“INSTANT DEAL”

A

Major Project

Submitted in partial fulfillment of the requirement for the Degree of

Bachelor of Engineering

In

Department of Computer Science & Engineering

Submitted To

BANSAL INSTITUTE OF SCIENCE & TECHNOLOGY,

BHOPAL (M.P.)

BGI BANSAL
GROUP OF INSTITUTES
BHOPAL|INDORE|MANDIDEEP

Submitted By
Rahul Gupta (0112cs191077)
Satyendra Kumar Saket (0112cs191097)
Rahi Turkar (0112cs191076)
Gagan Bisen (0112cs191046)
Ankit Kumar Chaubey (0112cs191026)

Under The guidance of

Prof. Manish Saxena

Department of Computer Science & Engineering

(SESSION: 2019-2023)

BANSAL INSTITUTE OF SCIENCE & TECHNOLOGY, BHOPAL (M.P.)

(Department of Computer Science & Engineering)

BGI BANSAL
GROUP OF INSTITUTES
BHOPAL|INDORE|MANDIDEEP

(SESSION: 2022-2023)

CERTIFICATE

*This is to certify that the entitled "***INSTANT DEAL***" being submitted by* **Rahul Gupta (0112cs191077), Satyendra Kumar Saket (0112cs191097), Rahi Turkar (0112cs191076), Gagan Bisen (0112cs191046), Ankit Kumar Chaubey (0112cs191026)** *student of VIII semesters of Bachelor of Engineering in the Department of Computer Science &Engineering***,** *has completed Major Project on "***INSTANT DEAL***" toward partial fulfillment of the requirement for the award of degree in Computer Science & Engineering of the RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.) is a record of bonafide work done by him/her under my supervision.*

Prof. Manish Saxena	***Prof. Manish Saxena***
Guide	HOD
Department of CSE	Department of CSE
B.I.S.T Bhopal (M.P.)	B.I.S.T Bhopal (M.P.)

BANSAL INSTITUTE OF SCIENCE & TECHNOLOGY, BHOPAL (M.P.)

(Department of Computer Science & Engineering)

(SESSION: 2022-2023)

DECLARATION

*We hereby declare that the work, which is being presented as the Major Project I Report entitled "***INSTANT DEAL***" in partial fulfillment of the requirements for the award of degree of* **Bachelor of Technology** *in* **Computer Science Engineering***, submitted in the department of Computer Science & Engineering, is our original work and the project has not formed the basisfor the award of any other degree, diploma, fellowship or any other similar titles.*

Place:

Date:

Rahul Gupta	(0112cs191077)
Satyendra Kumar Saket	(0112cs191097)
Rahi Turkar	(0112cs191076)
Gagan Bisen	(0112CS191040)
Ankit Kumar Chaubey	(0112CS191026)

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide **Prof. Manish Saxena** *for the guidance, inspiration and constructive suggestions that help us in the preparation of this project. We also thank our* **HOD Prof. Manish Saxena** *who took a step forward in correcting our mistakes and learning to not repeat them later.*

We would also like to thank **Dr. Damodar Tiwari**, *Director, BIST for whole hearted support.*

We are also grateful to our teachers for their constant support and guidance. We also wish to express our indebtedness to our parents as well as our family members for their blessings and support always helped me to face the challenges ahead

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

Rahul Gupta	(0112cs191077)
Satyendra Kumar Saket	(0112cs191097)
Rahi Turkar	(0112cs191076)
Gagan Bisen	(0112CS191040)
Ankit Kumar Chaubey	(0112CS191026)

ABSTRACT

The project "Instant Deal" is aimed at creating a platform for the buying and selling of second-hand bikes and scooters. The current system for purchasing and selling these vehicles can be tedious and time-consuming, involving multiple interactions with d by an alumnus from an institution is a long-time taking process. So, in our project we are trying to automate this process and saving time. In current scenario alumni has to visit institute first then he has to fill form for issuing T.C. then all the document verification, clear of dues and validation must be done on paper. But in our system, we are using the project "INSTANT DEAL" for automating the task in this project an alumni can apply online for transfer certificate. After that his details and credentials will be verified from the faculty members. Once all the details are verified, we will give green tick to alumni that yes, he is eligible for the issuing of transfer certificate. Now Alumni can book a slot for a particular working day of institution so that he can do a visit and collect transfer certificate. By this project we are automating the 4-5 days of manual process into a single day and providing ease in issuing of transfer certificate for alumni, staff and faculty members.

Group Members:

Rahul Gupta	(0112cs191077)
Satyendra Kumar Saket	(0112cs191097)
Rahi Turkar	(0112cs191076)
Gagan Bisen	(0112CS191040)
Ankit Kumar Chaubey	(0112CS191026)

TABLE OF CONTENTS

CHAPTER	Topic	Pg.-no.
1.	Introduction	1
2.	System Analysis	3
3.	Tools & Technology Specification	7
4.	Software Development and Methodology	15
5.	System Design	21
6.	Implementation	24
7.	Result Stimulation	40
9.	Future Scope	47
8.	References	48

CHAPTER 1: INTRODUCTION

1.1 Topic of the System

Title: - “INSTANT DEAL”

1.2 Project Abstract

The "Instant Deal" project is an online platform designed to simplify the process of buying and selling second-hand bikes and scooters. Our platform provides a secure and transparent environment for buyers and sellers to connect directly, with features such as a rating system, secure payment gateway, and automated tools for valuing vehicles and generating sales contracts.

1.3 Purpose of the System

The purpose of the "Instant Deal" project is to create an online platform for buying and selling second-hand bikes and scooters. The current process for buying and selling such vehicles is often time-consuming and lacks transparency.

Our system aims to simplify this process by providing a user-friendly interface where buyers and sellers can create accounts, list their vehicles, and browse listings. The system will also incorporate a rating and review system to increase transparency and trust between buyers and sellers.

By creating this platform, we are making it easier and more convenient for individuals to buy and sell second-hand bikes and scooters. This system will also help in reducing the environmental impact of these vehicles by promoting their reuse and reducing waste.

1.4 Target User

The target users of “Instant Deal” are individuals who are interested in buying or selling second-hand bikes and scooters. This includes students, commuters, and anyone looking for an affordable mode of transportation.

1.5 Overview

Instant Deal is a web-based platform designed for buying and selling second-hand bikes and scooters. The platform aims to simplify the process of buying and selling used vehicles while promoting their reuse and reducing waste.

The platform allows users to create accounts, list their vehicles, and browse listings based on their preferences. The search feature allows users to filter listings based on location, price, brand, and model. Additionally, Instant Deal incorporates a rating and review system to increase transparency and trust between buyers and sellers.

Sellers can list their vehicles for free and are only charged a fee once their vehicle is sold. The fee structure is designed to be affordable and transparent. The platform also provides various payment options to buyers and sellers for a hassle-free transaction.

Instant Deal is not only an affordable option for individuals looking for a mode of transportation but also promotes the reuse of vehicles, which helps to reduce waste and lessen the environmental impact of transportation. The platform's easy-to-use interface and user-friendly features make it a convenient option for anyone looking to buy or sell second-hand bikes and scooters.

CHAPTER 2: SYSTEM ANALYSIS

2.1 Existing System:

The existing system for issuing transfer certificates is manual and time-consuming. Alumni have to visit each department and wait for the faculty to verify their documents and clear dues. Our system automates this process, allowing alumni to book a slot on a specific day when all relevant faculty members are available. This reduces efforts and speeds up the process of issuing transfer certificates.

2.1.1 Limitations in Existing System:

- In the existing system, the process of buying and selling second-hand bikes and scooters is manual and time-consuming.
- Users have to physically visit multiple locations to list their vehicles, browse listings, and complete transactions.
- The lack of transparency and trust between buyers and sellers is a major issue.
- The existing system also fails to promote the reuse of vehicles, which can lead to increased waste and environmental impact.

2.2 Proposed System:

The proposed system will be an online platform where users can buy and sell second-hand bikes and scooters. Users can create an account and list their vehicles for sale with details such as price, mileage, and condition. Buyers can search for bikes and scooters based on their preferences, contact sellers directly, and negotiate prices. The system will have a secure payment gateway for transactions, and users can rate and review each other to build trust. The proposed system will be more convenient, efficient, and cost-effective than the existing methods of buying and selling vehicles.

2.2.1 Advantages over Existing System:

- Instant Deal offers a user-friendly and easy-to-use platform for buying and selling second-hand bikes and scooters, which saves time and effort for users.
- The search feature allows users to filter listings based on location, price, brand, and model, which makes it easier to find the desired vehicle.
- The rating and review system increases transparency and trust between buyers and sellers, providing a safer and more reliable platform.
- The affordable and transparent fee structure, along with various payment options, provides a hassle-free transaction experience for users.

2.3 Feasibility Study

Feasibility study is an analysis and evaluation of a proposed project to determine if it is feasible technical, economically, timely and operationally. Feasibility study for the project is conducted to analyze whether the proposed project is possible/ feasible to be developed within scheduled time, available resources, within estimated budget. After feasibility study is done, it is evaluated whether to proceed with the project or not. The feasibility study can be categorized into following four parts:

- Technically feasibility
- Economic feasibility
- Schedule feasibility
- Operational feasibility

2.4.1 Technical Feasibility

Technical feasibility is conducted to verify whether the project is feasible to be developed within the available resources or not. The technical feasible takes into account the technical requirements for the project, the technical resources required by the project for its successful and efficient completion. The specific technical resources to be available for completing the project successfully are given below:

Hardware Resources

Table 2.4.1: Hardware Resources

S. No.	Hardware Resources
1.	Processor- Core i3 and later version
2.	RAM (Memory) - 2 GB and more
3.	Monitor – Any standard monitor
4.	Pen Drive, Mouse, Printer, Keyboard
5.	System Type- 64-bit OS
6.	Hard Drive– More than 80 GB
7.	Accessories – Internet connection.

Software Resources

Table 2.4.2: Software Resources

S. No.	Software Resources
1.	XAMPP
2.	VS CODE
3.	Database Technology –MySQL
4.	Case Tools - Microsoft Project, Microsoft Visio and Star UML
5.	Documentation and Presentation tool - Microsoft Office 2010

Resources of Execution

Table 2.4.3: Resources for Execution

S. No.	Resources for Execution
1.	Operating system - Android 4.0 and later
2.	Android Smartphone - GPS enabled mobile phone, Rear Camera (minimum 3.2 MP), Internet services like Mobile Data.
3.	CPU: 600 MHZ
4.	RAM: 100 MB (estimated)
5.	Disk space: 50 MB (estimated)
6.	A good data/internet connection for customers

2.4.2 Economic Feasibility

The main purpose of conducting economic feasibility is to identify the financial benefits and costs associated with the project development. As for any system if the estimated benefits equal or surpasses the expected costs that is calculated for the system then the project would be economically feasible. Since the proposed system programming environment is based on android which is an open-source platform, so there is less of investment in the development portion. And any software or program that is produced needs to be tested which can be done without much investment. Only required investment would be a physical device to test the system.

2.4.3 Schedule Feasibility

Schedule feasibility for a project is done to verify whether the project can be completed within scheduled time or not as decided by the Gantt chart. Dates are fixed for each phase throughout the project and it is to be ensured that the project is able to complete within the specified dates and the schedule. If the project is able to be completed within the schedule, the scheduled feasibility is high. Gantt chart is a project management tool that can be used to measure the scheduled feasibility of the project. The proposed timeline for the application would consume approximately 38 weeks and time management for each task is carried out aptly through Gantt chart.

2.4.1 Operational Feasibility

Operational feasibility measures to which extent the proposed system resolves the problems identified in the starting phase of the project; how the project intends to fulfil the requirements identified during scope definition and how the system makes use of the opportunities identified during scope definition. The project will be operationally feasible undoubtedly as the main motive is to solve the problems of those who face problems in nowadays environment which are faced by masses.

2.4.2 Conclusion of Feasibility Study

The project is technically feasible since the hardware, software and the resources needed for executing the system are ready. After evaluating the cost and benefits incurred on and derived from the system, it is concluded that the project is economically feasible. The project will be completed on time and deadlines will be followed as scheduled in the Gantt chart. The system will be socially acceptable and will satisfy the needs of users.

2.6 Conclusion

Chapter 2 is all about the problems identified in the current area of study i.e. the problems in ticket environment. The developer has identified all the problems which is generally facing by the general user. The problems have been documented along with the justification for each problem. After all the problems are identified, the developer's next task was to provide appropriate solutions to resolve the problems. For the same purpose, the solution corresponding to each problem has been documented along with the justification. A feasibility study is then conducted for the system to ensure that the proposed system is technically, operationally, and economically feasible and can be completed within a specified time frame. The feasibility study is successful, creating a further path for the developer to move ahead with the project.

CHAPTER 3: TOOLS & TECHNOLOGY SPECIFICATION

3.1 HEROKU

Heroku is a cloud-based Platform as a Service (PaaS) that enables developers to deploy, manage, and scale web applications with ease. The platform provides a powerful and flexible infrastructure for hosting web applications and supports a wide range of programming languages, including Ruby, Node.js, Python, Java, and more. XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

With Heroku, developers can easily deploy their applications to the cloud, without worrying about server setup, maintenance, or scaling. The platform automatically manages load balancing, scaling, and application health, so developers can focus on building great applications, rather than managing infrastructure.

Heroku also provides a wide range of add-ons and integrations, including databases, email services, monitoring tools, and more. The platform is designed to be flexible and scalable, allowing developers to start small and scale up as their application grows.

3.2 Windows Server

It is a brand name for a group of server operating systems which is released by Microsoft. The first Windows server edition to be released with that brand was Windows Server 2003.

