TestExample > test_signup
def test_signup(self):
    self.selenium.get('http://127.0.0.1:8000/account/signup/')
    player_name = self.selenium.find_element('name','name')
    player_email = self.selenium.find_element('name','email')
    player_password= self.selenium.find_element('name','pass')
    submit = self.selenium.find_element('name','but1')
    player_name.send_keys('Aravind')
    player_email.send_keys('aravind2021b@mca.ajce.in')
    player_password.send_keys('Aravind@123')
    submit.send_keys(Keys.RETURN)

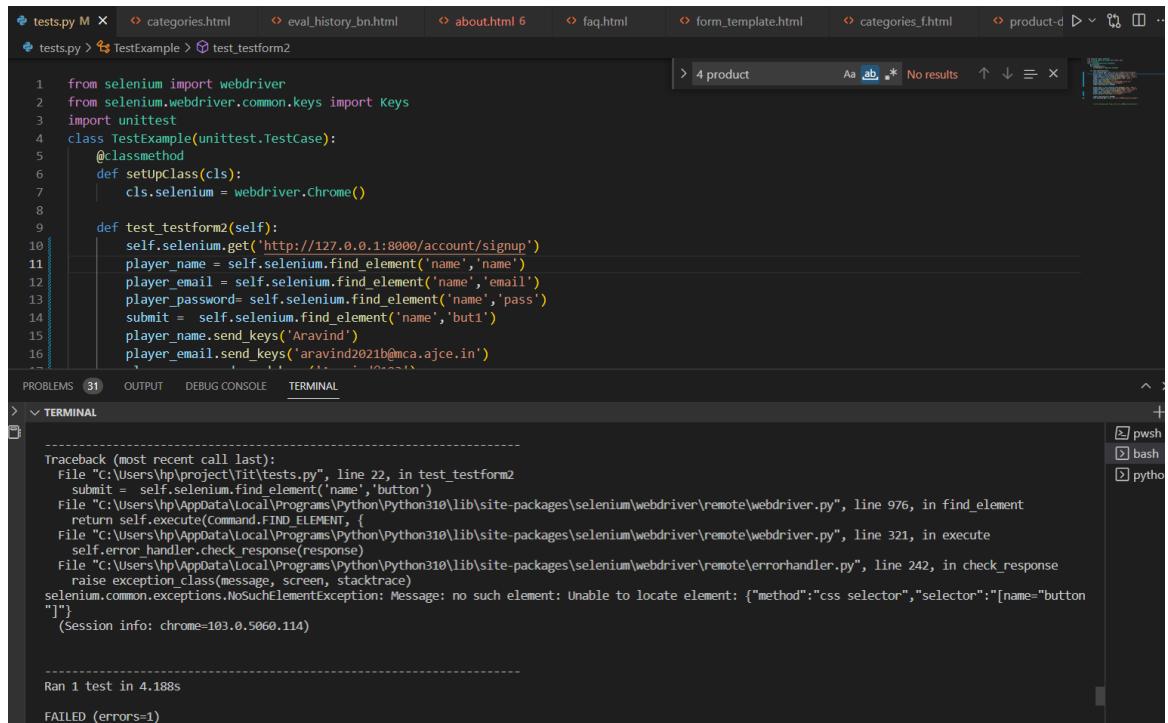
    player_email = self.selenium.find_element('name','email')
    player_password= self.selenium.find_element('name','pass')
    submit = self.selenium.find_element('name','button')
    player_email.send_keys('aravind2021b@mca.ajce.in')
    player_password.send_keys('Aravind@123')

    submit.send_keys(Keys.RETURN)
    self.selenium.get('http://127.0.0.1:8000/account/cus_home')
```

Second Sign Up & Outcome



Second Output



```

1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3  import unittest
4  class TestExample(unittest.TestCase):
5      @classmethod
6      def setUpClass(cls):
7          cls.selenium = webdriver.Chrome()
8
9      def test_form2(self):
10         self.selenium.get('http://127.0.0.1:8000/account/signup')
11         player_name = self.selenium.find_element('name','name')
12         player_email = self.selenium.find_element('name','email')
13         player_password= self.selenium.find_element('name','pass')
14         submit = self.selenium.find_element('name','but1')
15         player_name.send_keys('Aravind')
16         player_email.send_keys('aravind2021b@ma.ajce.in')

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

> v TERMINAL
-----
Traceback (most recent call last):
File "c:\Users\hp\project\Vit\tests.py", line 22, in test_form2
    submit = self.selenium.find_element('name','button')
File "C:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 976, in find_element
    return self.execute(Command.FIND_ELEMENT, {
File "C:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 321, in execute
    self.error_handler.check_response(response)
File "C:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-packages\selenium\webdriver\remote\errorhandler.py", line 242, in check_response
    raise exception_class(message, screen, stacktrace)
selenium.common.exceptions NoSuchElementException: Message: no such element: Unable to locate element: {"method":"css selector","selector": "[name="button"]"}
(Session info: chrome=103.0.5060.114)

