A SUMMER INTERNSHIP REPORT ON

ONLINE AUCTION SYSTEM

submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

MARTHALA SATYA SUREKHA

Roll No O190849

Under the Supervision of

Mrs. N Madhavi Latha

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

ONGOLE CAMPUS

2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, AP

ONGOLE CAMPUS

CERTIFICATE

This is to certify that the Internship report entitled **PYTHON PROGRAMMING & DEVELOPMENT INTERNSHIP** being submitted by **MARTHALA SATYA SUREKHA (O190849)** in partial fulfilment of the requirements for the award of the Bachelor of Technologies in Computer Science and Engineering to Rajiv Gandhi University of Knowledge Technologies-A.P. Ongole Campus, is a record of Bonafide project work carried out under my supervision during the academic year 2023-24.

The report hasn't been submitted previously in part or in full to this or any other university or institution for the award of my degree.

Mrs. M. Madhavi Latha,
Assistant Professor,
Department of CSE,
RGUKT, ONGOLE.

Mr . N. Mallikarjuna,
Head Of the Department(I/C),
Department of CSE,
RGUKT, ONGOLE.

APPROVAL SHEET

This report entitled Python Programming & Development Internship submitted by **Marthala Satya Surekha(O190849)** is approved for the degree of Bachelor of Technology in Computer Science and Engineering

Examiner(s):

_____

_____

Supervisor(s):

_____

_____

_____

Chairman:

_____

Date :_____

Place :_____

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature

_____

Marthala Satya Surekha

O190849

Date: _____

ACKNOWLEDGEMENT

I Would like to express my sincere gratitude to **INNOVATE INTERN**, My project guide **Mrs. N. MADHAVI LATHA**, **Assistant Professor** for her valuable suggestions and keen interest throughout progress of my course and research.

I express my sincere gratitude to **Mr. N MALLIKARJUNA, Head of the Department-Computer Science and Engineering,** Rajiv Gandhi University of Knowledge Technologies-A.P., Ongole Campus, for his suggestions, motivations and co-operation for the successful completion of the Work.

I extend my sincere thanks to **Mr. M RUPAS KUMAR, Dean of Academics**, RGUKT-AP, ONGOLE CAMPUS for his encouragement and constant help.

I extend my sincere thanks to **Dr. BHASKAR PATEL** Director, RGUKT-AP, ONGOLE CAMPUS for his encouragement.

I have taken efforts in this project. However, it could not have been possible without the kind support and help of many individuals and RGUKT. We would like to extend my sincere thanks to all of them.

My thanks and appreciation also go to my colleagues in developing the project and people who have willingly helped me out with their abilities.

With Sincere Regards,

Marthala Satya Surekha(0190849)

Date: _____

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, AP

ONGOLE CAMPUS

2024

INTERNSHIP COMPLETION CERTIFICATE

INNOVATE

Internship Completion Certificate

This is to certify that

MARTHALA SATYA SUREKHA

[AICTE STUDENT ID: **STU663732F78A0801714893559**]

COMPUTER SCIENCE AND ENGINEERING,

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES DR A P J ABDUL KALAM IIIT ONGOLE

has effectively completed a 6-week internship in **Python Programming & Development**, spanning from **11-May-2024** to **22-Jun-2024**. Additionally, the intern has satisfactorily completed and submitted a project titled "**Secure Online Auction System: Implementing Robust Cybersecurity Measures with Python**". We extend our best wishes for the future endeavors.

AICTE

MAKE IN INDIA

Certificate No:
INNOVATE/ppd/000030

Aravindhan D.
Executive Director
Innovate Intern.

www.InnovateIntern.com | hello@innovateintern.com | (+91) 970-970-3885 | CORPORATE612ee3672666d21638662823

ABSTRACT

This internship report outlines the successful completion of the project, The "Online Auction System using Python" aims to develop a secure platform for conducting online auctions. Leveraging the versatility of Python, this project integrates with the python frame work Django and ensure the integrity of auction processes. Through rigorous testing and validation, the system aims to provide a trustworthy environment for buyers and sellers to engage in transactions with confidence, mitigating risks associated with unauthorized access.