However, the first server edition of Windows was Windows NT 3.1 advanced server that followed by three Servers (Windows NT 3.5 Server, Windows NT 4.0 Server, and Windows 2000 Server); the latter was the first server edition to feature many things like Active Directory, DNS Server, DHCP Server, Group Policy, and many other popular features used today. Written in C, C++ and assembly[14].

3.3 MySQL

SQL stands for Structured Query Language. MySQL is an open source Relational Database Management System (RDBMS); it is a popular database for use in web applications, and is a central part of the greatly used LAMP (Linux, Apache, MySQL, Perl/PHP/Python) open-source web application software stack.

MySQL is used by many applications like, WordPress, Joomla, TYPO3, Drupal, MyBB, phpBB, MODX and other software. Numerous large scale websites including Google, YouTube, Facebook, Twitter, and Flickr arealso using MySQL.

Fig. 3.3-

On all platforms excluding Windows, MySQL sends with no GUI (Graphical User Interface) to administer MySQL databases or managing the data held within the databases. Users may install MySQL Workbench by downloading separately or simply may use the command line tools. Numbers of third party GUI tools are also available.

Swedish company has created MySQL which is written in C and C++. The first version of MySQL revealed on 23 may 1995. It has various versions. The general accessibility of MySQL

5.7 was broadcast in Oct 2015, and the version which is used in my project is 5.6.17.

3.5 Python

Python is a high-level, interpreted programming language that was first released in 1991. It is known for its simplicity, ease of use, and versatility. Python has a clear syntax that is easy to learn, making it a great language for beginners to start with. However, it is also powerful enough to be used in complex applications such as artificial intelligence and data science..

One of the advantages of Python is its extensive library of modules and packages that make development faster and more efficient. This includes libraries for web development, scientific computing, data analysis, machine learning, and more. Python is also cross-platform, meaning that it can run on multiple operating systems, including Windows, Linux, and macOS.

3.5 DjangoAdmin

Django Admin is a built-in application of the Django web framework that provides an easy-to-use user interface for managing a website's backend. It allows developers to create, read, update, and delete records in the database through a web interface.

The Django Admin site automatically generates forms, tables, and views for models registered with it, making it easy to manage website content without writing any code.

Django Admin is highly customizable, allowing developers to create their own views, forms, and templates for more advanced functionality.

Overall, Django Admin is a powerful and flexible tool that simplifies website management and reduces development time.

3.6 VS Code Editor

Visual Studio Code, also commonly referred to as VS Code,[9] is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

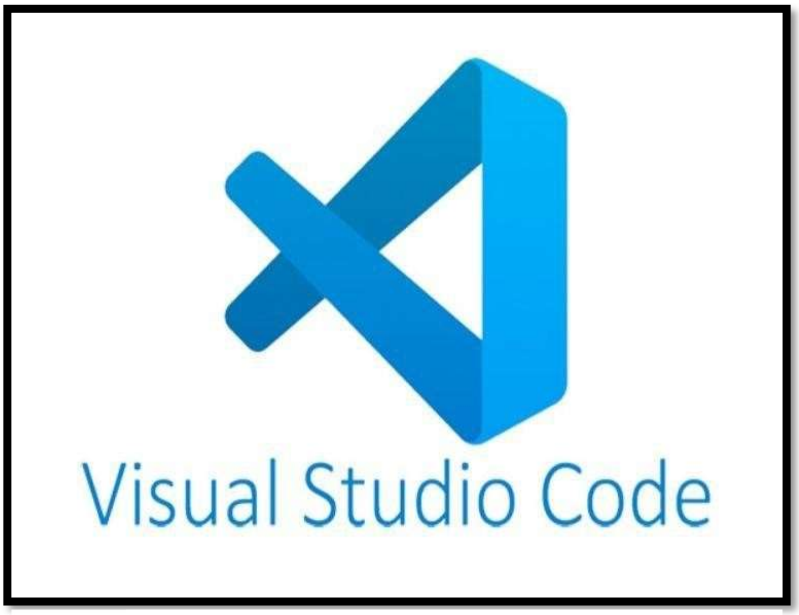


Fig. 3.6-

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 82,000 respondents, with 70% reporting that they use it.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran. It is based on the Electron framework,[20] which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

An orange version of the Visual Studio Code logo for the insiders' version of Visual Studio Code

Visual Studio Code Insiders logo

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports many programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

Visual Studio Code can be extended via extensions,[24] available through a central repository. This includes additions to the editor[25] and language support.[23] A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, time travel debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

3.7 HTML AND CSS

HTML stands for Hypertext Markup Language and CSS stands for Cascading Style Sheets are the crucial technologies for creating web pages. HTML supplies the structure of the page, and CSS the layout, for diversity of devices. Together with scripting and graphics, HTML and CSS are the fundamental of building Web Applications and Web pages.

Fig. 3.7-

HTML provides designers and developers the following facilities,

□ To design forms for directing transactions with remote services, for use in making reservation, searching for information, ordering products, and others

□ Retrieving online information through hypertext links.

- □ To include video and sound clips, spread sheets, and other applications straight in their documents

- □ Designer can publish online documents with text, headings, tables, photos and others.

CSS describes the Web pages presentation, involving layout, colors, and fonts. It enables the designer to adjust the presentation to various types of devices, like a small screens, large screens, or printers.

CSS is separate from HTML, and their separation makes it easy to preserve and maintain sites, share style sheets across pages, and accommodate pages to various environments.

3.8 Frameworks

3.8.1 Bootstrap

Bootstrap is front-end framework and collection of tools and mechanisms for building web applications. It consists of HTML and CSS based design templates for navigations, forms, buttons, typography, and other interface elements, and also JavaScript extensions.

Bootstrap is free and open source, and its purpose is to make easy the development of dynamic websites and web applications. It is the most starred project on GitHub, with more than 85,000 stars and 34,000 forks.

Fig. 3.8-

Bootstrap was developed by Mark Otto and Jacob Thomton and named Twitter Bluprint. Before Bootstrap framework, designers were using different libraries for interface development, which had many inconsistencies and their maintenance were difficult.

On 31 Jan, 2012, Bootstrap 2 was released. This framework has brought many changes to the existing components and, also, added 12 column grid layout and responsive design constituents. On August, 19, 2013, Bootstrap 3 was announced, which moved to first approach of mobile and using a flat design. The first alpha version of Bootstrap was spread out on 19, Aug, 2015.

3.8.2 JavaScript Framework(jQuery)

JQuery is JavaScript library intended to make simple the client-side scripting of HTML. It is the most popular JavaScript framework, which is free and open-source software licensed under the MIT License .

Fig. 3.9-

Several of the largest companies, including,

□ Google

□ IBM

□ Microsoft and

□ Netflix are using the jQuery.

3.8.3 Semantic UI

It is a development framework which is very helpful in building responsive and beautiful layouts utilizing human friendly HTML. In this framework the words and classes are treated as interchangeable concepts.

In Semantic UI framework, the Classes use syntax from natural language like plurality, the word order, and noun or modifier relationships to connect and join concepts naturally and instinctively.

Fig. 3.10-

3.8.4 Web Template

A website template (web template) is a pre designed webpage which any developer can use to plug-in their own text and text and script component and images into that to create a website. Website Templates are generally built with HTML and CSS code. By using web templates everyone is allowed to setup a website without hiring a professional web designer or developer. This brings the facility for anyone to create a logically priced business or a personal web that may be listed in search engines thus users can seek for your particular service.

CHAPTER 4: SOFTWARE & DEVELOPMENT METHODOLOGY

Software requirement specification considers all the phases of the software Life cycle, which begins with a formal problem specification, and programmers to the design of a solution, it simple mentation as a program, testing of the program and maintenance. Software engineer develop software tools and collections of tools called programming environments to improve the development process. For example, tools can help to manage the many components of a large program that Is being written by a team of programmers.

Software requirement specification is an abstract description of structured and methodological development and modification process applied to the main stages of producing and developing software.

4.1 Requirements Analysis

The needs of users are called requirements. To make these requirements computerized into software, we need to specify the basic needs and wishes of users to do this we have to think about the problem domain of the system that user requests processes to do this, this is requirement engineering.

4.2 Input Media and Device

Source data are input into the system through the keyboard. For example, If we fill the form of vaccinated people then we use the keyboard for input the required information.

4.3 Controlling The Amount of Data

In this system reducing data requirement is a major concern. By reducing input requirement, we speed the entire process from data capture to processing to provide results to user.

4.4 Avoiding Delays

When processing is delayed wing to data preparation or data entry the cause is called abbot bottleneck. To avoid bottlenecks when designing input is one of the objectives of this system.

4.5 Keeping The Process Simple

In this system achievement of all the objectives are mentioned in the simplest possible manner which is user friendly.

4.6 Software Engineering

Software engineering process required the development of software properly. It is important to understand the steps of system engineering. It helps to design software develop a software properly.

4.7 Development Methodology

As illustrated by (Erickson, 2005) software methodology provides a systematic pathway or a road map for the successful development of software. It helps to reduce risks; help handle the complexity of the software and develop the software within budget and time with fulfilling requirements with quality. It helps to determine if the development is deviating from the scheduled plan. The developer has evaluated almost all the available methodologies to come up with the best suitable methodology that would meet the requirements of the project such as technical, organization, and project considerations. After a brief analysis and research study of the software development methodologies, the developer has concluded to implement "V-MODEL" for the development of the project.

V-model Methodology

After reviewing all software methodologies, the developer chose "V-MODEL" as it fits in with the constraints of the project. Any kind of changes in the system can be implemented at a minimum cost because of its frequency of new increments that are produced. The modifications can be discussed and features can be improvised or deleted based on the feedback. This would effectively give its customer the finished system they want or need. This methodology, divides its tasks into smaller time frames so that targets can be achieved.

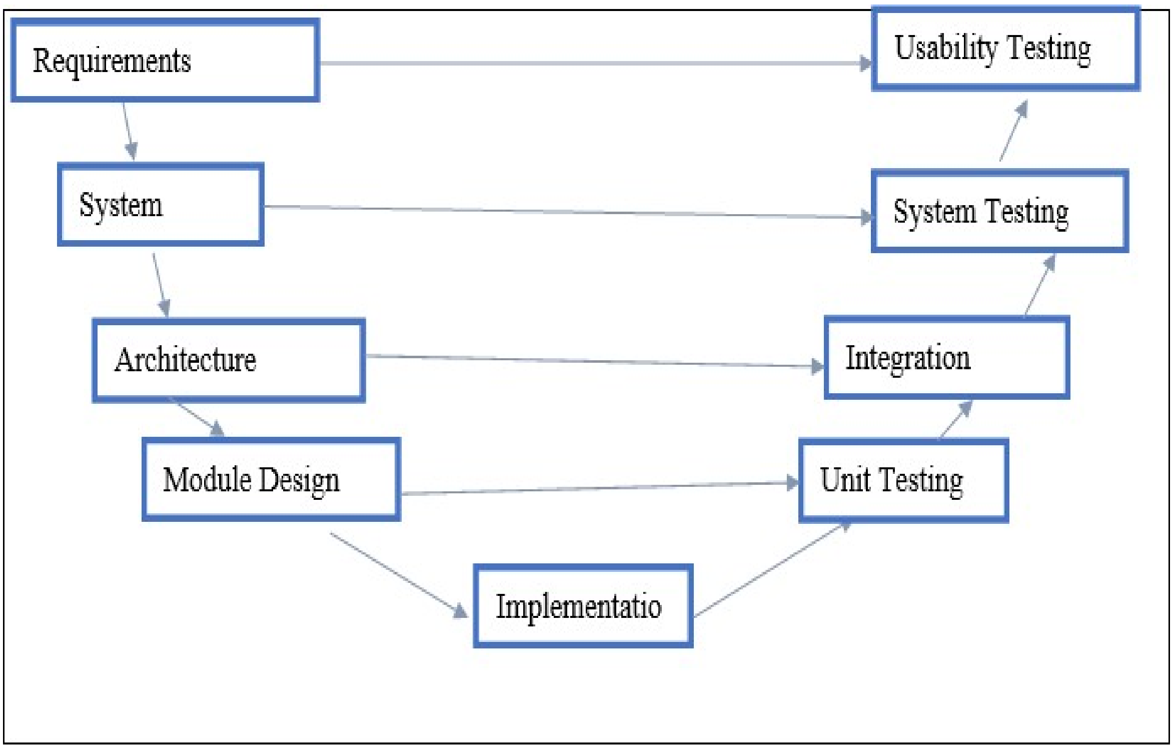


Fig. 4.7-

Picture: Phases of V-Model Methodology

This methodology has following phase that developer will pursue while developing the system.

 Requirement Analysis-It is the first step in the verification process. In this stage the developer will not be going to deliberate how the system is going to be built; it is going to be a generalized dialogue and a user requirement document is put onwards. This document will convey information regarding the function of the system, performance, security, data, interface etc. This document is required by the business analysts to convey the function of the system to the users. Meanwhile, it will simply be a recommendation.

 System Design-In this stage the possible design of the product is expressed. It is framed after keeping in mind the requirement summaries. Furthermore, while pursuing the documents, if there is somewhat that doesn't appropriate in the design, then the developer is made responsive of it and changes are consequently scheduled.

 Architecture Design- It is also known as the computer architecture design or the software design should understand the modules and the functionality of the modules which have to be integrated.

 Module Design-In this stage, the architectural design is again fragmented into sub components accordingly they can be planned and described distinctly. The units are known as modules. It can separately be decrypted by the programmer.

The Validation Phases of the V model

 Unit Testing-It is design in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps reduce bugs at an early stage, however all faults cannot be revealed by unit testing.

 Integration Testing or Interface Testing- It is related with the architectural design phase. Integration tests are performed to test the existence and communication of the internal modules within the system. In other words, in this phase the separate entities will be tested together to find out the faults in the interfaces.