-----
```

Ran 1 test in 4.188s

FAILED (errors=1)

5.2.2 Integration Testing

A methodical approach to building the program's structure and running tests at the same time to find interface issues is called integration testing. Building a programme structure that is predetermined by design using unit-tested components is the goal. An overall test is run on the entire software. Since the size of the overall software makes it difficult to isolate the causes, correction is challenging. New errors arise after these ones are fixed, and the process keeps repeating itself in an apparent never-ending cycle. All the modules were integrated after doing unit testing in the system to check for any interface discrepancies. Additionally, discrepancies in programme structure were eliminated, and a special programme structure developed.

5.2.3 Validation Testing or System Testing

The testing process ends with this phase. In this, the complete system, including all forms, code, modules, and class modules, was tested as a whole. Black box testing or system tests are common names for this type of testing.

The functional requirements of the software are the main focus of black box testing. In other words, Black Box testing enables the software engineer to develop sets of input circumstances that will fully exercise all functional requirements for a programme.

The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors

5.2.1 Output Testing or User Acceptance Testing

User acceptance testing is done on the system under consideration; in this case, it must meet the needs of the company. While it is being developed, the software should stay in touch with the user and perspective system to make any necessary modifications. Regarding the following points, this is accomplished:

- Design of Input Screens
- Design of output Screens

Testing is conducted as stated above using various test data types. In order to test a system effectively, test data must be prepared. The system under investigation is then put to the test utilising the test data that was prepared earlier. Errors in the test data are found again during system testing, and they are fixed using the methods described above. The fixes are also recorded for use in the future.

.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The step of the project known as implementation is where the conceptual design is translated into a functional system. Gaining the users' trust that the new system will operate as intended, be reliable, and effective can be viewed as the most important stage in creating a successful new system. User education and documentation are its main concerns. Normally, conversion happens either at the same time that the user is being trained or after. Implementing a new system design simply refers to putting it into action, which is the process of turning a newly revised system design into an operational one.

At this point, the user department is responsible for the majority of the workload, the most disruption, and the most influence on the current system. It can lead to chaos and confusion if the implementation is not adequately planned or managed.

The process of converting from the old system to the new system is referred to as implementation. The new system might be entirely different, taking the place of an existing human or automated system, or it might be an adjustment to an existing system. For a system to be dependable and satisfy organisational needs, proper implementation is crucial. System implementation is the action of putting a developed system into practical use. This comprises all of the tasks necessary to switch from the old to the new system. The system can only be put into use after extensive testing and if it is determined to be operating in accordance with the requirements. The system staff examines the system's viability.

The following tasks are included in the implementation state:

- Planning with care.
- Investigation of the system and restrictions
- The creation of transitional methods.

6.2 IMPLEMENTATION METHODS

The ultimate installation of the software package in its intended environment, as well as the system's functionality and fulfilment of its intended applications, are all considered to be aspects of software implementation. In many businesses, the software development project is commissioned by someone who will not be using it. People have early reservations about the programme, but we need to prevent resistance from growing because it's important to make sure that:

- The active user needs to understand the advantages of utilising the new system.
- Their faith in the software is increased.
- The user is provided with the appropriate training so that he feels confident using the application.

The user must be aware that the server software must be running on the server in order to view the results before proceeding to view the system. The real process won't happen if the server object isn't operating on the server and is up and running.

6.2.1 Training of Users

The purpose of user training is to get the user ready for system testing and conversion. It is crucial for the participants to have faith in their roles in the new system in order to achieve the goal and benefits anticipated from the computer-based system. The necessity for training is more critical as systems get more sophisticated. By receiving user training, a user learns how to enter data, handle error warnings, query the database, call up a procedure to generate reports, and execute other important tasks.

6.2.2 Application Software Training

The user will need to be instructed on the new application software after receiving the appropriate basic training on computer awareness. This will explain the fundamental principles of how to use the new system, such as how the screens flow, how they are designed, what kinds of errors can occur while entering data, how to fix those errors, and how each entry is validated. The information needed by the particular user or group to utilise the system or a specific component of the system should then be covered as the program's training on the application is being given. There could be variations in this training depending on the user groups and hierarchical levels.

6.2.3 System Maintenance

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just "Finding Mistakes."

6.3 Hosting

The system is hosted in PythonAnywhere. PythonAnywhere provide free and easy hosting of python projects upto 500 MB in project size. The database is Google Firestore which is already hosted by Google. Pythonanywhere provide a free domain with our registered username.

Steps to be followed:

Step 1:

Sign up to pythonanywhere.com

Step 2:

As our code is ready on GitHub, we will clone it using Bash Console. Create a Bash Console by clicking on the Console Tab. You will see the terminal interface where you need to type git clone command.

Step 3:

Create VirtualEnv and Install Django and Dependencie, In your Bash console, create a virtualenv, named after your project name, and choose the version of Python you want to use

mkvirtualenv --python=/usr/bin/python3.9 mysite-virtualenv

Step 4:

Once the Virtual Env is activated. Install Django inside it.

Step 5:

Create a Web Application with Manual Config

Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » [Django](#)
- » [web2py](#)
- » [Flask](#)
- » [Bottle](#)
- » [Manual configuration \(including virtualenvs\)](#) Click Here**

Step 6:

Once your web app is created, scroll down and you need to add some paths.

Make sure you add folders' names accordingly to your project names.

- * Code – /home/myusername/mysite/
- * Virtualenv – /home/username/.virtualenvs./mysite-virtualenv

Step 7:

Edit WSGI File to Point our Django Project to Server.

Code:

What your site is running.

Source code:	/home/danielthewilliam/Django-Quiz-Application	Go
Working directory:	/home/danielthewilliam/	Go
WSGI configuration file:	/var/www/danielthewilliam_pythonanywhere_com_wsgi.py	
Python version:	3.7	Click On it

Click on the WSGI configuration file link. It will take us to the Hosted Django Project WSGI File and add the following code.

```
import os
import sys
# assuming your Django settings file is at
# '/home/myusername/mysite/mysite/settings.py'
path = '/home/myusername/mysite'
if path not in sys.path:
    sys.path.insert(0, path)
os.environ['DJANGO_SETTINGS_MODULE'] = 'mysite.settings'
from django.core.wsgi import get_wsgi_application
application = get_wsgi_application()
```

That's it. Now the final step is to go to Web Tab and reload the web app and click on the link of your web app

The url of the hosted website is: <https://tittat.pythonanywhere.com>.

The screenshot shows the PythonAnywhere web interface for managing a site named 'tittat.pythonanywhere.com'. At the top, there's a warning message: 'Warning! You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#).'. Below this, the configuration for the site is shown, including the WSGI configuration code. There are several buttons and links: 'Add a new web app', 'Reload tittat.pythonanywhere.com', 'Run until 3 months from today', and 'Paying users' sites stay up forever without any need to log in to keep them running.' At the bottom, there's a section for 'Traffic' and 'How busy is your site?'. The interface has a clean, modern design with a blue header and white background.

The screenshot shows the PythonAnywhere dashboard. At the top, there's a message: "Warning: You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#)." Below this, there's a logo for Anaconda and a note: "PythonAnywhere is now part of Anaconda! We will continue to make and run the service in the way that you have come to expect. Anaconda's investment in PythonAnywhere will allow us to expand and improve it to better serve you, our users. To learn more about what's going on take a look at the [press release](#) and our [F.A.Q.](#)".

Dashboard

Welcome, [tittat](#)

CPU Usage: 0% used – 0.00s of 100s. Resets in 12 hours, 44 minutes [More Info](#)

File storage: 65% full – 331.3 MB of your 512.0 MB quota [More Info](#) [Upgrade Account](#)

Recent Consoles

- Bash console 24978734
- New console: [\\$ Bash](#) [>>> Python](#) [More...](#)

Recent Files

- /home/tittat/tit/admins/views.py
- /home/tittat/tit/account/views.py
- /home/tittat/tit/Tit/views.py

Recent Notebooks

Your account does not support Jupyter Notebooks. Upgrade your account to get access!

All Web apps

tittat.pythonanywhere.com [Open Web tab](#)

The screenshot shows the PythonAnywhere files section. At the top, there's a message: "Warning: You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#)." Below this, there's a note: "Open Bash console here 65% full – 331.3 MB of your 512.0 MB quota [More Info](#)".

Directories

- .cache/
- .local/
- .virtualenvs/
- tit/

Files

- bashrc 2022-07-20 20:11 559 bytes
- gitconfig 2022-07-20 20:11 256 bytes
- profile 2022-07-20 20:11 79 bytes
- pythonstartup.py 2022-07-20 20:11 77 bytes
- vimrc 2022-07-20 20:11 4.6 KB
- README.txt 2022-07-20 20:11 232 bytes

[Upload a file](#) 100MB maximum size

Hosted Website

The screenshot shows the homepage of the Tit-Tat website. The header includes a logo, navigation links for Home, Categories, Activities, Products, Contact Us, My Account, and Post Ads. The main banner features a smiling man and the text "Welcome to Tit-Tat". Below the banner, there's a subtext: "Buy And Sell Any Product From Any Where Which is Used Or New Free Of Cash, But Based On The Quality Of The Product Which Is Evaluated By An External Verified Evaluator". There are two buttons: "Browse Ads" and "Post an Ad". At the bottom, there are search fields for "Select a Location", "Select Category", and a "Search" button, along with a "Feedback" icon.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no way to use that website if there is no money in the hand of user if he wishes to buy a product when he has a good used useful product in his hand. The proposed system introduces facility for customer to brought and sell product with no money in their hand. Also verified users and products only in the website which is evaluated by the evaluator.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that no money for buy and sell products
- Customers can able to do advanced search options
- Every product is verified by the evaluator thus actual value of the product preserved
- Customers can update and add details of their profile and product details
- Data security can be enhanced.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- Pankaj Jalote, “*Software engineering: a precise approach*”, 2006.
- James Lee and Brent Ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- www.agilemodeling.com/artifacts/useCaseDiagram.html
- www.geeksforgeeks.com
- www.djangoproject.com
- www.youtube.com
- Draw.io, figma.com
- Google firebase documentation

CHAPTER 9

APPENDIX

9.1 Sample Code

views.py (signin)