This project not only highlight technical skills but also actively enhance user experience and boost productivity.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Under this Internship of Python Programming and Development I assigned with a project titled **"Online Auction System"** using python and Django frameworks. The Online Bidding System is a web-based application developed using the Django framework with Python, alongside front-end technologies including HTML, CSS, JavaScript, and Bootstrap. This system facilitates online auctions, allowing users to bid on items in a streamlined and efficient manner. Our platform aims to provide a secure, user-friendly interface where users can easily participate in various auctions, place bids, and manage their auction activities.

1.1. OBJECTIVE OF THE PROJECT

The primary objectives of the Online Bidding System are:

- To provide a convenient platform for users to participate in online auctions.
- To ensure secure and transparent bidding processes.
- To offer real-time updates on bid status and auction activities.

CHAPTER 2
LITERATURE SURVEY

I have Studied Several Papers and articles based on my project “Online Auction System”.

1. Security in Online Auctions: A study by Wang, C. et al. (2015) explores the security challenges in online auction systems, including issues like bid shielding, sniping, and auction fraud. The study emphasizes the need for robust security measures to protect both buyers and sellers.

2. User Behavior in Online Auctions: Research by Bapna, R. et al. (2003) analyzes user behavior in online auctions, focusing on bidding strategies and the impact of auction design on user participation. The findings suggest that certain auction formats and rules can significantly influence bidding behavior and auction outcomes.

3. Real-Time Bidding Systems: A paper by Zhang, H. et al. (2016) discusses the architecture and performance of real-time bidding (RTB) systems, which are used in online advertising auctions. The concepts and technologies from RTB systems provide valuable insights into handling high-frequency bidding and real-time data processing**.**

CHAPTER 3
REQUIREMENT ANALYSIS

3.1. EXISTED SYSTEM

eBay is one of the most well-known and established online auction platforms. It allows users to buy and sell a wide variety of goods and services worldwide. Users can participate in auctions or purchase items at fixed prices.

3.2. PROPOSED SYSTEM

This System allows auction for ‘anyone, anywhere & anytime’ and It is open for both buyers and sellers. They both can come together and exchange their items.

3.3. REQUIREMENTS SPECIFICATIONS

- Backend: Django (Python)
- Frontend: HTML, CSS, JavaScript, Bootstrap
- Database: SQLite
- Version Control: Git

3.3.1. PURPOSE

The purpose of the Online Bidding System is to revolutionize the auction experience by leveraging the power of the internet and modern web technologies. By providing a secure, efficient, and user-friendly platform, the project aims to make online auctions more accessible and attractive to a broad audience, thereby enhancing the overall auction experience for all participants.

3.3.2. SCOPE

The scope of the Online Bidding System encompasses the essential features and functionalities required to conduct secure and efficient online auctions. By focusing on usability, performance, and security, the project aims to deliver a robust platform that meets the needs of both users and administrators. Future enhancements will further expand the system's capabilities and improve the overall user experience**.**

3.3.3. OVERALL DESCRIPTION

The development of our Online Bidding System is influenced by the successes and challenges of existing platforms like eBay and Amazon Auctions, as well as research studies on security and user behavior. Leveraging modern technologies like Django, Bootstrap allows us to create a robust, secure, and user-friendly auction platform.

CHAPTER 4
SYSTEM DESIGN

4.1 UML DIAGRAMS

Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualize the way system has been designed. It is quite similar to blueprints used in other fields of engineering. UML is not a programming language, it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system.