 System Testing-It is directly allied with the System design phase. Its check the entire system functionality and the statement of the system under development with peripheral systems. In addition, most of the software and hardware compatibility issues can be revealed during system test execution.

 Acceptance Testing-In the acceptance testing, related with the business requirement analysis phase and comprises testing the product in user environment. It exposes the compatibility issues with the other systems accessible in the user environment. Acceptance tests alsodetermines the non-functional concerns such as load and performance faults in the actual userenvironment.

 Release Testing-It is in at this time that decision has to be made if the product or software which is created is suitable for the end user.

Justification for choosing V-Model

After some discussion, the developer chooses Waterfall Model for the development of this system. The most attractive factor for selection of V-Model is-

 Stable project requirements: As in our project most of the user requirements are restriction at the time of PSF so it specifies an unchanging project requirements and Waterfall methodology entirely supports a project which has requirements definite in advance.

 Progress of system is measurable: After each phase, it produces the documentation and as the structure of our Final year project we need to submit the documentation after each phase so it will be best suitable.

 Strict sign-off requirements: As the developers aim will be to content the user and until the user will be gratified the developer will be providing the user preferred functionalities and proper features so this methodology will be best appropriate.

 The highlighting on requirements and design before writing a single line of code confirms minimal wastage of time and effort and reduces the risk of schedule slippage, or of end user expectations not being met.

 In modified waterfall model life cycle phases are acceptable to overlay. Because of the phases overlap, a lot of suppleness has been familiarized in the modified waterfall model in software engineering. Meanwhile, a number of tasks can function concurrently, which ensures that the defects in the software are removed in the development stage itself and the overhead cost of making changes to the software before implementation is saved.

 Making changes to the basic design is also possible, as there are a number of phases lively at one point of time. In circumstance, there are any errors introduced because of the changes made, rectifying them is also easy (Testing can be done). This helps to reduce any error concerns.

4.7 Development Plan

Table 4.7: Development Plan

TOTAL DURATION	15 weeks	START DATE	2nd JAN, 2023	END DATE	22th APRIL, 2023
PHASE AND DURATION	**TASKS**				
PROJECT DEFINITION *Duration: 2 weeks*	↳ Idea Generation ↳ Project Title Selection ↳ Abstract Draft Project Proposal ↳ Project Proposal Form				

PROJECT PLANNING *Duration: 2 weeks*	➥ Work Breakdown Structure ➥ Schedule and Time Estimation ➥ Gantt Chart
REQUIREMENT ANALYSIS *Duration: 2 weeks*	➥ **Define and Finalize Requirement Specification** » Project Background » Problem Context » User Requirements » Set Objectives » Identify Scope of Project » Features and Functionalities ➥ **Organizing Project Specification Form** ➥ **Research and Analysis** » **Research** ✓ Academic Research ✓ Secondary Research ✓ Human-Computer Interaction » **Analysis** ✓ Domain Analysis ✓ Existing System Analysis ✓ User Requirements ✓ User Profiling and Modelling ✓ Risk Analysis
SYSTEM DESIGN *Duration: 2 Weeks*	➥ **Navigational Design** » Storyboarding ➥ **Abstract Interface Design** » Functionality Design » Interactivity Design ➥ **Design for Test Plan** » Acceptance Test » System Test » Interface Test
PROTOTYPING	➥ Creating Prototypes ➥ Evaluate Prototype feedbacks

Duration: 2 weeks	
PRODUCTION AND IMPLEMENTATION *Duration: 2 weeks*	↳ Program Code Generation ↳ Integrating Backend and Front end ↳ Module creation and Integration ↳ Implementation of working system ↳ Midpoint Interview
TESTING & EVALUATION *Duration: 2 weeks*	↳ Prototype Evaluation ↳ Test Plans ↳ Unit Testing ↳ Integration Testing ↳ System Testing ↳ Critical Evaluation
PROJECT ENDING *Duration: 1 weeks*	↳ Submission of the finished product

4.8 Conclusion

Chapter 4 is all about the research methods. The researcher conducted primary and secondary research under which technical research was conducted to come to a final conclusion of user requirements and technologies and tools to be used. Couple of research methods have been used by the developer, which includes Questionnaires and Interviews. The research was necessary to avoid waste of time in a later point of development phase. Now, the researcher is pretty sure of the features to be included in the system and how to remove existing problems. The developer is confident enough to implement the proposed app after performing extreme research related to domain, technology, language, tools etc.

CHAPTER 5: SYSTEM DESIGN

5.1 Introduction

The design phase starts when the developer is done with the research and analysis phase. The design phase includes the transformation of user's specifications and software components into the software interface to build a platform for establishing the system. It covers the building of blue prints of the system that include physical modelling using software engineering tools and methods.

In the design phase for the proposed system, the design is taking the requirements as the input and will produce a guideline for the implementation as output. Three to four design elements are to be produced for each and every module after completing the research and analysis part which will be fed as input to the design.

Class Diagram

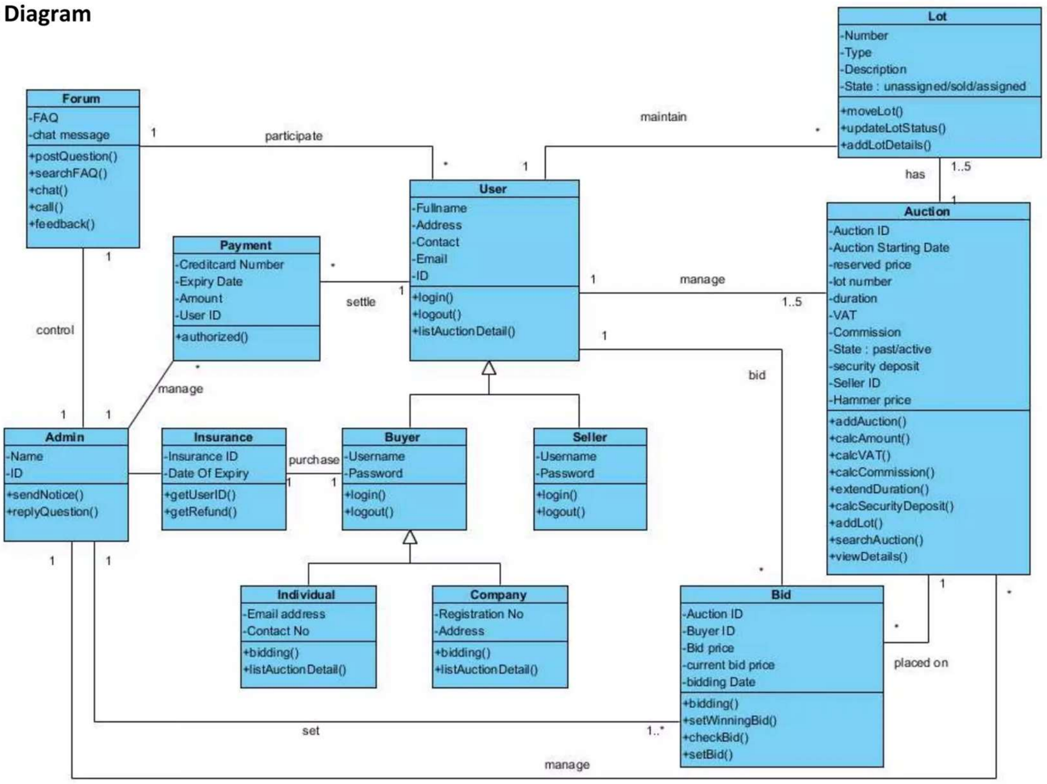


Fig. 5.1-

5.2 UML Diagram

Use Cases: Use case will be required by the developer so that the flow of the functionalities mentioned in the proposed system could be reviewed to validate the architecture and evaluation of the complexity level of individual modules could be assessed easily. Therefore, it would be helpful in testing the system through forward and reverse engineering. (Booch, Rumbaugh & Jacobson, 2008)

Activity Diagrams: Activity Diagrams will be used by the developer to show the flow of control of program modules from activity to activity. This would help us to show the concurrency as well as branches of control in the proposed system. (Booch, Rumbaugh & Jacobson, 2008)

Sequence Diagrams: Sequence Diagrams would help the developer to show the series of interactions prevailing between users and system/modules. (Booch, Rumbaugh & Jacobson, 2008)

Class Diagrams: UML class diagrams will be required to provide an independent description of the objects that would be used in the system and which would be helpful in the implementation phase. The design can be used as a reference to implement on any platform using any development environment. (Booch, Rumbaugh & Jacobson, 2008)

Entity-Relationship Diagram (ERD): ER diagrams will be required by the developer to identify the entities and relationship between them is beneficial in storing the data which would be then converted into tables in the normalization phase. (Booch, Rumbaugh & Jacobson, 2008)

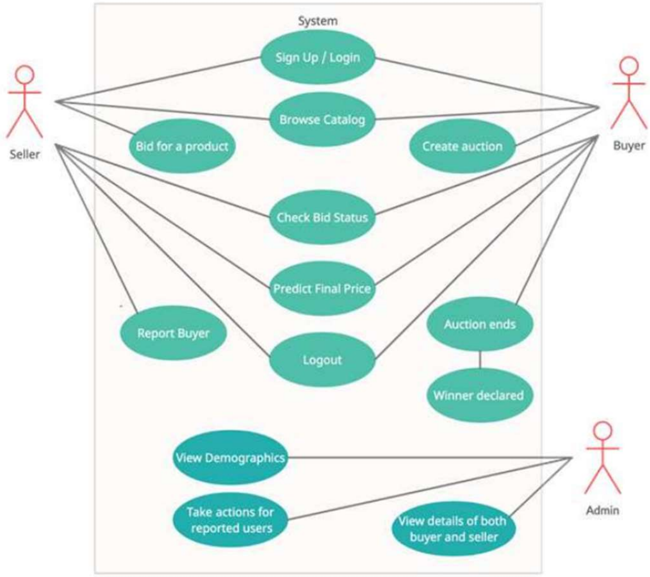
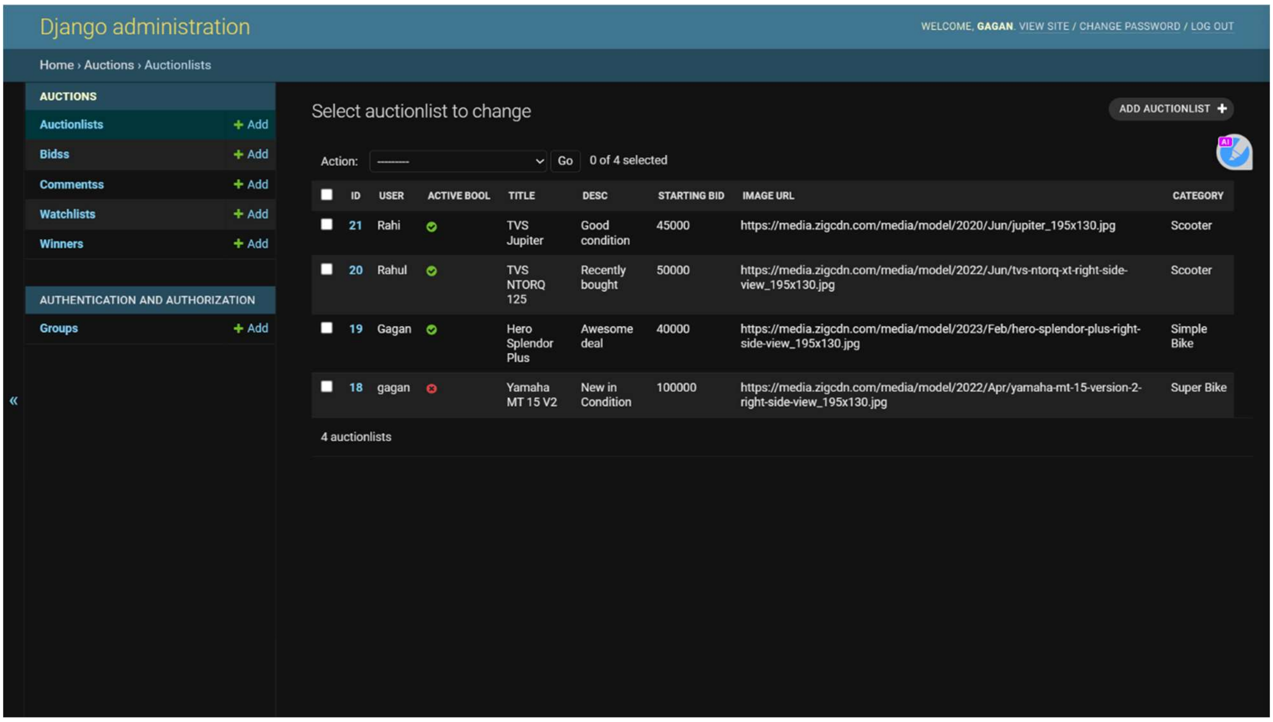


Fig. 1 : Use Case Diagram

5.3 Database Tables Structure



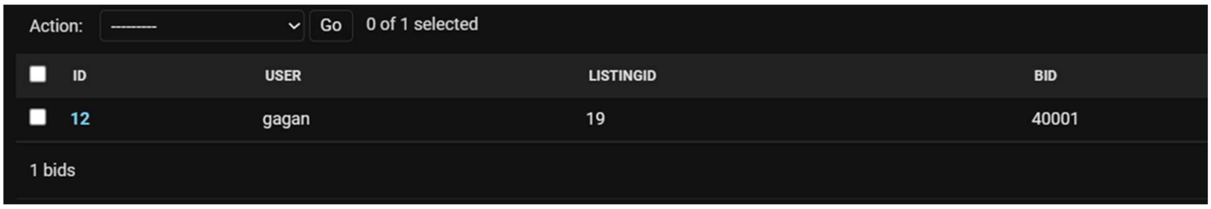
Select auctionlist to change

Action: --------- Go 0 of 4 selected

□	ID	USER	ACTIVE BOOL	TITLE	DESC	STARTING BID	IMAGE URL	CATEGORY
□	21	Rahi	✓	TVS Jupiter	Good condition	45000	https://media.zigcdn.com/media/model/2020/Jun/jupiter_195x130.jpg	Scooter
□	20	Rahul	✓	TVS NTORQ 125	Recently bought	50000	https://media.zigcdn.com/media/model/2022/Jun/tvs-ntorq-xt-right-side-view_195x130.jpg	Scooter
□	19	Gagan	✓	Hero Splendor Plus	Awesome deal	40000	https://media.zigcdn.com/media/model/2023/Feb/hero-splendor-plus-right-side-view_195x130.jpg	Simple Bike
□	18	gagan	✗	Yamaha MT 15 V2	New in Condition	100000	https://media.zigcdn.com/media/model/2022/Apr/yamaha-mt-15-version-2-right-side-view_195x130.jpg	Super Bike