```

def signin(request):
    return render(request, 'signin.html')
def postsign(request):
    email = request.POST.get('email')
    passw = request.POST.get('pass')
    try:
        user = authe.sign_in_with_email_and_password(email, passw)
    except:
        message = "invalid credentials"
        return render(request, "signin.html", { "messg": message })
    session_id = user['idToken']
    request.session['uid'] = str(session_id)
    idtoken = request.session['uid']
    a = authe.get_account_info(idtoken)
    a = a['users']
    a = a[0]
    a = a['localId']
    request.session['local'] = a
    datas = db.collection('user').document(a).get().to_dict()
    request.session['name']= datas["name"]
    per_data=db.collection('user').document(a).collection('personal').get()
    for per in per_data:
        perd=per.to_dict()
        request.session['url']=perd["url"]

    print(request.session['name'])
    if datas["role"] == "Evaluator":
        return redirect(evaluator_dash)
    else:
        return redirect(hello)
@cache_control(no_cache=True, must_revalidate=True, no_store=True)
def hello(request):
    if request.session.is_empty():
        return redirect(signin)
    return redirect(cus_home)

```

urls.py (signin)

```

from django.urls import path
from . import views
urlpatterns=[
    path('signin', views.signin),
    path('postsign',views.postsign),
]

```

signin.html

```
{% load static %}

<!doctype html>
<html class="no-js" lang="en">

    <head>
        <meta charset="utf-8">

        <title>Sign In | Tit-Tat</title>
        <meta name="description" content="">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link rel="shortcut icon" href="{% static 'images/logo.png' %}" type="image/png">

        <link rel="stylesheet" href="{% static 'css/magnific-popup.css' %}">

        <link rel="stylesheet" href="{% static 'css/nice-select.css' %}">

        <link rel="stylesheet" href="{% static 'css/slick.css' %}">

        <link rel="stylesheet" href="{% static 'css/ion.rangeSlider.min.css' %}">

        <link rel="stylesheet" href="{% static 'css/fontawesome.min.css' %}">

        <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">

        <link rel="stylesheet" href="{% static 'css/default.css' %}">

        <link rel="stylesheet" href="{% static 'css/style.css' %}">
    </head>

    <body class="gray-bg">
        <script src="{% static 'script.js' %}" type="text/javascript">
        </script>

        <div class="preloader">
            <div class="loader">
                <div class="ytp-spinner">
                    <div class="ytp-spinner-container">
                        <div class="ytp-spinner-rotator">
                            <div class="ytp-spinner-left">
                                <div class="ytp-spinner-circle"></div>
                            </div>
                            <div class="ytp-spinner-right">
                                <div class="ytp-spinner-circle"></div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </body>
```

```

        </div>
    </div>
</div>

<header class="header_area">
    <div class="header_navbar">
        <div class="container">
            <nav class="navbar navbar-expand-lg">
                <a class="navbar-brand" href="/">
                    
                </a>
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                    <span class="toggler-icon"></span>
                    <span class="toggler-icon"></span>
                    <span class="toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse sub-menu-bar" id="navbarSupportedContent">
                    <ul class="navbar-nav m-auto">
                        <li>
                            <a href="/">Home <span class="line"></span></a>
                        </li>
                        <li>
                            <a href="../categories">Categories <span class="line"></span></a>
                        </li>
                        <li>
                            <a href="#">Activities <i class="fa fa-angle-down"></i> <span class="line"></span></a>
                            <ul class="sub-menu">
                                <li><a href="../about">About</a></li>
                                <li><a href="../faq">FAQ</a></li>
                            </ul>
                        </li>
                        <li><a href="#">Products <span class="line"></span></a>
                            </li>
                            <li><a href="../contact">Contact <span class="line"></span></a></li>
                        </ul>
                    </li>
                    <div class="dropdown">
                        <ul>
                            <li>
                                <div class="dropdown">
                                    <a class="active" href="#" class="dropdown-toggle" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">My Account</a>
                                    <div class="dropdown-menu" aria-labelledby="dropdownMenuLink">

```

```

        <ul>
            <li><a class="active" href="signin"><i class="fas fa-sign-in-alt"></i> Sign In</a></li>
                <li><a href="signup"><i class="fa fa-user"></i> Sign Up</a></li>

            </ul>
        </div>
    </div>
    </li>
    <li><a class="sign-up" href="signin">Post Ads</a></li>
</ul>
</div>
</nav>
</div>
</div>
<div class="page_banner bg_cover" style="background-image: url({% static 'images/page-banner.jpg' % })">
    <div class="container">
        <div class="page_banner_content">
            <h3 class="title">Sign In</h3>
            <ul class="breadcrumb">
                <li><a href="/">Home</a></li>
                <li>Sign In</li>
            </ul>
        </div>
    </div>
</div>
</div>
</header>

<section class="sign_in_area pt-120 pb-120">
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-lg-5 col-md-7 col-sm-9">
                <div class="sign_in_form">
                    <div class="sign_title">
                        <h5 class="title">Sign In Now</h5>
                    </div>
                    <form action="postsign" method="post" name="myform" onsubmit="return validate()">
                        {% csrf_token %}
                        <div class="sign_form_wrapper">
                            <div class="single_form">
                                <input type="email" placeholder="Email ID" name="email" required>
                                    <i class="fal fa-user"></i>
                                </div>
                            <div class="single_form">
                                <input type="password" placeholder="Password" name="pass" required>
                                    <i class="fal fa-key"></i>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</section>

```



```
<script>
    function validate() {
        if (document.myform.email.value.trim() == "") {
            alert("Please put your email");
            document.myform.email.focus();
            return false;
        }
        var x = document.myform.email.value;
        var atposition = x.indexOf("@");
        var dotposition = x.lastIndexOf(".");
        if (atposition < 1 || dotposition < atposition + 2 || dotposition + 2 >= x.length) {
            alert("Please enter a valid e-mail address");
            return false;
        }
        if (document.myform.pass.value.trim() == "") {
            alert("Please put your password");
            document.myform.name.focus();
            return false;
        } else {
            return true;
        }
    }
</script>
```

```
<a href="#" class="back-to-top"><i class="fa fa-angle-up"></i></a>

<script src="{ % static 'js/vendor/jquery-1.12.4.min.js' % }"></script>
<script src="{ % static 'js/vendor/modernizr-3.7.1.min.js' % }"></script>