STRUCTURAL UML DIAGRAMS:

Structural diagrams depict a static view of a structure of a system. It is widely used in the Documentation of software architecture. The Structural UML Diagrams involves 7 diagrams They are:

Class Diagram
Object Diagram

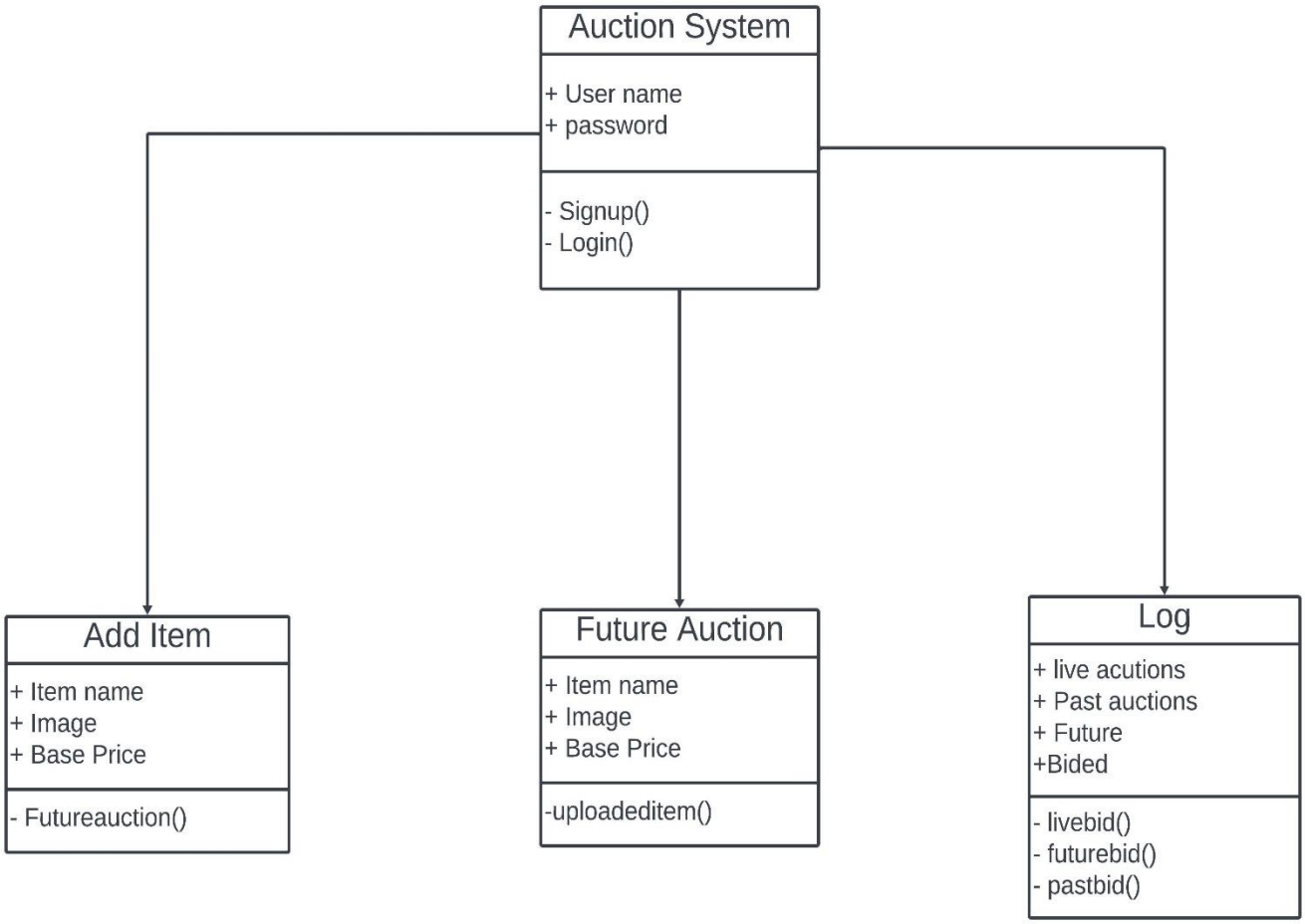
BEHAVIORAL UML DIAGRAMS:

Behavioral diagrams portray a dynamic view of a system or the behavior of a system which describes the functioning of the system. It involves 7 diagrams They are:

Use case Diagram
Activity Diagram

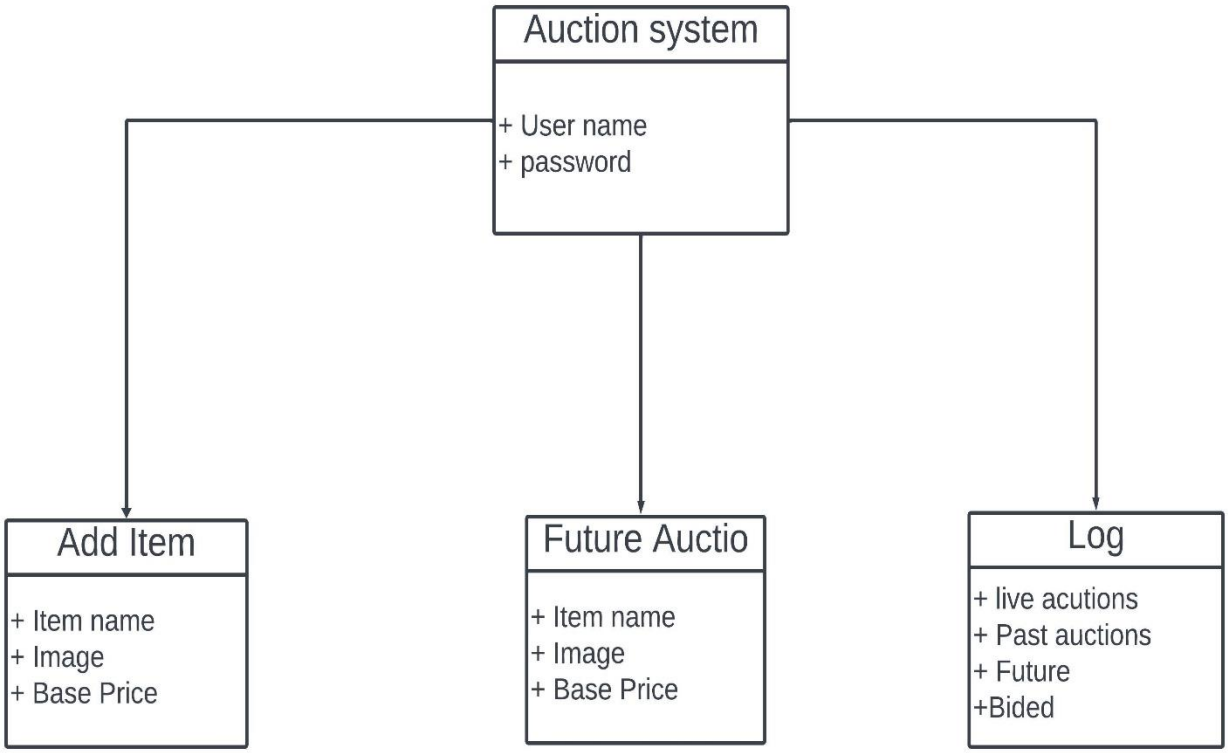
CLASS DIAGRAM

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modelling.