4 auctionlists

Fig. 5.4.1- Users and Their Ads

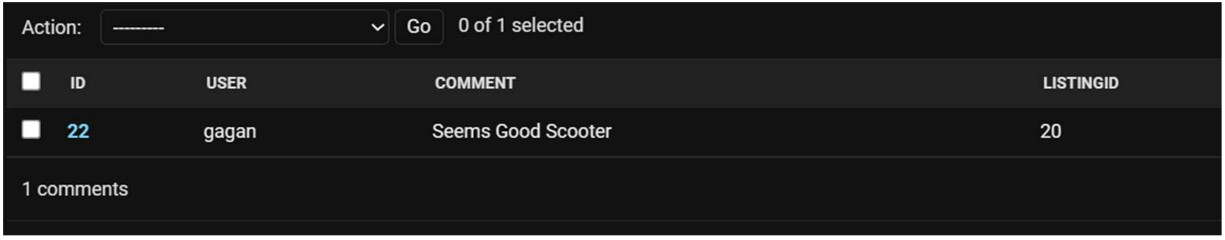


Action: --------- Go 0 of 1 selected

□	ID	USER	LISTINGID	BID
□	12	gagan	19	40001

1 bids

Fig. 5.4.2- Registered Users

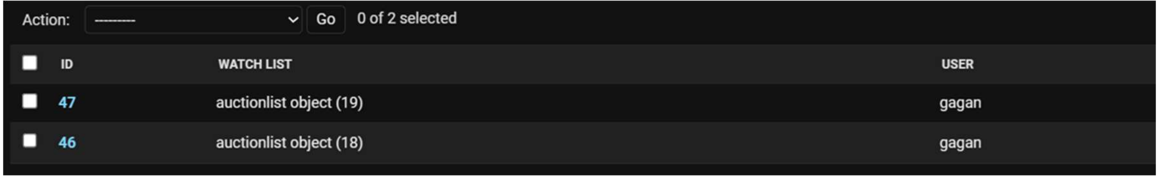


Action: --------- Go 0 of 1 selected

□	ID	USER	COMMENT	LISTINGID
□	22	gagan	Seems Good Scooter	20

1 comments

Fig. 5.4.3- Comments on the Product



Action: --------- Go 0 of 2 selected

□	ID	WATCH LIST	USER
□	47	auctionlist object (19)	gagan
□	46	auctionlist object (18)	gagan

Fig. 5.4.4- Watch List Products

CHAPTER 6: IMPLEMENTATION

6.1 Introduction

After the completion of the design phase, the developer analyzed that it was necessary that the requirements quoted down must meet the proposed system development schema. Implementation phase is one of the most essential stages of any project as the developer would work to implement the system design into a real system. Implementation phase involves the vivid usage of tools to convert the design modules into a real working system. For working on the real system, the prerequisite for the developer would be a programming language and a programming tool

6.2 Tools used for Implementation

The developer has mentioned all the tools that has been used in the development of the system.

Table 6.1: Tools used for Implementation

Tools	Purpose
Windows 10	Used as an operating system to run the all the below-mentioned tools for the development of the system.
Microsoft Visio 2019	Used to draw all the UML diagrams.
Microsoft Word 2019	Used to document the project.
Microsoft Project 2019	Used to prepare the Gantt Chart.
Microsoft PowerPoint 2019	Used to prepare the presentation slide.
Vs Code	It is used to design the pages of the application.
Apache	It is used to run the backend of the project
MySQL 5.6	Used as a remote database (in the server side).

6.3 Coding

6.3.1 Landing Page

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>{% block title %}Auctions{% endblock %}</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
```

```
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
        <link href="{% static 'auctions/styles.css' %}" rel="stylesheet">
        <link
href="https://fonts.googleapis.com/css2?family=Nunito:wght@300&display=swap"
rel="stylesheet">
    </head>
    <body>
        <h1>Instent Deal</h1>
        <div>
            {% if user.is_authenticated %}
                Signed in as <strong>{{ user.username }}</strong>.
            {% else %}
                Not signed in.
            {% endif %}
        </div>
        <ul class="nav nav-pills nav-fill">
            <li class="nav-item">
                <a class="nav-link" href="{% url 'index' %}">Active Listings</a>
            </li>

            <li class="nav-item">
                <a class="nav-link" href="{% url 'cat_list' %}">Categories</a>
            </li>

            {% if user.is_authenticated %}
                <li class="nav-item">
                    <a class="nav-link" href="{% url 'logout' %}">Log Out</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="{% url 'create' %}">Create Listing</a>
                </li>

                <li class="nav-item">
                    <a class="nav-link" href="{% url 'watchlistpage' user.username %}">Watch
List</a>
                </li>

                <li class="nav-item">
                    <a class="nav-link" href="{% url 'winnings' %}">Your Winnings</a>
                </li>



            {% else %}
                <li class="nav-item">
                    <a class="nav-link" href="{% url 'login' %}">Log In</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="{% url 'register' %}">Register</a>
```

```
                            </li>
                        {% endif %}
                    </ul>
                    <hr>
                    {% block body %}
                    {% endblock %}
                </body>
            </html>
```

- **Login Page**

```
{% extends "auctions/layout.html" %}

{% block body %}

    <h2>Login</h2>

    {% if message %}
        <div>{{ message }}</div>
    {% endif %}

    <form action="{% url 'login' %}" method="post">
        {% csrf_token %}
        <div class="form-group">
            <input autofocus class="form-control" type="text" name="username"
placeholder="Username">
        </div>
        <div class="form-group">
            <input class="form-control" type="password" name="password" placeholder="Password">
        </div>
        <input class="btn btn-primary" type="submit" value="Login">
    </form>

    Don't have an account? <a href="{% url 'register' %}">Register here.</a>

{% endblock %}
```

- **Register Page**

```
{% extends "auctions/layout.html" %}

{% block body %}

    <h2>Register</h2>

    {% if message %}
    <div>{{ message }}</div>
  {% endif %}

  <form action="{% url 'register' %}" method="post">
    {% csrf_token %}
    <div class="form-group">
      <input class="form-control" autofocus type="text" name="username"
placeholder="Username">
    </div>
    <div class="form-group">
      <input class="form-control" type="email" name="email" placeholder="Email Address">
    </div>
    <div class="form-group">
      <input class="form-control" type="password" name="password" placeholder="Password">
    </div>
    <div class="form-group">
      <input class="form-control" type="password" name="confirmation" placeholder="Confirm
Password">
    </div>
    <input class="btn btn-primary" type="submit" value="Register">
  </form>

  Already have an account? <a href="{% url 'login' %}">Log In here.</a>

{% endblock %}
```

- **Watchlist Page**

```
{% extends "auctions/layout.html" %}

{% block body %}
  <h2>Watch List</h2>
  <div id="activelist">
  {% for list in user_watchlist %}
  {% if list.watch_list.active_bool == True %}
  <div class = "card">
    <a class="title">{{list.watch_list.title}}</a>
    <a class= "cat">{{list.watch_list.category}}</a>
    {% if list.watch_list.image_url %}
      <img src="{{ list.watch_list.image_url }}" class="list_img">
    {% else %}
      <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/6/6c/No_image_3x4.svg/1024px-
No_image_3x4.svg.png" alt="No image Provided" class="list_img">
    {% endif %}
    <a><a class="c_price">Current Price: </a> {{ list.watch_list.starting_bid }} $ </a>
    <a href="{% url 'listingpage' list.watch_list.id %}">View Bid</a>
```

```
        <form action = "{% url 'deletewatchlist' %}">
        <button type = "submit" value = {{ list.id }} name = "listid" >Delete</button>
        </form>
    {% endif %}
    </div>
    {% endfor %}
    </div>

<style>

#activelist{
    margin-top: 4vw;
    grid-gap: 2vw;
    display:grid;
    width: 90%;
    margin-left: 5%;
    grid-template-rows: repeat(auto,1fr);
    grid-template-columns: 1fr 1fr 1fr;
    font-family: 'Nunito', sans-serif;
    text-align: center;
}
.card{
    display: grid;
    grid-template-rows: repeat(auto,1fr);
    grid-template-columns: 1fr;
    width: 90%;
    margin-left: 5%;
}

img{
        width: 100%;
    }

.title{

    font-size: 2vw;
    font-weight: bold;
}
.c_price{
    font-weight: bold;
}
button{
    width: 100%;
}
</style>
{% endblock %}
```

- **Winning Page**

```
{% extends "auctions/layout.html" %}

{% block body %}
   <h2>Your Winnings</h2>
   <div id="activelist">
   {% for list in user_winlist %}
   <div class = "card">
      <a class="title">{{list.bid_win_list.title}}</a>
      <a class= "cat">{{list.bid_win_list.category}}</a>
      {% if list.bid_win_list.image_url %}
         <img src="{{ list.bid_win_list.image_url }}" class="list_img">
      {% else %}
         <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/6/6c/No_image_3x4.svg/1024px-
No_image_3x4.svg.png" alt="No image Provided" class="list_img">
      {% endif %}
      <p class="desc">{{ list.bid_win_list.desc }}</p>
      <div>This bid was listed by <a class = "listuser">{{ list.bid_win_list.user }}</a>.</div>
   </div>
   {% endfor %}
   </div>

<style>

#activelist{
   margin-top: 4vw;
   grid-gap: 2vw;
   display:grid;
   width: 90%;
   margin-left: 5%;
   grid-template-rows: repeat(auto,1fr);
   grid-template-columns: 1fr 1fr;
   font-family: 'Nunito', sans-serif;
   text-align: center;
}
.card{
   display: grid;
   grid-template-rows: repeat(auto,1fr);
   grid-template-columns: 1fr;
   width: 90%;
   margin-left: 5%;
}

img{
      width: 100%;
   }

.title{

   font-size: 2vw;
   font-weight: bold;
}
.listuser{
   font-weight: bold;
}
button{
   width: 100%;
}
</style>
{% endblock %}
```

- **Catogery Page**

```
{% extends "auctions/layout.html" %}

{% block body %}
   <h2>All Categories</h2>
   <div id="activelist">
   {% for list in cat_list %}
   <div class = "card">
      <a class="title bg-light" href={% url 'cat' list.category %}>{{list.category}}</a>
   </div>
   {% endfor %}
   </div>

<style>

#activelist{

   margin-top: 4vw;
   grid-gap: 2vw;
   display:grid;
   width: 40%;
   margin-left: 30%;
   grid-template-rows: repeat(auto,1fr);
   grid-template-columns: 1fr;
   font-family: 'Nunito', sans-serif;
   text-align: center;
}
.card{
   display: grid;
   grid-template-rows: repeat(auto,1fr);
   grid-template-columns: 1fr;
   width: 90%;
   margin-left: 5%;
   background-color: grey;
}

img{
      width: 100%;
```

```
    }

.title{

    font-size: 2vw;
    color:black;
}
.listuser{
    font-weight: bold;
}
button{
    width: 100%;
}
</style>
{% endblock %}
```

- **Create Page**

```
{% extends "auctions/layout.html" %}

{% block body %}
    <h2 style = "font-weight: bold;text-align: center; font-family: Nunito;">Create Listing</h2>
    <div id="activelist">
    <form method="POST">
        {% csrf_token %}
        <input required placeholder="Title" autofocus type="text" name="create_title"/>
        <textarea placeholder="Add Description" required type="text" name="create_desc"></textarea>
        <input required placeholder="Starting bid" type="number" name="create_initial_bid"/>
        <input placeholder="Image URL (optional)" type="text" name="img_url"/>
        <input required placeholder="Category" type="text" name="category"/>
        <button class="btn btn-outline-primary" type="submit">Submit</button>
    </form>
    </div>

<style>

#activelist{
    width: 60%;
    margin-left:20%;

}
textarea{
    height: 10vw;
}
form{
    padding-top: 2vw;
    display:grid;
    grid-template-rows: repeat(auto,1fr);
    grid-template-columns: 1fr;
    grid-gap: 2vw;
    font-family: 'Nunito', sans-serif;
    }

.btn{

    font-family: Nunito;
    width: 30%;
    margin-left: 70%;
}
</style>
{% endblock %}
```

- **Admin .py**

```
from django.contrib import admin
from .models import *

class auction(admin.ModelAdmin):
    list_display = ("id" , "user", "active_bool","title" , "desc" , "starting_bid" , "image_url" , "category")

class watchl(admin.ModelAdmin):
    list_display = ("id", "watch_list" , "user")

class bds(admin.ModelAdmin):
    list_display = ("id","user","listingid","bid")

class comme(admin.ModelAdmin):
    list_display = ("id","user", "comment", "listingid")

class win(admin.ModelAdmin):
    list_display = ("id","user", "bid_win_list")

# Register your models here.
admin.site.register(auctionlist, auction)
admin.site.register(bids, bds)
admin.site.register(comments, comme)
admin.site.register(watchlist, watchl)
admin.site.register(winner, win)
```

- **Apps.py**