<script src="{ % static 'js/popper.min.js' % }"></script>
<script src="{ % static 'js/bootstrap.min.js' % }"></script>

<script src="{ % static 'js/slick.min.js' % }"></script>

<script src="{ % static 'js/jquery.magnific-popup.min.js' % }"></script>

<script src="{ % static 'js/jquery.nice-select.min.js' % }"></script>

<script src="{ % static 'js/waypoints.min.js' % }"></script>
<script src="{ % static 'js/jquery.counterup.min.js' % }"></script>

<script src="{ % static 'js/ion.rangeSlider.min.js' % }"></script>

<script src="{ % static 'js/ajax-contact.js' % }"></script>

<script src="{ % static 'js/main.js' % }"></script>
</body>
```

```
</html>
```

views.py (search)

```
def search(request):
    a=request.session['local']
    district=request.POST.get('district')
    category=request.POST.get('category')
    prolist=[]
    if district != 'none' and category != 'none' :
        user=db.collection('user').get()
        for u in user:
            personal=db.collection('user').document(u.id).collection('personal').where('district','==',district).get()
            for person in personal:
                parent=person.reference.parent.parent.id
                if parent != a:
                    persondict = person.to_dict()
                    print("gefdf")
                    print(persondict)

            prod=db.collection('user').document(parent).collection('products').where('category','==',category).get()
            for pp in prod:
                product=pp.to_dict()
                product["userid"]=parent
                product["productid"]=pp.id
                product["district"]=persondict["district"]
                product["state"]=persondict["state"]
                res = product
                prolist.append(res)
            print(prolist)
        return render(request,'all_products.html',
        {"prd":prolist,"district":district,"category":category})
```

urls.py (search)

```
from django.urls import path
from . import views
urlpatterns=[
    path('search',views.search),
]
```

all_products.html

```
{% load static %}  
<!doctype html>  
<html class="no-js" lang="en">  
<head>  
    <meta charset="utf-8">  
    <title>Products | Tit-Tat</title>  
    <meta name="description" content="">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
  
    <link rel="shortcut icon" href="{% static 'images/favicon.png' %}" type="image/png">  
  
    <link rel="stylesheet" href="{% static 'css/magnific-popup.css' %}">  
  
    <link rel="stylesheet" href="{% static 'css/nice-select.css' %}">  
  
    <link rel="stylesheet" href="{% static 'css/slick.css' %}">  
  
    <link rel="stylesheet" href="{% static 'css/ion.rangeSlider.min.css' %}">  
  
    <link rel="stylesheet" href="{% static 'css/fontawesome.min.css' %}">  
  
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">  
  
    <link rel="stylesheet" href="{% static 'css/default.css' %}">  
  
    <link rel="stylesheet" href="{% static 'css/style.css' %}">  
</head>  
  
<body class="gray-bg">  
  
    <div class="preloader">  
        <div class="loader">  
            <div class="ytp-spinner">  
                <div class="ytp-spinner-container">  
                    <div class="ytp-spinner-rotator">  
                        <div class="ytp-spinner-left">  
                            <div class="ytp-spinner-circle"></div>  
                        </div>  
                        <div class="ytp-spinner-right">  
                            <div class="ytp-spinner-circle"></div>  
                        </div>  
                    </div>  
                </div>  
            </div>  
        </div>  
    </div>  
    <header class="header_area">  
        <div class="header_navbar">  
            <div class="container">  
                <nav class="navbar navbar-expand-lg">  
  
                    <a class="navbar-brand" href="cus_home">  
                          
                    </a>
```

```

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span class="toggler-icon"></span>
            <span class="toggler-icon"></span>
            <span class="toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse sub-menu-bar" id="navbarSupportedContent">
            <ul class="navbar-nav m-auto">
                { % if nam %}
                <li>
                    <a href="/account/cus_home">Home <span class="line"></span></a>
                </li>
                { % else %}
                <li>
                    <a href="cus_home">Home <span class="line"></span></a>
                </li>
                { % endif %}
                <li>
                    <a href="/account/categories">Categories <span class="line"></span></a>
                </li>
                <li>
                    <a class="active" href="#">Activities <i class="fa fa-angle-down"></i> <span class="line"></span></a>
                    <ul class="sub-menu">
                        <li><a href="/account/about">Content</a></li>

                        <li><a href="/account/faq">FAQ</a></li>

                    </ul>
                </li>
                <li><a href="#">Products <i class="fa fa-angle-down"></i> <span class="line"></span></a>
                    <ul class="sub-menu">
                        <li><a href="/account/requested_products">Requested Products</a></li>
                        <li><a href="/account/product_history">Ads History</a></li>
                    </ul>
                </li>
                <li><a href="/account/contact">Contact <span class="line"></span></a></li>
            </ul>
        </div>
        <div class="navbar_btn">
            <ul>
                <li>
                    <div class="dropdown">
                        { % if nam %}
                            <a href="#" class="dropdown-toggle" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Hello, { { nam } }</a>
                            <div class="dropdown-menu" aria-labelledby="dropdownMenuLink">
                                <ul>
                                    <li><a href="/account/profilesettings"><i class="fas fa-user-alt"></i>Profile
                                        </a></li>
                                    <li><a href="/account/myads"><i class="fal fa-layer-group"></i> My
                                </ul>
                            </div>
                        { % else %}
                            <a href="#" class="dropdown-toggle" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Logout</a>
                            <div class="dropdown-menu" aria-labelledby="dropdownMenuLink">
                                <ul>
                                    <li><a href="/account/logout">Logout</a></li>
                                </ul>
                            </div>
                        { % endif %}
                    </div>
                </li>
            </ul>
        </div>
    </div>

```

```

Ads</a></li>
    <li><a href="{% url 'log' %}"><i class="fal fa-sign-out"></i> Sign
Out</a></li>
    </ul>
</div>
{%
else %
    <a href="#" class="dropdown-toggle" id="dropdownMenuLink" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">My Account</a>

<div class="dropdown-menu" aria-labelledby="dropdownMenuLink">
    <ul>
        <li><a href="/account/signin"><i class="fas fa-sign-in-alt"></i> Sign
In</a></li>
        <li><a href="/account/signup"><i class="fa fa-user"></i> Sign
Up</a></li>

    </ul>
</div>
{%
endif %

</div>
{%
if nam %
    <li><a class="sign-up" href="/account/postadd">Post Ads</a></li>
{%
else %
    <li><a class="sign-up" href="account/signin">Post Ads</a></li>
{%
endif %

</ul>
</div>
</nav>
</div>
</div>
<div class="page_banner bg_cover" style="background-image: url({% static 'images/page-
banner.jpg' % })">
    <div class="container">
        {%
if district %
            <div class="page_banner_content">
                <h3 class="title"> Products in {{district}} with {{category}}</h3>
                <ul class="breadcrumb">
                    {%
if district %
                        <li><a href="cus_home">Home</a></li>
{%
else %
                        <li><a href="cus_home">Home</a></li>
{%
endif %
                        <li>Products in {{district}} in {{category}}</li>
                    </ul>
                </div>
{%
else %
            <div class="page_banner_content">
                <h3 class="title">{{category}}</h3>
                <ul class="breadcrumb">
                    {%
if district %
                        <li><a href="cus_home">Home {{district}}</a></li>
{%
else %
                        <li><a href="cus_home">Home</a></li>
{%
endif %
                        <li>{{category}}</li>
                    </ul>
                </div>
        </div>
    </div>
{%
endif %
}

```