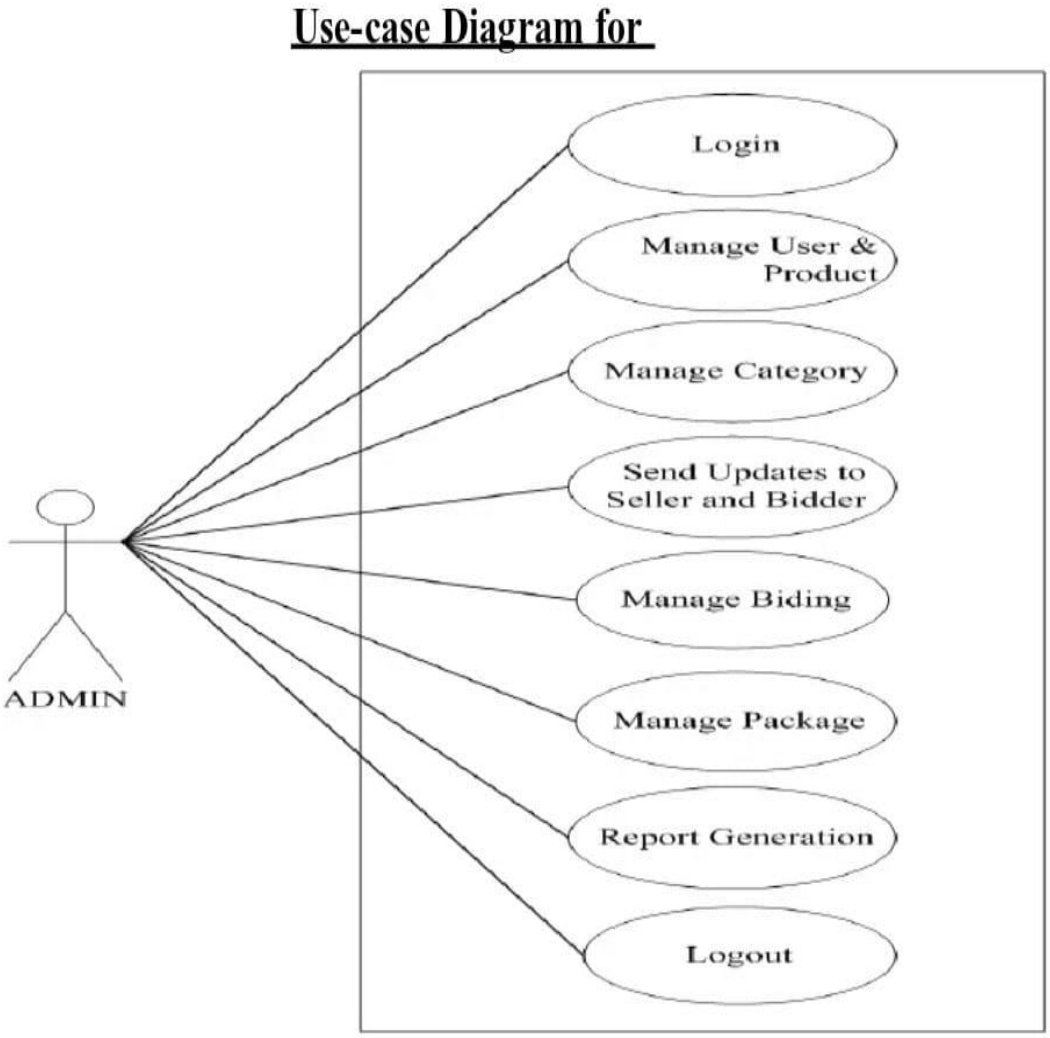
OBJECT DIAGRAM

It describes the static structure of a system at a particular point in time. It can be used to test the accuracy of class diagrams. It represents distinct instances of classes and the relationship between them at a time.