```
from django.apps import AppConfig


class AuctionsConfig(AppConfig):
    name = 'auctions'
```

- **Models.py**

```
from django.contrib.auth.models import AbstractUser
from django.db import models

class User(AbstractUser):
    pass

class auctionlist(models.Model):
    user = models.CharField(max_length=64)
    title = models.CharField(max_length=64)
    desc = models.TextField()           #CharField cannot be left without giving a max_length, Textfield
can
    starting_bid = models.IntegerField()
    image_url = models.CharField(max_length=228, default = None, blank = True, null = True)
    category = models.CharField(max_length=64)
    active_bool = models.BooleanField(default = True)

class bids(models.Model):
    user = models.CharField(max_length=30)
    listingid = models.IntegerField()
    bid = models.IntegerField()


class comments(models.Model):
    user = models.CharField(max_length=64)
    comment = models.TextField()
    listingid = models.IntegerField()


class watchlist(models.Model):
    watch_list = models.ForeignKey(auctionlist, on_delete=models.CASCADE)
    user = models.CharField(max_length=64)

class winner(models.Model):
    bid_win_list = models.ForeignKey(auctionlist, on_delete = models.CASCADE)
    user = models.CharField(max_length=64, default = None)
```

- **Urls.py**

```
from django.urls import path

from . import views

urlpatterns = [
    path("", views.index, name="index"),
    path("login", views.login_view, name="login"),
    path("logout", views.logout_view, name="logout"),
    path("register", views.register, name="register"),
    path("create", views.create, name="create"),
    path("auctions/<int:bidid>", views.listingpage, name="listingpage"),
    path("watchlist/<str:username>", views.watchlistpage, name = "watchlistpage"),
    path("added", views.addwatchlist, name = "addwatchlist"),
    path("delete", views.deletewatchlist, name = "deletewatchlist"),
    path("bidlist", views.bid, name="bid"),
    path("comments", views.allcomments, name="allcomments"),
    path("win_ner", views.win_ner, name="win_ner"),
    path("winnings", views.winnings, name="winnings"),
    path("cat_list", views.cat_list, name="cat_list"),
    path("categories/<str:category_name>", views.cat, name="cat"),
]
```

- **Views.py**

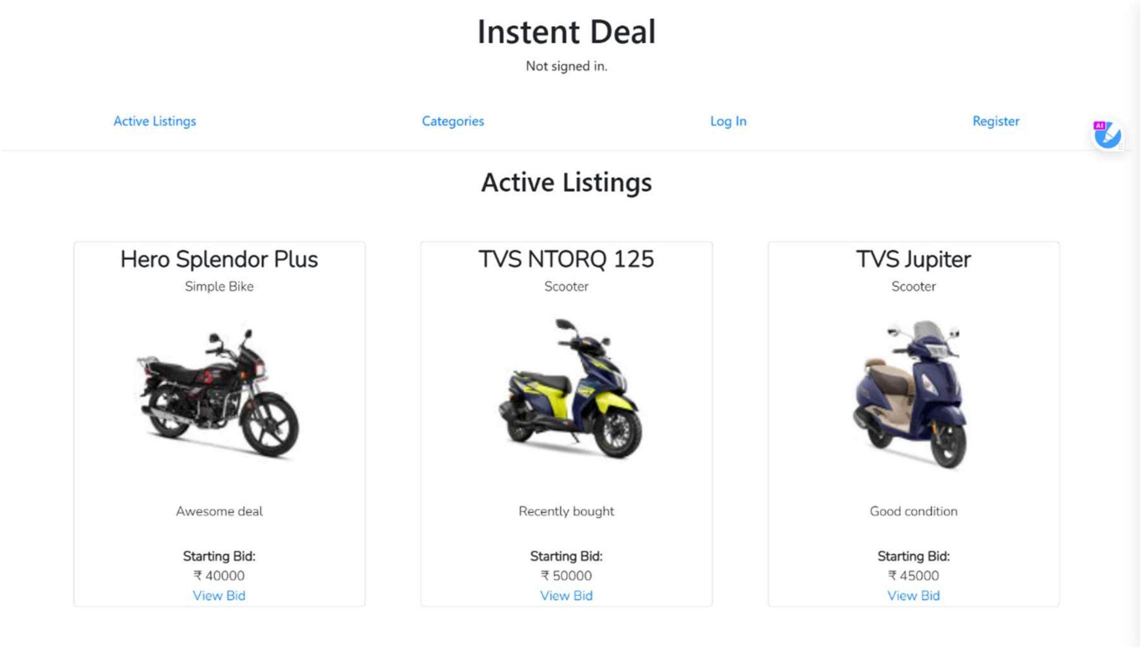
```
from django.contrib.auth import authenticate, login, logout
from django.db import IntegrityError
from django.http import HttpResponse, HttpResponseRedirect
from django.shortcuts import render, redirect
from django.urls import reverse
from .models import *
from django.contrib.auth.decorators import login_required
from django.contrib import messages


def index(request):
    return render(request, "auctions/index.html",{
        "a1": auctionlist.objects.filter(active_bool = True),
            })


def login_view(request):
    if request.method == "POST":

        # Attempt to sign user in
        username = request.POST["username"]
        password = request.POST["password"]
        user = authenticate(request, username=username, password=password)

        # Check if authentication successful
        if user is not None:
            login(request, user)
            return HttpResponseRedirect(reverse("index"))
        else:
            return render(request, "auctions/login.html", {
                "message": "Invalid username and/or password."
            })
    else:
        return render(request, "auctions/login.html")


def logout_view(request):
    logout(request)
    return HttpResponseRedirect(reverse("index"))


def register(request):
    if request.method == "POST":
        username = request.POST["username"]
        email = request.POST["email"]

        # Ensure password matches confirmation
        password = request.POST["password"]
        confirmation = request.POST["confirmation"]
        if password != confirmation:
            return render(request, "auctions/register.html", {
                "message": "Passwords must match."
            })

        # Attempt to create new user
        try:
            user = User.objects.create_user(username, email, password)
            user.save()
        except IntegrityError:
            return render(request, "auctions/register.html", {
                "message": "Username already taken."
            })
        login(request, user)
        return HttpResponseRedirect(reverse("index"))
    else:
        return render(request, "auctions/register.html")


@login_required(login_url='login')
def create(request):
    if request.method == "POST":
        m = auctionlist()
        m.user = request.user.username
        m.title = request.POST["create_title"]
        m.desc = request.POST["create_desc"]
        m.starting_bid = request.POST["create_initial_bid"]
        m.image_url = request.POST["img_url"]
        m.category = request.POST["category"]
        # m = auctionlist(title = title, desc=desc, starting_bid = starting_bid, image_url = image_url,
category = category)
        m.save()
        return redirect("index")
    return render(request, "auctions/create.html")



def listingpage(request, bidid):
    biddesc = auctionlist.objects.get(pk = bidid, active_bool = True)
    bids_present = bids.objects.filter(listingid = bidid)

    return render(request, "auctions/listingpage.html",{
        "list": biddesc,
        "comments" : comments.objects.filter(listingid = bidid),
        "present_bid": minbid(biddesc.starting_bid, bids_present),
    })


@login_required(login_url='login')
def watchlistpage(request, username):

    # present_w = watchlist.objects.get(user = "username")
    list_ = watchlist.objects.filter(user = username)
    return render(request, "auctions/watchlist.html",{
        "user_watchlist" : list_,
    })


@login_required(login_url='login')
def addwatchlist(request):
    nid = request.GET["listid"]

    # below line of code will select a table of watchlist that has my name, then
    # when we loop in this watchlist, there r two fields present, to browse watch_list
    # watch_list.id == auctionlist.id, similar for all

    list_ = watchlist.objects.filter(user = request.user.username)

    # when you below line, you shld convert id to int inorder to compare or else == wont work

    for items in list_:
        if int(items.watch_list.id) == int(nid):
            return watchlistpage(request, request.user.username)

    newwatchlist = watchlist(watch_list = auctionlist.objects.get(pk = nid), user =
request.user.username)
    newwatchlist.save()
        # this message remains untill u reload
    messages.success(request, "Item added to watchlist")

    return listingpage(request, nid)


@login_required(login_url='login')
```

```
def deletewatchlist(request):
    rm_id = request.GET["listid"]
    list_ = watchlist.objects.get(pk = rm_id)

    # this message remains untill u reload
    messages.success(request, f"{list_.watch_list.title} is deleted from your watchlist.")
    list_.delete()

    # you cannot call a fuction  from views as a return value
    return redirect("index")


# this function returns minimum bid required to place a user's bid
def minbid(min_bid, present_bid):
    for bids_list in present_bid:
        if min_bid < int(bids_list.bid):
            min_bid = int(bids_list.bid)
    return min_bid


@login_required(login_url='login')
def bid(request):
    bid_amnt = request.GET["bid_amnt"]
    list_id = request.GET["list_d"]
    bids_present = bids.objects.filter(listingid = list_id)
    startingbid = auctionlist.objects.get(pk = list_id)
    min_req_bid = startingbid.starting_bid
    min_req_bid = minbid(min_req_bid, bids_present)

    if int(bid_amnt) > int(min_req_bid):
        mybid = bids(user = request.user.username, listingid = list_id , bid = bid_amnt)
        mybid.save()
        messages.success(request, "Bid Placed")
        return redirect("index")

    messages.warning(request, f"Sorry, {bid_amnt} is less. It should be more than ₹{min_req_bid}.")
    return listingpage(request, list_id)


# shows comments made by different user and allows to add comments
@login_required(login_url='login')
def allcomments(request):
    comment = request.GET["comment"]
    username = request.user.username
    list_id = request.GET["listid"]
    new_comment = comments(user = username, comment = comment, listingid = list_id)
    new_comment.save()
    return listingpage(request, list_id)


# shows message abt winner when bid is closed
```

```
def win_ner(request):
    bid_id = request.GET["listid"]
    bids_present = bids.objects.filter(listingid = bid_id)
    biddesc = auctionlist.objects.get(pk = bid_id, active_bool = True)
    max_bid = minbid(biddesc.starting_bid, bids_present)
    try:
        # checks if anyone other than list_owner win the bid
        winner_object = bids.objects.get(bid = max_bid, listingid = bid_id)
        winner_obj = auctionlist.objects.get(id = bid_id)
        win = winner(bid_win_list = winner_obj, user = winner_object.user)
        winners_name = winner_object.user

    except:
        #if no-one placed a bid, and if bid is closed by list_owner, owner wins the bid
        winner_obj = auctionlist.objects.get(starting_bid = max_bid, id = bid_id)
        win = winner(bid_win_list = winner_obj, user = winner_obj.user)
        winners_name = winner_obj.user

    #Check Django Documentary for Updating attributes based on existing fields.
    biddesc.active_bool = False
    biddesc.save()

    # saving winner details
    win.save()
    messages.success(request, f"{winners_name} won {win.bid_win_list.title}.")
    return redirect("index")

# checks winner
def winnings(request):
    try:
        your_win = winner.objects.filter(user = request.user.username)
    except:
        your_win = None

    return render(request, "auctions/winnings.html",{
        "user_winlist" : your_win,
    })

#shows lists that are present in a specific category
def cat(request, category_name):
    category = auctionlist.objects.filter(category = category_name)
    return render(request, "auctions/index.html",{
        "a1" : category,
    })

#shows all categories in which object is listed
def cat_list(request):

    # unlike filter that takes a values of object_name in model, to
    # display objectname use .values('name of section from your object')
    # and when you add distinct() along with it
    # it shows only unique names, omits duplicates