```

        </ul>
    </div>
    { % endif %
    </div>
</div>
</header>

<section class="product_page pt-70 pb-120">
    <div class="container">
        <div class="row">
            <div class="col-lg-3">
                <div class="product_sidebar pt-20">
                    <div class="sidebar_categories mt-30">
                        <div class="sidebar_title">
                            <h5 class="title">Categories</h5>
                        </div>
                        <div class="sidebar_categories_content">
                            <div class="accordion" id="accordionExample">
                                <div class="card">
                                    <div class="card-header" id="headingSeven">
                                        <a href="/account/all_product" style="font-size:16px;"><i
style="color:rgb(255, 0, 0);></i>&ampnbsp&ampnbsp&ampnbspAll Products</a>
                                    </div>
                                </div>
                            </div>
                        <div class="card">
                            <div class="card-header" id="headingOne">
                                <a href="/account/all_products/Computer & Accessories" style="font-
size:16px;"> <i style="color:rgb(255, 0, 0);></i> Computer &
Accessories</a>
                            </div>
                            </div>
                            <div class="card">
                                <div class="card-header" id="headingTwo">
                                    <a href="/account/all_products/Furnitures" style="font-size:16px;"> <i
style="color:rgb(255, 0, 0);></i>&ampnbsp Furnitures</a>
                                </div>
                            </div>
                            <div class="card">
                                <div class="card-header" id="headingThree">
                                    <a href="/account/all_products/Phone & Accesories" style="font-
size:16px;"> <i style="color:rgb(255, 0, 0);></i>&ampnbsp&ampnbsp&ampnbspPhone & Accesories</a>
                                </div>
                            </div>
                            <div class="card">
                                <div class="card-header" id="headingFour">
                                    <a href="/account/all_products/Camera & Accesories" style="font-
size:16px;"> <i style="color:rgb(255, 0, 0);></i>&ampnbsp&ampnbspCamera
& Accesories</a>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>

```

```

        </div>

        </div>
        <div class="card">
            <div class="card-header" id="headingFour">
                <a href="/account/all_products/Vehicles" style="font-size:16px;"><i
style="color:rgb(255, 0, 0);><i>&ampnbsp&ampnbspVehicles</a>
            </div>

            </div>
            <div class="card">
                <div class="card-header" id="headingFour">
                    <a href="/account/all_products/Electronics" style="font-size:16px;"><i
style="color:rgb(255, 0, 0);><i>&ampnbsp Electronics</a>
                </div>

            </div>

            </div>
            <div class="card">
                <div class="card-header" id="headingSix">
                    <a href="/account/all_products/Sports & Gaming" style="font-
size:16px;"><i style="color:rgb(255, 0, 0);><i>&ampnbspSports &
Gaming </a>
                </div>

                </div>
                <div class="card">
                    <div class="card-header" id="headingSeven">
                        <a href="/account/all_products/Rare Products" style="font-
size:16px;"><i style="color:rgb(255, 0, 0);><i>&ampnbsp&ampnbsp&ampnbspRare
Products</a>
                    </div>
                    </div>
                    </div>
                    </div>
                    </div>
                    </div>
                    </div>
                    <div class="col-lg-9">
                        <div class="product_topbar d-md-flex align-items-center justify-content-between pb-
30">
                            </div>
                            <div class="tab-content" id="myTabContent">
                                <div class="tab-pane fade show active" id="grid" role="tabpanel" aria-
labelledby="grid-tab">
                                    <div class="product_grid">
                                        <div class="row">
                                            { %for p in prd % } { % if request.session.local != p.userid % }
                                            <div class="col-xl-4 col-sm-6">
                                                <div class="single_ads_card mt-30">
                                                    <div class="ads_card_image">
                                                        
                                                    </div>
                                                    <div class="ads_card_content">
                                                        <div class="meta d-flex justify-content-between">

```



```
<a href="#" class="back-to-top"><i class="fa fa-angle-up"></i></a>
<script src="{% static 'js/vendor/jquery-1.12.4.min.js' %}"></script>
<script src="{% static 'js/vendor/modernizr-3.7.1.min.js' %}"></script>

<script src="{% static 'js/popper.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>

<script src="{% static 'js/slick.min.js' %}"></script>

<script src="{% static 'js/jquery.magnific-popup.min.js' %}"></script>

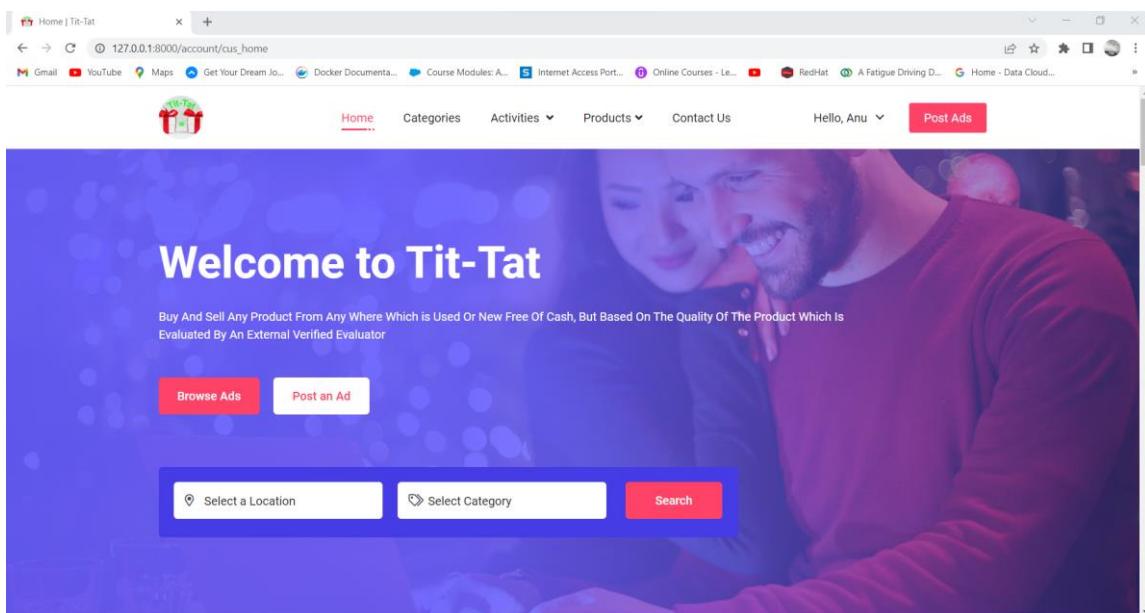
<script src="{% static 'js/jquery.nice-select.min.js' %}"></script>

<script src="{% static 'js/waypoints.min.js' %}"></script>
<script src="{% static 'js/jquery.counterup.min.js' %}"></script>
<script src="{% static 'js/ion.rangeSlider.min.js' %}"></script>
<script src="{% static 'js/ajax-contact.js' %}"></script>