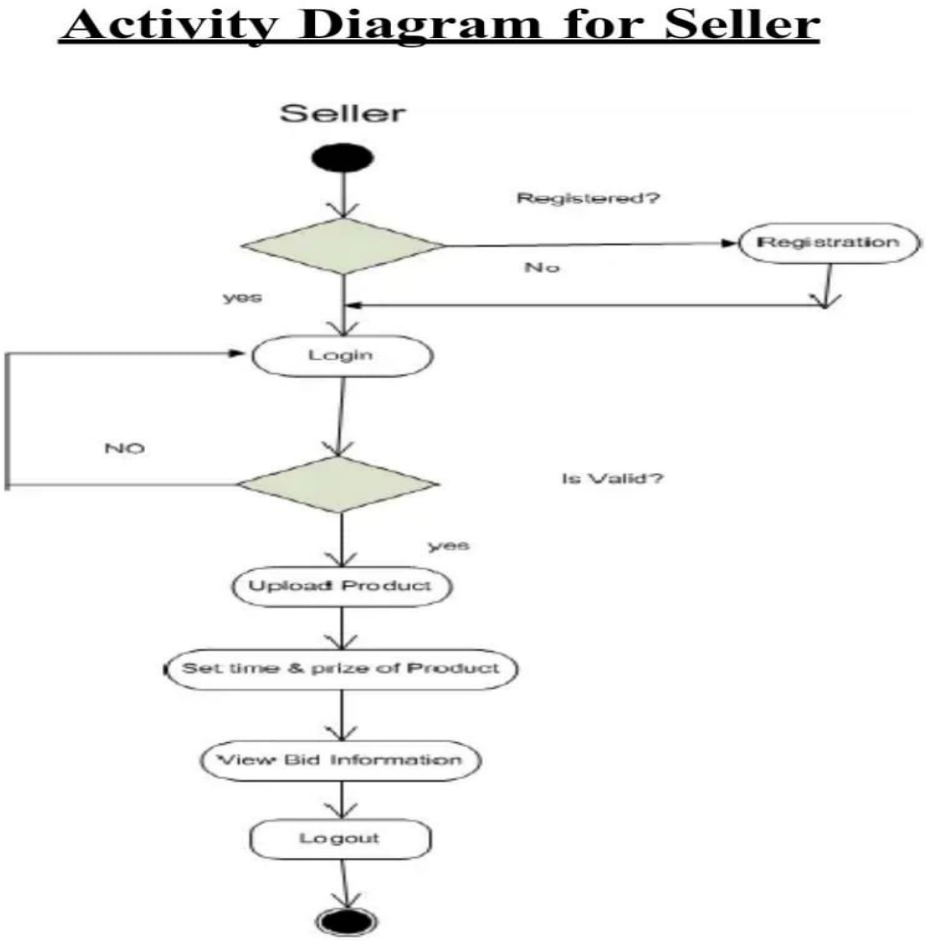
USE CASE DIAGRAM

Use case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.



ACTIVITY DIAGRAM

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent .



CHAPTER 5
IMPLEMENTATION

5.1. MODULES

LOGIN & REGISTER

HOME

ADD ITEM

FUTURE AUCTION

LOG

MY PROFILE

LOGOUT

5.2. MODULE DESCRIPTION

LOGIN & REGISTER

The new user can register by providing their details where as the existing users can log in through their details

HOME

The user can view and bid the auction items from this module. They can Navigate to different modules through this modules.

ADD ITEM

Through this module the bidder adds the items that are to be bid by specifying the item name description, base price, auction date and many more

FUTURE AUCTION

The future auction details will be displayed in this module so that the bidder can have idea of future biddings

LOG

The log module consists of information like your items in past, your items in live, your items in future, items bided by you

MY PROFILE

My Profile module consists of the things of logined user and the details like user first name, last name, email id

LOGOUT

The logout module is to get back to landing page.

5.3. INTRODUCTION OF TECHNOLOGIES USED

Backend: Django (Python)

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It follows the model-template-views (MTV) architectural pattern and provides a range of built-in features such as authentication, routing, and an ORM for database interaction. Django is designed to help developers take applications from concept to completion as quickly as possible.

Frontend: HTML, CSS, JavaScript, Bootstrap

HTML (Hypertext Markup Language) is the standard language for creating and structuring content on the web. It forms the skeleton of web pages by defining elements like headings, paragraphs, links, images, and more.

CSS (Cascading Style Sheets) is used to style and layout web pages. It allows developers to control the appearance of HTML elements, including their colors, fonts, positioning, and spacing, to create visually appealing and responsive designs.

JavaScript is a versatile, high-level programming language that enables interactive and dynamic content on web pages. It allows for the creation of features like form validation, animations, and asynchronous content updates, enhancing the user experience.

Bootstrap is a popular front-end framework for building responsive and mobile-first web pages. It provides a collection of pre-designed components and utilities, such as navigation bars, buttons, forms, and grids, which helps developers quickly create modern and consistent designs.

Database: SQLite

SQLite is a lightweight, serverless, self-contained SQL database engine. It is designed for simplicity and ease of use, making it ideal for small to medium-sized applications, development, and testing environments. SQLite stores the entire database in a single file and supports most SQL operations, making it a convenient choice for embedded database solutions.