    category_present = auctionlist.objects.values('category').distinct()
    return render(request, "auctions/category.html",{
        "cat_list" : category_present,
    })
```

CHAPTER 7: RESULT STIMULATION

Landing Page Output



Instent Deal

Not signed in.

Active Listings | Categories | Log In | Register

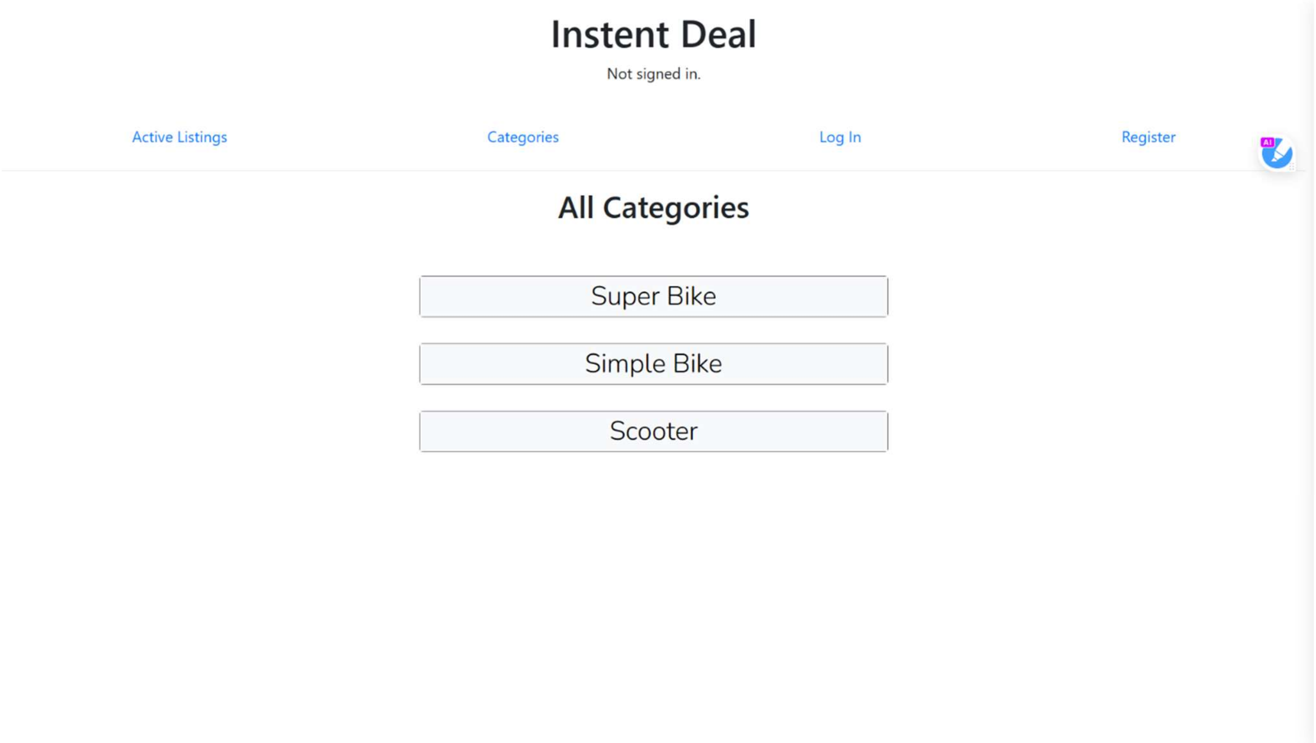
Active Listings

Hero Splendor Plus
Simple Bike
Awesome deal
Starting Bid:
₹ 40000
View Bid

TVS NTORQ 125
Scooter
Recently bought
Starting Bid:
₹ 50000
View Bid

TVS Jupiter
Scooter
Good condition
Starting Bid:
₹ 45000
View Bid

Category Page



Instent Deal

Not signed in.

Active Listings | Categories | Log In | Register

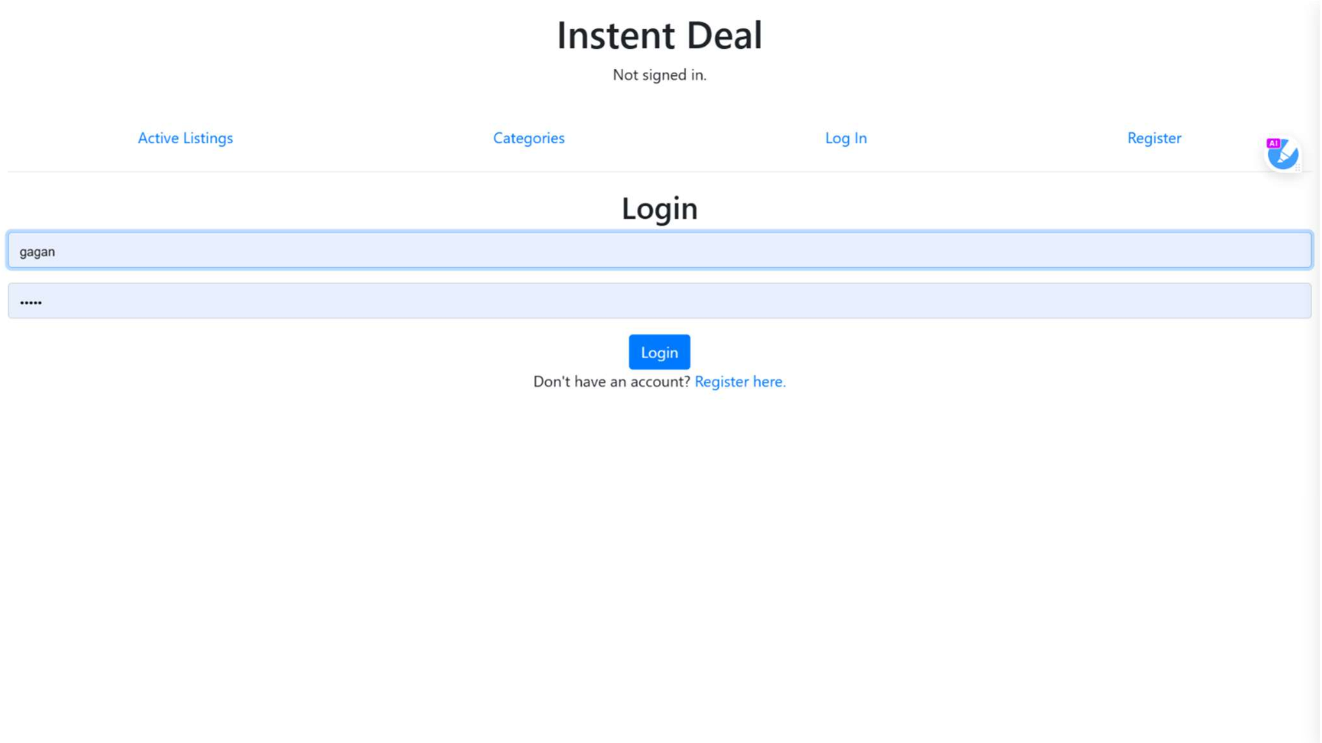
All Categories

Super Bike

Simple Bike

Scooter

Login Page



Instent Deal

Not signed in.

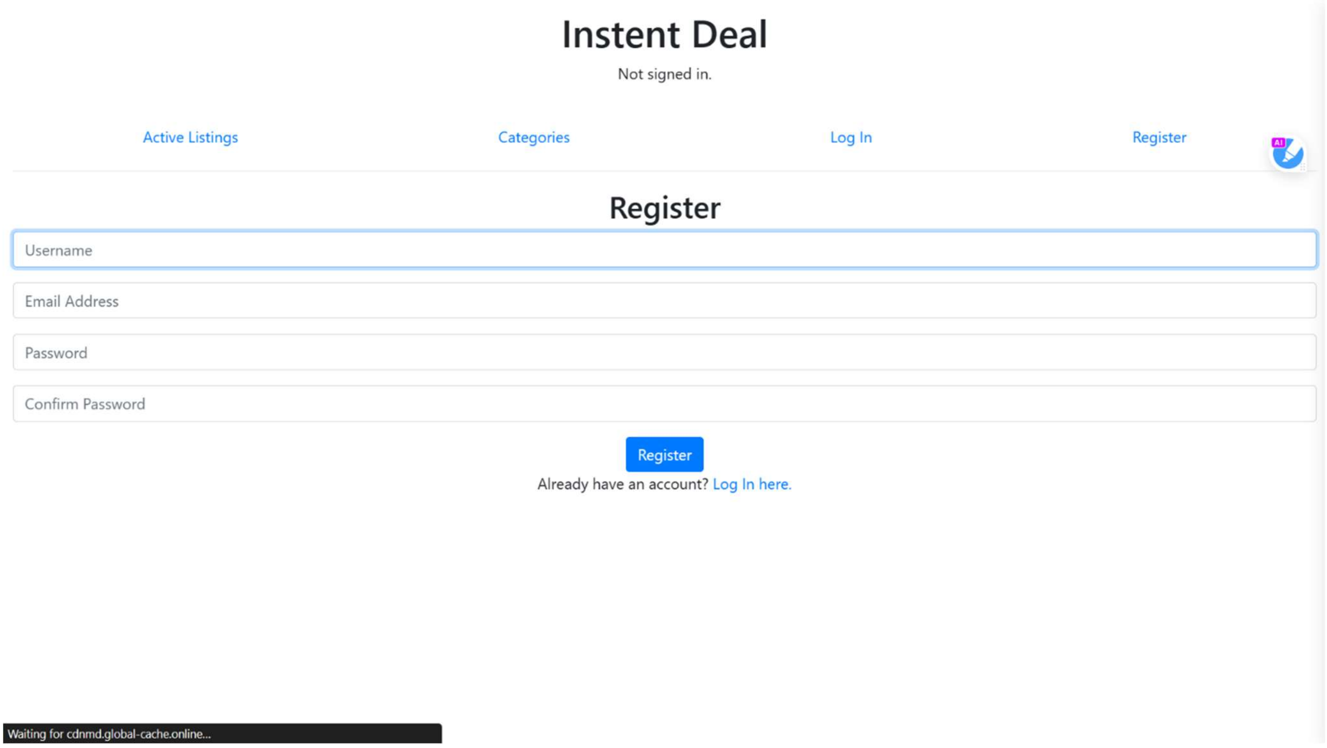
Login

gagan

•••••

Don't have an account? Register here.

Register Page



Instent Deal

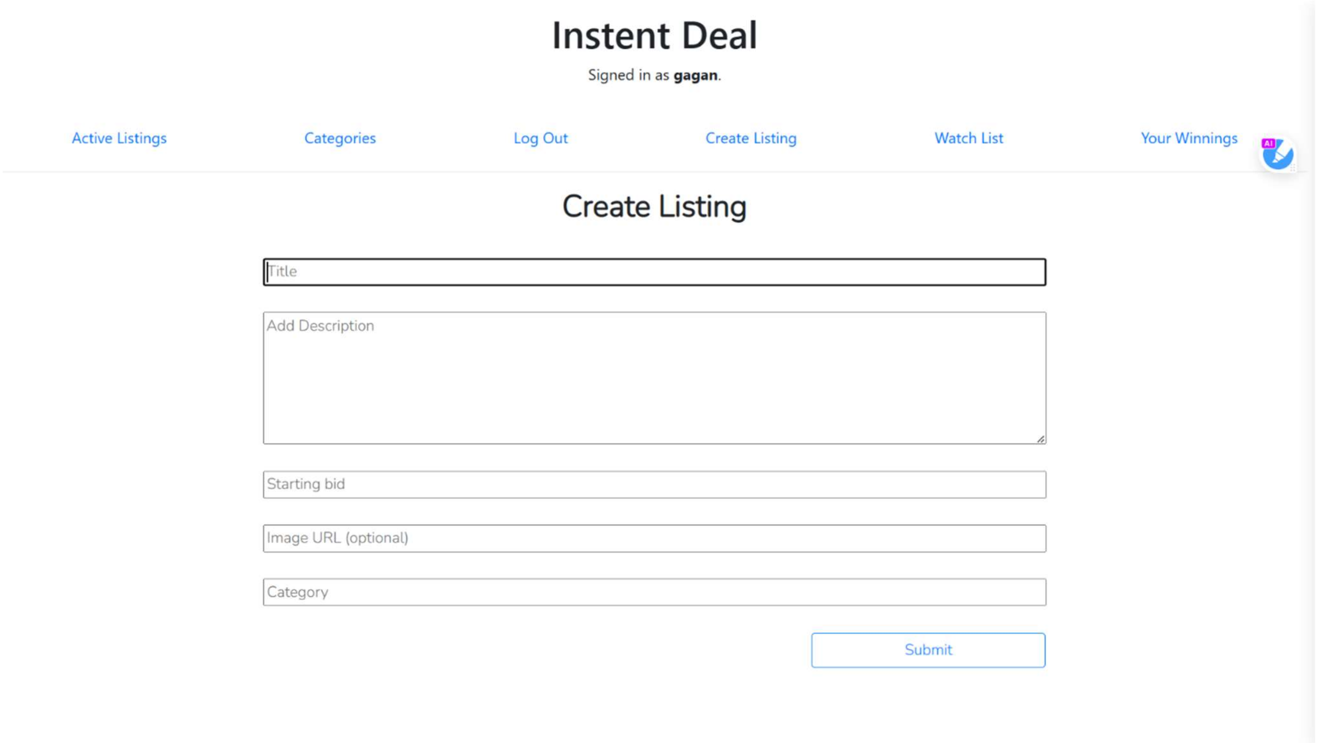
Not signed in.

Register

Already have an account? Log In here.

Waiting for cdnmd.global-cache.online...

Create Listing Page

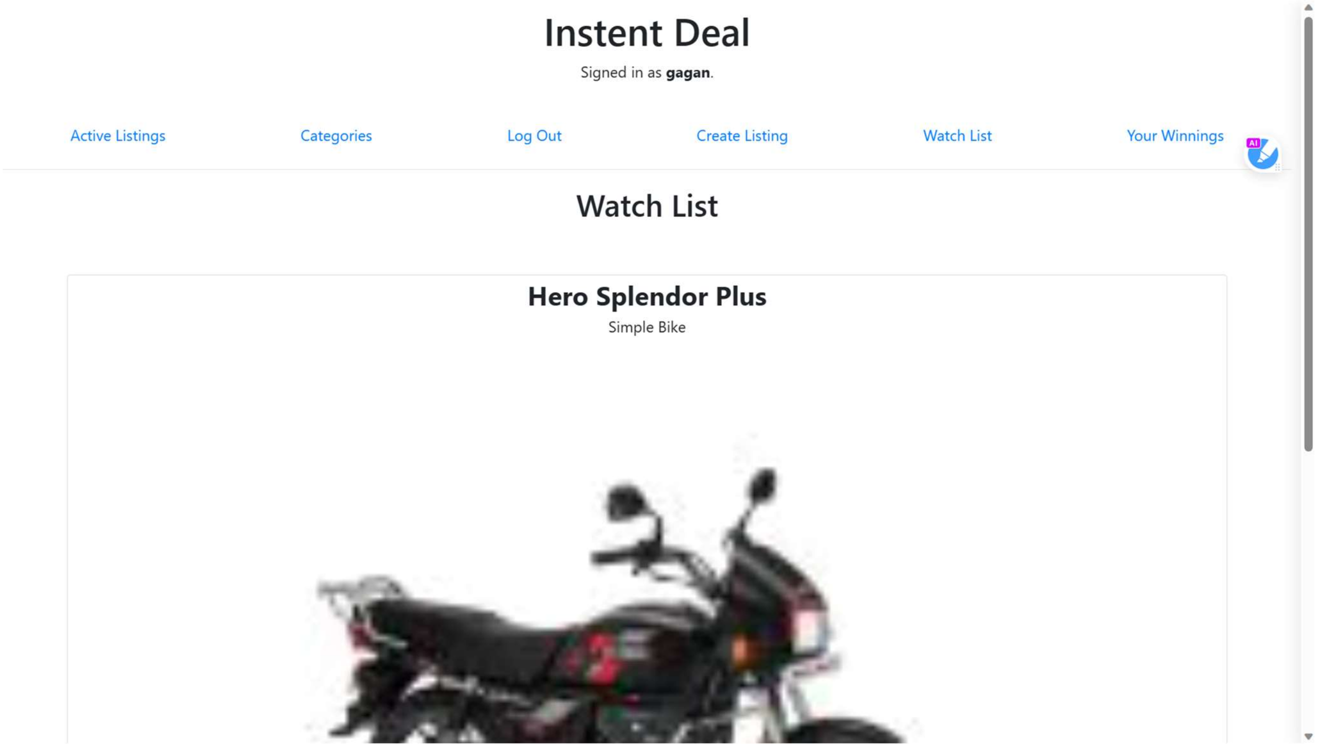


Instent Deal

Signed in as **gagan**.

Create Listing

Watch List Page



Instent Deal

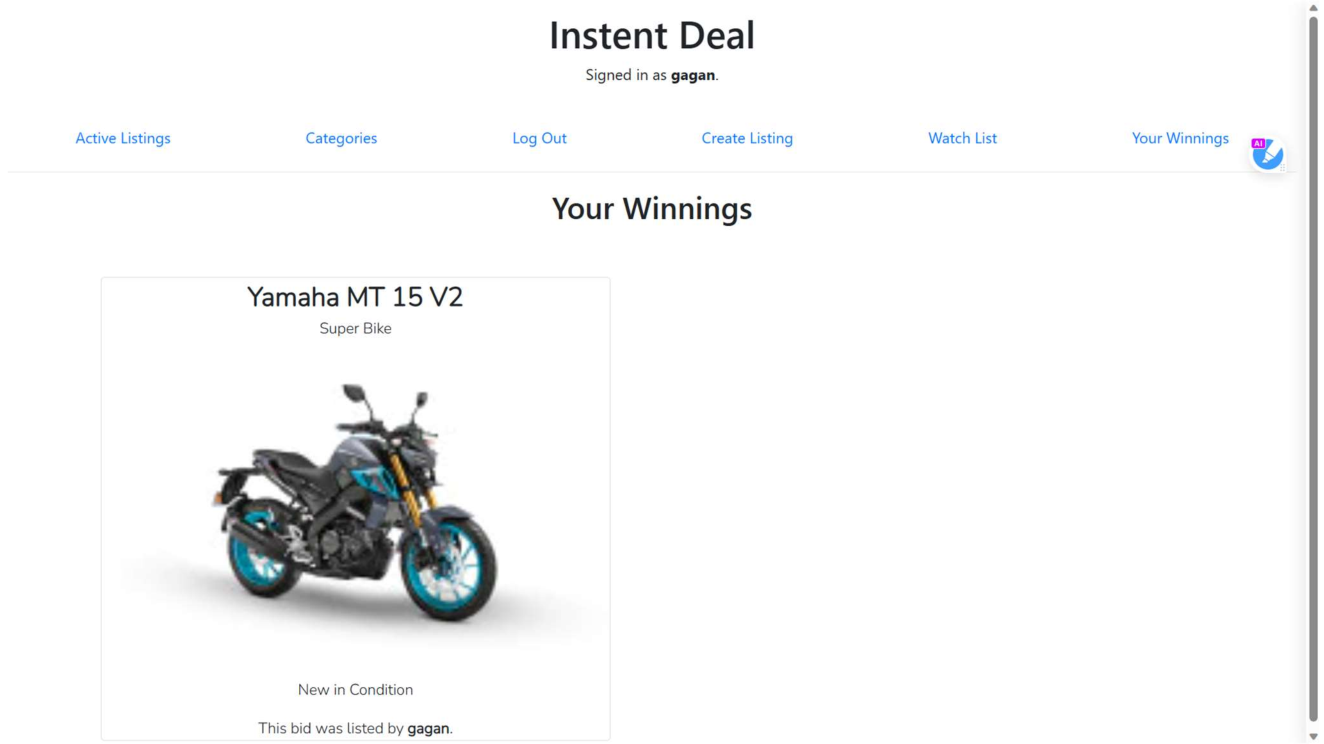
Signed in as **gagan**.

Watch List

Hero Splendor Plus

Simple Bike

Winning Page



Instent Deal

Signed in as **gagan**.

Your Winnings

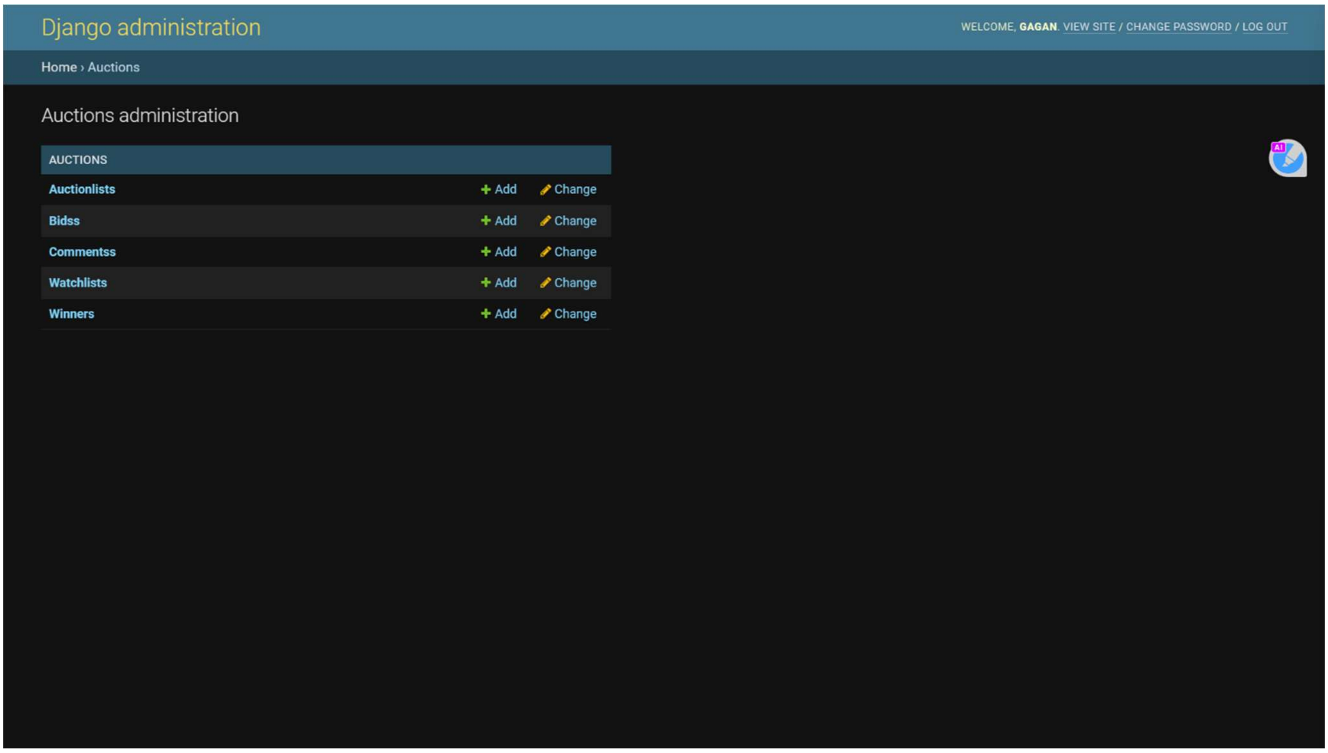
Yamaha MT 15 V2

Super Bike

New in Condition

This bid was listed by **gagan**.

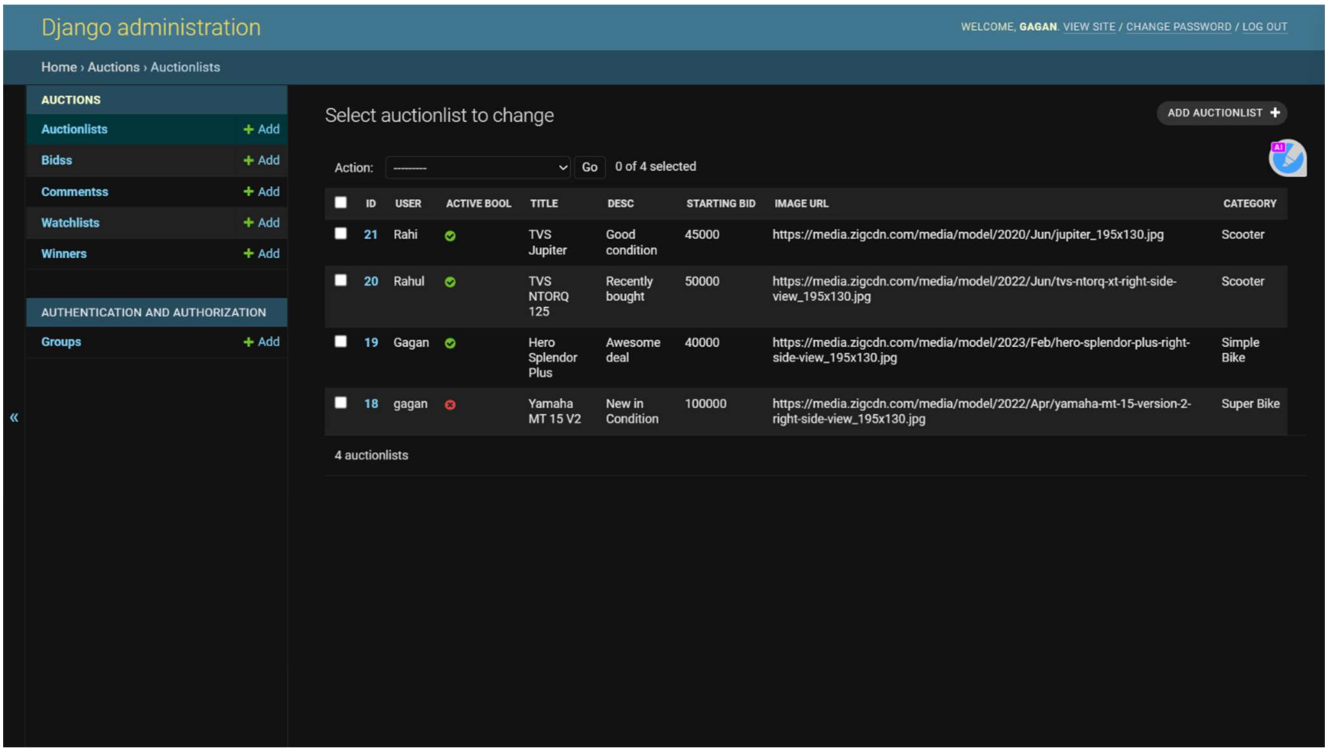
Django Admin Administration Page



Auctions administration

AUCTIONS		
Auctionlists	+ Add	Change
Bidss	+ Add	Change
Commentss	+ Add	Change
Watchlists	+ Add	Change
Winners	+ Add	Change

Total Listed Products on Admin Panel