<script src="{% static 'js/main.js' %}"></script>
</body>
</html>
```

9.2 Screen Shots

Home page (After login)



Coins Check

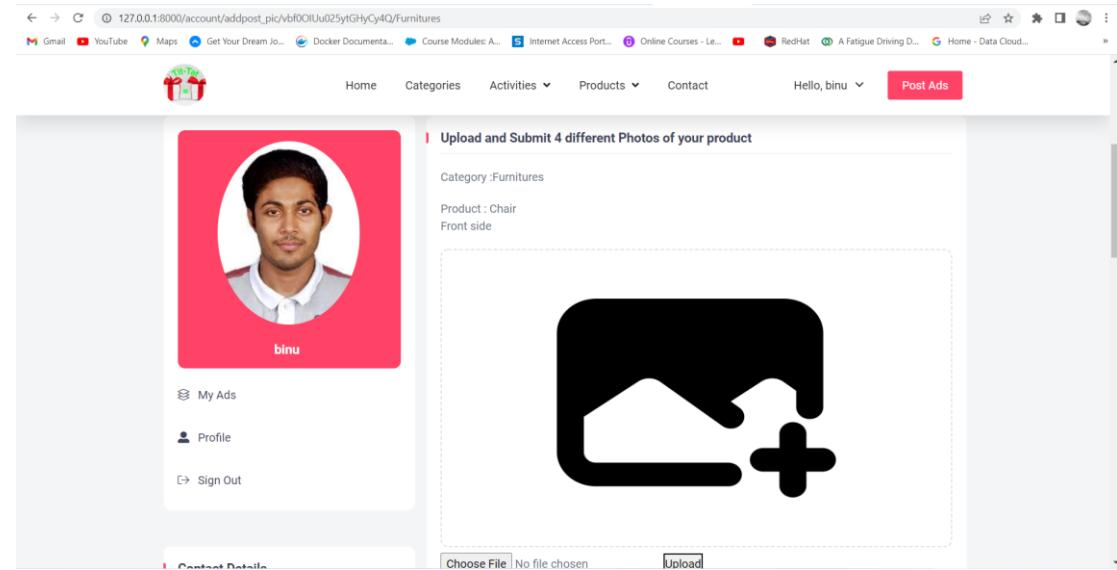
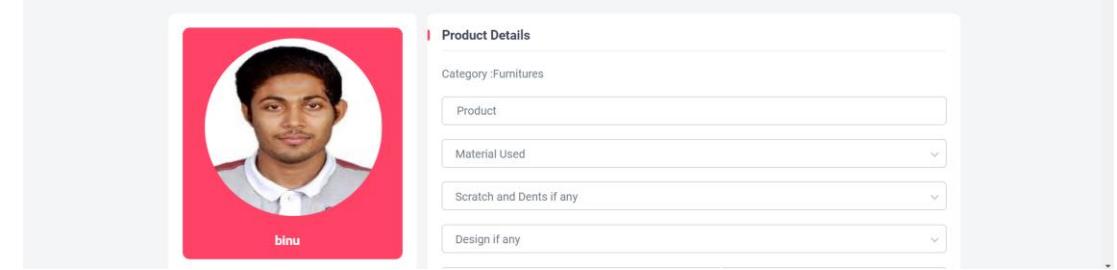
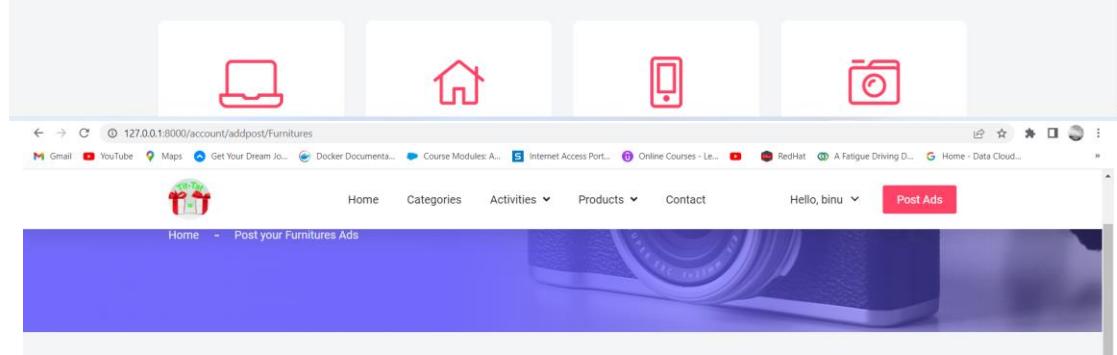
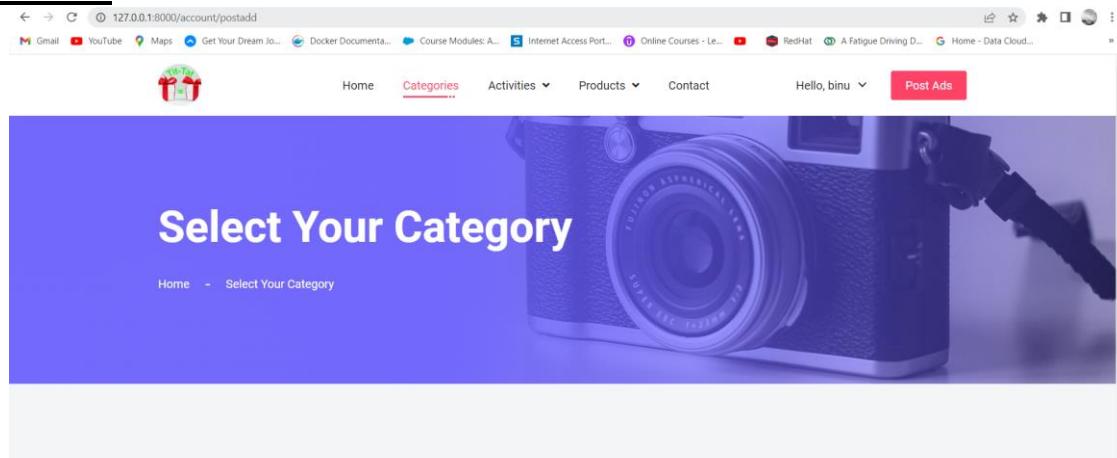
The screenshot shows a product listing for a Samsung 2018 Smart Phone. The listing includes a photo of the phone, the title "Samsung 2018 Smart Phone", the ad ID "9YGP002TwaBAwMvzx4xX", and the price "10000". To the right, there is a sidebar for the owner, Binu Augustine, showing their profile picture, address "Bethel House Uchakkada Ponkunnam Trivandrum", and phone number "8512351423". There are two buttons: "Click to See Number" and "Check My Coins".

This screenshot compares two sections: "Available Ads" and "My Ads". The "Available Ads" section shows one item: "Samsung 2018 Smart Phone" (Ad ID: 9YGP002TwaBAwMvzx4xX). The "My Ads" section shows three items: "Samsung 2018 Smart Phone" (Ad ID: 4m9e3YzgUJm0MQiGill), "Readme 2019 Smart Phone" (Ad ID: Sskku9P8YeV7MD01mg), and another "Samsung 2018 Smart Phone" (Ad ID: 11ceef9nkP23dnI6mUe1). A "Send Request" button is located at the bottom.

Available Ads separated by categories

This screenshot shows a product listing for a Laptop. The listing includes a photo of the laptop, the category "Computer & Accessories", the sub-category "Laptop", the location "Trivandrum, Kerala", and the price "18000 Coins" (last updated "July 1, 2022"). To the left, there is a sidebar titled "Categories" with a list of product types: All Products, Computer & Accessories, Furnitures, Phone & Accessories, Camera & Accessories, Vehicles, Electronics, Sports & Gaming, and Rare Products. Navigation arrows and page numbers (1, 2, 3) are at the bottom.

Add Post



Demand Product Details

Select Categories
Demand Product
Demand Brand/Material
Demand Color
Year of Manufacture

Submit

Requested Products as a seller

Requested & Available Ads

Action	Photo	Title	Coins
* Requested		Samsung 2018 Smart Phone Ad ID: 4m9E3YkZgIJUm0MQIGII	10000
* Available		Samsung 2018 Smart Phone Ad ID: 9YGP002TwBaWmVzx4XK	10000

Approval Requests as a buyer

Requested & Available Ads

Action	Photo	Title	Coins
* My Product		Samsung 2018 Smart Phone Ad ID: 4m9E3YkZgIJUm0MQIGII	10000
* Requested Product		Samsung 2018 Smart Phone Ad ID: 9YGP002TwBaWmVzx4XK	10000

Approve **Reject**

Received and sold as a seller

Received & Sold Ads

Action	Photo	Title	Coins
* Requested Product		Samsung 2018 Smart Phone Ad ID: 4m9E3YkZgUJm0MQiGll	10000
* My Product		Samsung 2018 Smart Phone Ad ID: 9YGP002TwaBAwMvzx4xX	10000

Satisfied **Not Satisfied**

Sold and Received as a buyer

Sold & Received Ads

Action	Photo	Title	Coins
* My Product		Wood 2018 Teak Chair Ad ID: 6uXmPaafahTy11Qe5RQ	10000
* Requested Product		Sony 2018 Smart Phone Ad ID: JAslKoV2ig286weVzSL	10000
* My Product		Samsung 2018 Smart Phone Ad ID: 4m9E3YkZgUJm0MQiGll	10000
* Requested Product		Samsung 2018 Smart Phone Ad ID: 9YGP002TwaBAwMvzx4xX	10000

Satisfied **Not Satisfied**

Satisfied received products as a buyer

Received & My Ads

Action	Photo	Title	Coins	Owner