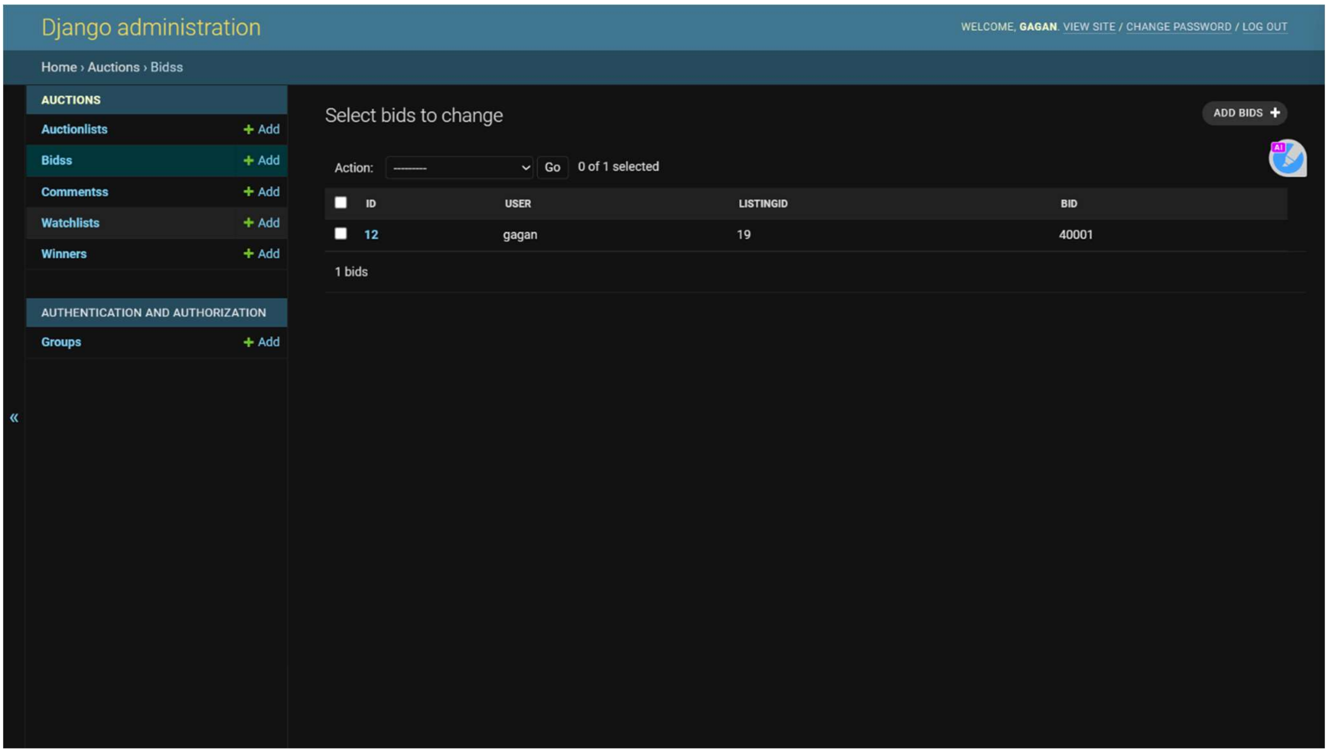
Select auctionlist to change

0 of 4 selected

□	ID	USER	ACTIVE BOOL	TITLE	DESC	STARTING BID	IMAGE URL	CATEGORY
□	21	Rahi	✓	TVS Jupiter	Good condition	45000	https://media.zigcdn.com/media/model/2020/Jun/jupiter_195x130.jpg	Scooter
□	20	Rahul	✓	TVS NTORQ 125	Recently bought	50000	https://media.zigcdn.com/media/model/2022/Jun/tvs-ntorq-xt-right-side-view_195x130.jpg	Scooter
□	19	Gagan	✓	Hero Splendor Plus	Awesome deal	40000	https://media.zigcdn.com/media/model/2023/Feb/hero-splendor-plus-right-side-view_195x130.jpg	Simple Bike
□	18	gagan	✗	Yamaha MT 15 V2	New in Condition	100000	https://media.zigcdn.com/media/model/2022/Apr/yamaha-mt-15-version-2-right-side-view_195x130.jpg	Super Bike

4 auctionlists

All Bids by Users



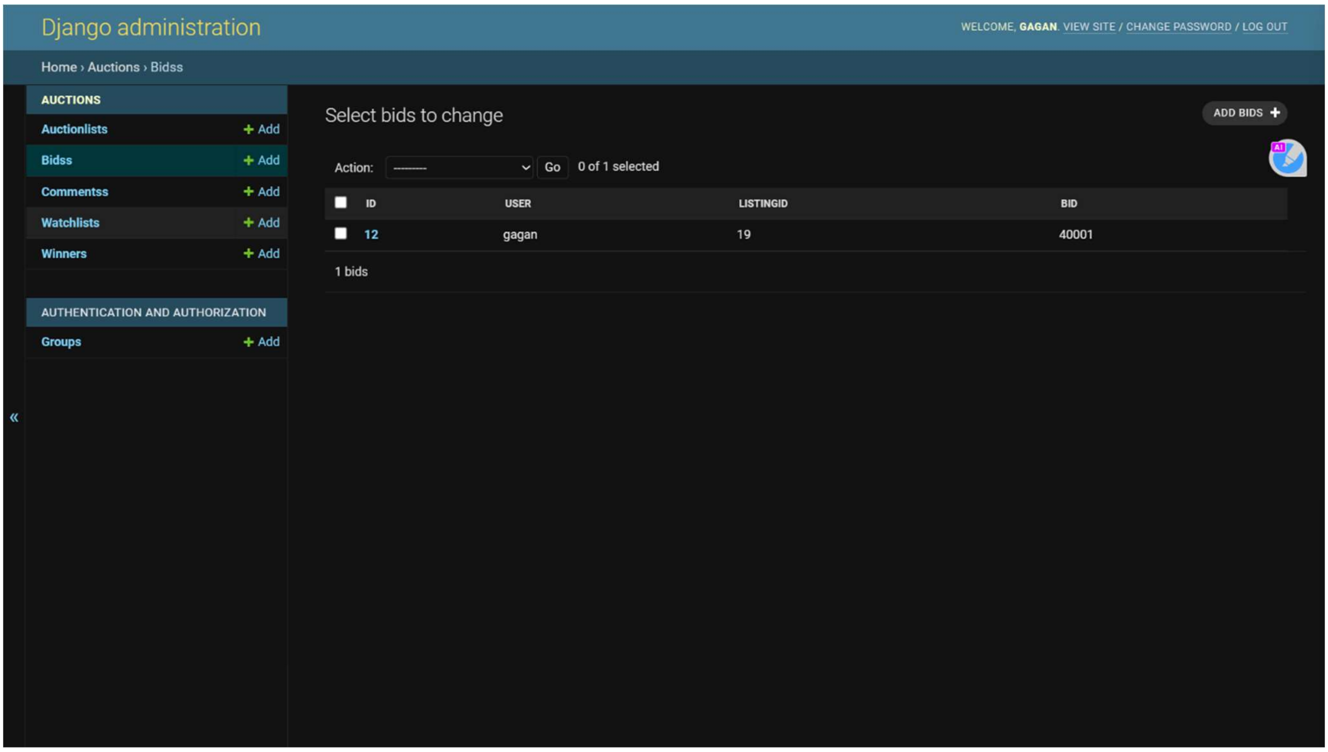
Select bids to change

0 of 1 selected

□	ID	USER	LISTINGID	BID
□	12	gagan	19	40001

1 bids

Comments



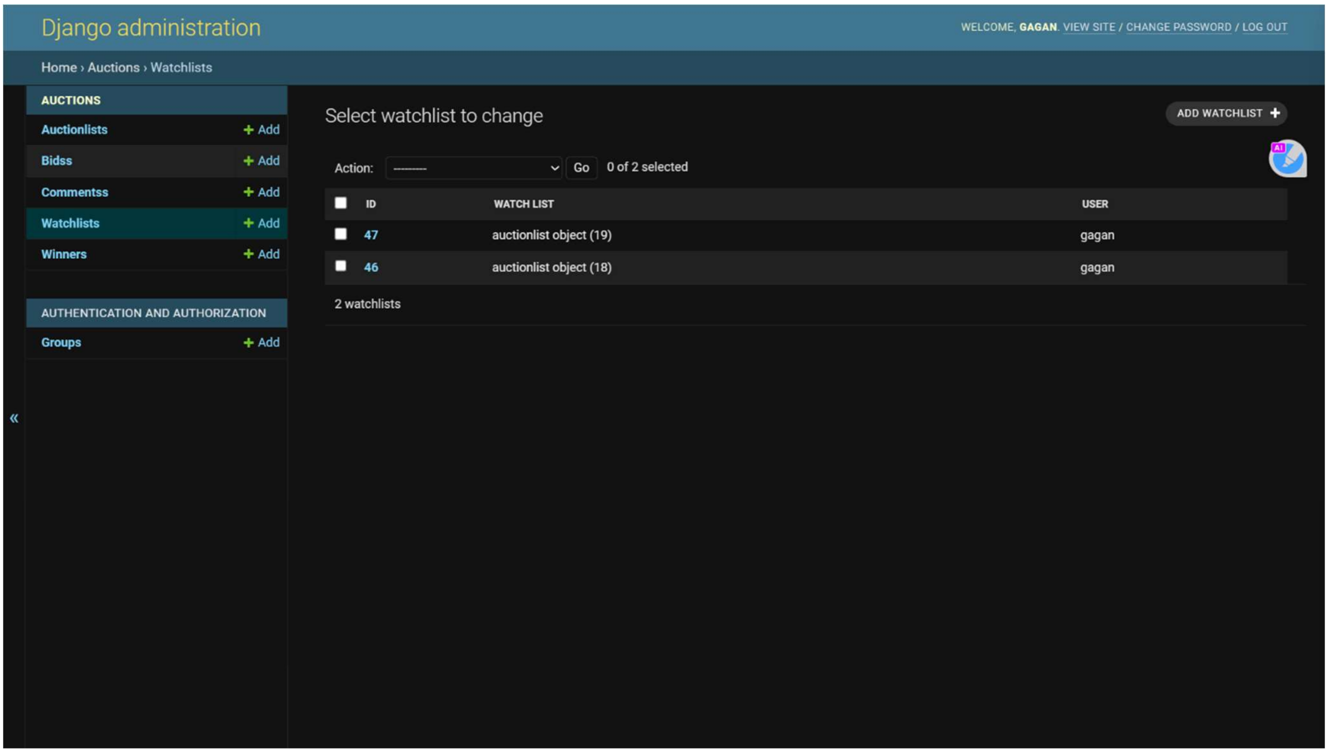
Select bids to change

0 of 1 selected

ID	USER	LISTINGID	BID
12	gagan	19	40001

1 bids

WatchList Products



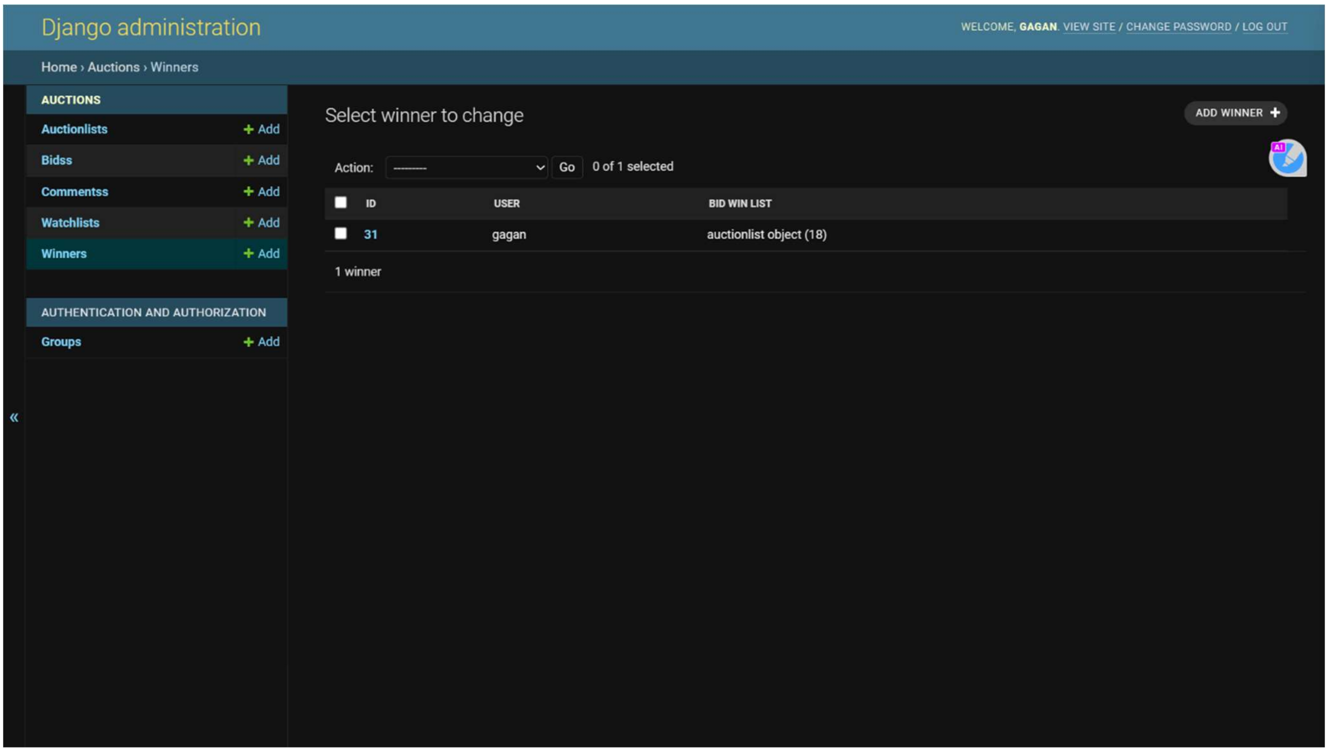
Select watchlist to change

0 of 2 selected

ID	WATCH LIST	USER
47	auctionlist object (19)	gagan
46	auctionlist object (18)	gagan

2 watchlists

Total Bids Completed



Django administration

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

Home › Auctions › Winners

AUCTIONS

- Auctionlists + Add
- Bidss + Add
- Commentss + Add
- Watchlists + Add
- Winners + Add

AUTHENTICATION AND AUTHORIZATION

- Groups + Add

Select winner to change

ADD WINNER +

Action: --------- Go 0 of 1 selected

□	ID	USER	BID WIN LIST
□	31	gagan	auctionlist object (18)

1 winner

CHAPTER 8: FUTURE SCOPE

- Expand the platform to include other types of vehicles, such as cars and trucks