* My Product		Samsung 2018 Smart Phone Ad ID: 4m9E3YkZgUJm0MQiGll	10000	don
* Received Product		Samsung 2018 Smart Phone Ad ID: 9YGP002TwaBAwMvzx4xX	10000	binu

Satisfied by the owner binu and don on June 26, 2022, 3:01 a.m.

Satisfied received products as a seller

The screenshot shows a user profile for 'binu' with a circular photo. Below the photo are two buttons: 'Profile' and 'My Ads'. To the right, a section titled 'Requested & Available Ads' lists two entries:

Action	Photo	Title	Coins	Owner
* Get Product		Samsung 2018 Smart Phone Ad ID: 4m9E3YkZgUJm0MQiGill	10000	don
* Given Product		Samsung 2018 Smart Phone Ad ID: 9YGP002TwaBAwMvzx4xX	10000	binu

A note below the table states: 'Satisfied by the User binu and don on June 26, 2022, 3:01 a.m.'

Home Page with chatbot

The home page features six categories: Computer & Accessories, Furnitures, Phone & Accessories, Electronics, Vehicles, and Sports & Gaming. The Sports & Gaming category is highlighted with a pink background. On the right, a chatbot window is open with the following conversation:

- TitTatAgent: hi
- TitTatAgent: 11:49 AM
- User: I want to sell a product
- TitTatAgent: 11:49 AM
- TitTatAgent: You can sell your used product here
- User: did i get money
- TitTatAgent: 11:50 AM
- TitTatAgent: No, you will get a product based on your product value

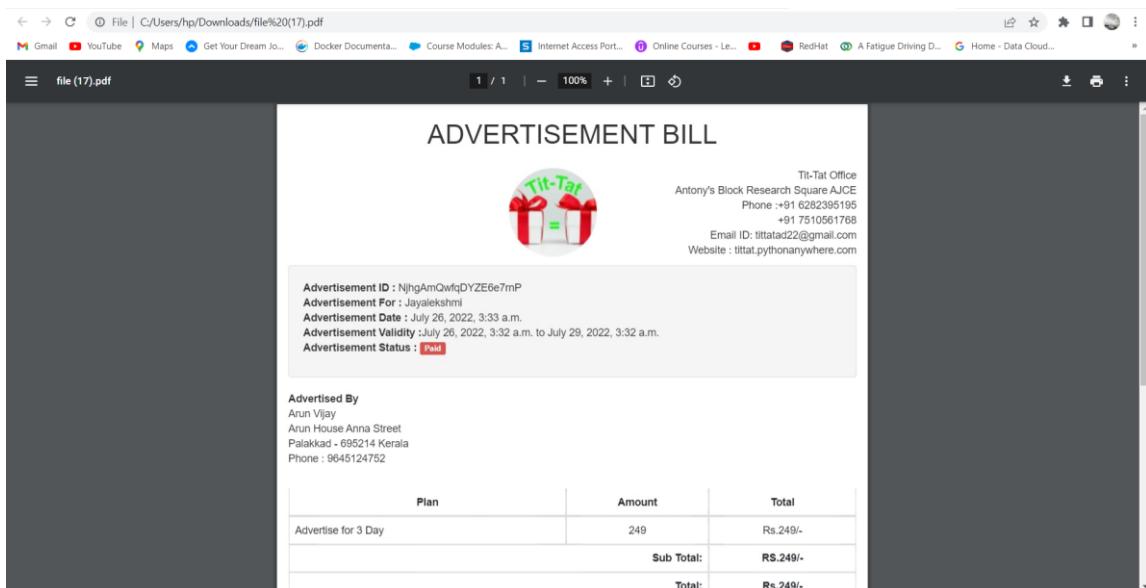
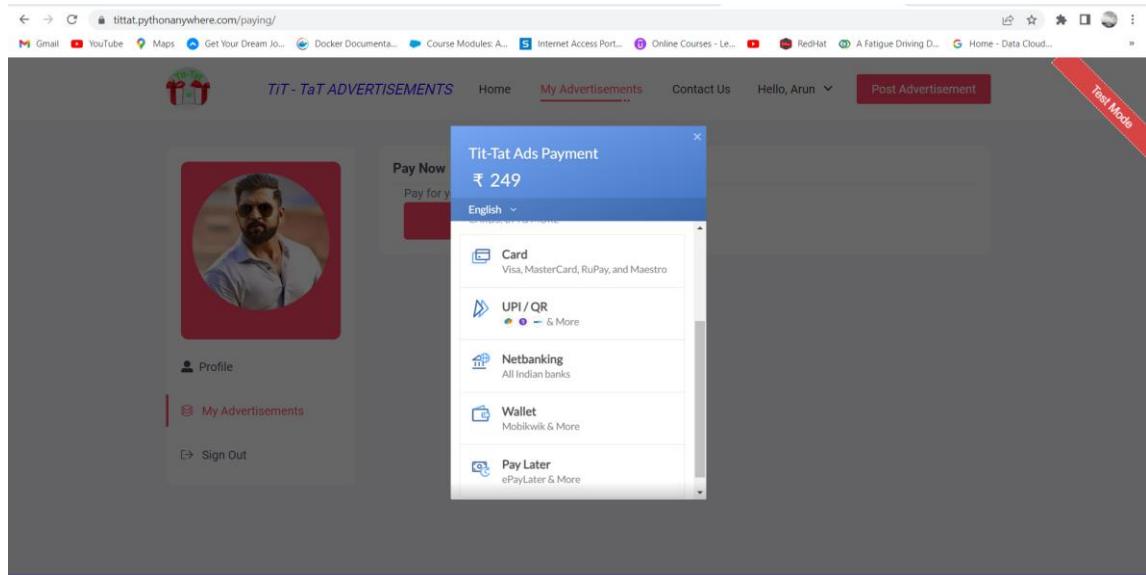
A text input field at the bottom says 'Type your message...'.

Paid Advertisements

The page title is 'TiT-TaT Advertisements'. It features four large advertisement cards:

- HP**: Business with U.
- GROCERY**: Enjoy good fortune and great deals with Pongal.
- UNIBRO**: THE WHITENESS OF DHOTI THE SOFTNESS OF T-SHIRT.
- MURUGAN**: 15% OFF + 50% OFF = 30% OFF.

Below each card is the text 'BOOST YOUR BUSINESS WITH U:'.



Evaluator dashboard

The screenshot shows the "Evaluator Dashboard" interface. At the top, there is a sidebar with links for "Profile", "Advertisements", "Evaluated Products", "In-Active Products", "Active Products", "Rejected Products", "Sold Products", and "Ads History From". The main area displays a table titled "Evaluator Dashboard" with columns: Photo, Title, Category, Ad Status, Date, and Coins. The table lists several ads and products:

Photo	Title	Category	Ad Status	Date	Coin
	Samsung 2018 Smart Phone Ad ID: 4m9E3YKZgUJm0MQiGll	Sports & Gaming	ACTIVE	June 26, 2022, 3:01 a.m.	10000
	Readme 2019 Smart Phone Ad ID: 5skkuy9P8Yle7MDQ1mg	Phone & Accessories	ACTIVE	July 8, 2022, 12:07 a.m.	4000
	None 2018 Teak Chair Ad ID: 6uXmPaafahTx11Qe5RQ	Furnitures	SOLD	July 4, 2022, 9:25 a.m.	10000
	Samsung 2018 Smart Phone Ad ID: 11ceeh9rnkP23dn16mUe1	Phone & Accessories	ACTIVE	July 6, 2022, 12:18 p.m.	14000
	Sony 2018 Smart Phone Ad ID: JAstIKoV2ig286weVZSL	Phone & Accessories	SOLD	July 10, 2022, 3:50 p.m.	10000

In-Active Products

Photo	Title	Category	Ad Status	Date	Coin
	Samsung 2018 Fridge Ad ID: HTIXZJefems061dtJQQW	Computer & Accessories	IN-ACTIVE	July 22, 2022, 2:33 a.m.	12000
	None 2018 Chair Ad ID: vbf0IUu025ytGHyC4Q	Furnitures	IN-ACTIVE	July 22, 2022, 11:40 a.m.	5200

History of products

Photo	Title	Category	Ad Status	Coin	Date	Action
	Readme 2019 Smart Phone Ad ID: Sakkuv9P8YleV7MDQ1mg	Phone & Accessories	Active	4000	July 8, 2022, 12:07 a.m.	
	None 2018 Teak Chair Ad ID: 6uXmPaafahXy11Qe5RQ	Furnitures	Inactive	10000	July 4, 2022, 9:25 a.m.	
	Samsung 2018 Smart Phone Ad ID: 1lceehHnkp23dn6mUe1	Phone & Accessories	Active	14000	July 6, 2022, 12:18 p.m.	
	Samsung 2018 Fridge Ad ID: HTIXZJefems061dtJQQW	Computer & Accessories	Inactive	12000	July 22, 2022, 2:33 a.m.	

Evaluation

Owner :

Binu Augustine

Bethel House Uchakkada
Trivandrum Ponkunnam
Phone : 8512351423

Click to See Number

Evaluate

Criteria for Furnitures Evaluation

All the details of the product should be evaluated honestly

Price is considered in Indian rupees, The coins will be same as the price

Per year depreciation - 2%
Per popular to Quality materials to non quality materials- 2% to 10%
Per dents small to large depreciation - 1% to 5%
Per scratch small to large depreciation - 1% to 5%

If the material used is wood there will be an evaluation based on the market price only

Based on dents, scratches etc.. the value of the product get reduced

Special price and evaluation for rare products

Evaluation will be based on the product details updated, online evaluation/offline evaluation

Evaluator predicted coins will be the final value/coins of the product

Product Details

Evaluation Mode

Category : Furnitures
 Yes No

Product : Chair
 Yes No

Material Used : Wood
 Yes No

Scratch and Dents if any : No Scratch
 Yes No

Design : No Dents

Admin Home

Admin Home

User Evaluator Products In-Active Products Processing Products

Approved Products Demanded Products Advertisemet Sign Out

Evaluated/Processing product to approve by admin

Status

Admin Here

Binu Augustine

Bethel House Uchakkada
Perambra, Thrissur
8512351423

Click to See Number

Approve Reject Home

PAPER NAME

second_submitted.pdf

WORD COUNT

6792 Words

CHARACTER COUNT

36213 Characters

PAGE COUNT

49 Pages

FILE SIZE

1.6MB

SUBMISSION DATE

Jul 19, 2022 2:24 PM GMT+5:30

REPORT DATE

Jul 19, 2022 2:25 PM GMT+5:30

● 14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 14% Internet database
- Crossref database
- 0% Publications database

● Excluded from Similarity Report

- Bibliographic material
- Cited material
- Quoted material
- Small Matches (Less than 8 words)

● 14% Overall Similarity

Top sources found in the following databases:

- 14% Internet database
- Crossref database
- 0% Publications database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	issuu.com	6%
	Internet	
2	geeksforgeeks.org	1%
	Internet	
3	itprojectsforyou.com	1%
	Internet	
4	origin.geeksforgeeks.org	<1%
	Internet	
5	tutorialspoint.com	<1%
	Internet	
6	slideshare.net	<1%
	Internet	
7	researchpublish.com	<1%
	Internet	
8	192.192.246.204	<1%
	Internet	
9	developers.hail.to	<1%
	Internet	

10	irigs.iiu.edu.pk:64447	<1%
	Internet	
11	whatis.techtarget.com	<1%
	Internet	
12	tudr.thapar.edu:8080	<1%
	Internet	
13	coursehero.com	<1%
	Internet	
14	omg.org	<1%
	Internet	
15	apidocs.callrail.com	<1%
	Internet	
16	inba.info	<1%
	Internet	
17	itsourcecode.com	<1%
	Internet	
18	mrlaptop.com.pk	<1%
	Internet	
19	silo.pub	<1%
	Internet	
20	widuri.raharja.info	<1%
	Internet	