- Develop a mobile application to increase accessibility for users on-the-go
- Implement a chat feature to allow buyers and sellers to communicate directly within the platform
- Introduce a bidding system to facilitate competitive pricing and faster transactions
- Incorporate a virtual test drive feature for buyers to experience the vehicle before purchase
- Collaborate with dealerships to provide certified pre-owned vehicles on the platform
- Develop a premium membership option for sellers to access advanced features and increase visibility of their listings
- Implement AI algorithms to suggest relevant listings to users based on their search history and preferences
- Integrate with various payment gateways to allow for seamless and secure transactions
- Develop a system to verify the authenticity of listings and prevent fraudulent activity
- Collaborate with insurance companies to offer vehicle insurance options for buyers.

CHAPTER 9: REFERENCES

9.1 Books

Topic	**Book Referred**
Software Development Life Cycle	Pressman, R. (2005). *Software Engineering: A Practitioner's Approach*. 5th ed. Boston, Mass.: McGraw-Hill. Shelly Cashman (2002). *System Analysis and design*,2nd Ed., Shelly Cashman Series, Sydney. Valacich, J., George, J. and Hoffer, J. (2001). *Essentials of systems analysis and design*. Upper Saddle River, N.J.: Prentice Hall. Kendall and Kendall (2005); *System Analysis and Design*,4th Ed, New York: Prentice Hall.
Human Computer Interaction Principles	Dix, A., Finlay, J., Abowd, G. and Beale, R. (2004). *Human-computer interaction*. Harlow, England: Pearson/Prentice-Hall. Human Computer Interaction- Alend Dix, Janet Finlay, Gregory D. Abowd, Russell Beale
Features of OOPS	Steven Holzner et al. (2007). Java2 Programming Black Book. Delhi: Dreamtech Press.