5.4. SAMPLE CODE

Manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'bidding.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

settings.py

```
import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'vhzy_m5%i=mo*cq4+j$i5-2)kc_9s+%&t5s+dkz5ck_%m5dhhs'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'accounts',
    'items',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
```

```
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'bidding.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'bidding.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
```

```
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(r'C:\OnlineBiddingSystemPython-master', 'db.sqlite3'),
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'
MEDIA_URL='/media/'
MEDIA_ROOT = os.path.join(BASE_DIR,'media')

# EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
EMAIL_USE_TLS = True
EMAIL_USE_SSL =False
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_HOST_USER = 'spotsmartweb@gmail.com'
EMAIL_HOST_PASSWORD = "ives tsbs ulhp ykku"

STATICFILES_DIRS = [
   os.path.join(BASE_DIR,'static')
 ]

STATIC_ROOT = os.path.join(BASE_DIR,'assets')
```

Models.py

```
from django.db import models


# Create your models here.


class Detail(models.Model):
    username=models.CharField(max_length=50)
    contact = models.CharField(max_length=10)
```

views.py

```
from django.shortcuts import render,redirect
from django.http import HttpResponse
from django.contrib import messages
from django.contrib.auth.models import User,auth
from django.contrib.auth import authenticate
from django.contrib.auth.decorators import login_required
from items.models import Item
from .models import Detail
from django.core.mail import send_mail
from datetime import date
import datetime
# Create your views here.
def login(request):
    if request.method == 'POST':
        uname = request.POST.get('un','')
        pass1 = request.POST.get('pa','')
        user = auth.authenticate(username=uname,password=pass1)
```

```
        if user == None:
            messages.info(request,"invalid username/password")
            return redirect('login')
        else:
            auth.login(request,user)
            return redirect("home")

    else:
        return render(request,'login.html')

def register(request):
    if request.method == 'POST':
        fname=request.POST['fname']
        lname=request.POST['lname']
        name = request.POST['name']
        mail = request.POST['email']
        p1 = request.POST['p1']
        p2 = request.POST['p2']

        contact = request.POST['contact']
        if p1 == p2:
            if User.objects.filter(email=mail).exists():
                messages.info(request,"Already an User with this Email")
                return redirect('register')
            elif User.objects.filter(username=name).exists():
                messages.info(request,"Already an User with this Username")
                return redirect('register')
            else:
```

```
            user =
User.objects.create_user(first_name=fname,last_name=lname,email=mail,password=
p1,username=name)
            user.save()
            obj = Detail(username=name,contact=contact)
            obj.save()
            subject = "Online Bidding"
            msg     = "Congratulations you are registered successfully."
            to      = mail
            res     = send_mail(subject, msg, "spotsmartweb@gmail.com", [to])
            if res == 1:
               return redirect('/')
            else:
               messages.info(request,"Some thing is wrong")
               return redirect('register')
      else:
         messages.info(request,"Password does not match")
         return redirect('register')
   else:
      return render(request,'register.html')



@login_required(login_url='login')
def sendMailTowinners(request):
   today = date.today()
   yesterday = today - datetime.timedelta(days=1)
   item =
Item.objects.filter(start_date=yesterday).filter(sold="sold").filter(sendwinmail="unse
nded")
   for i in item :
```

```
        # print("1")
        try:
            # print("2")
    winnerid = i.highest_bidder
    # print(winnerid)
    user_obj = User.objects.get(id=winnerid)
    winnermail = user_obj.email

    winuser = user_obj.username
    # wincon=""
    #-----------------------------------------------------------
    obj = Detail.objects.get(username=winuser)
    wincon = obj.contact
    # print(wincon)

    itemmail = i.ownermail
    itemUserobj = User.objects.get(email=itemmail)
    itemuser = itemUserobj.username

    obj2 = Detail.objects.get(username=itemuser)
    itemcon = obj2.contact
    # print(itemcon)
    # print(winnermail)
    #-------------------------------------------------------------
    #To winner
    #send ownwer contact
    subject = "Online Bidding"
```

```
            msg     = "Congratulations you are winner of item"+i.name+"'s, Seller Email-id is "+i.ownermail+"  contact him for further informations. phone no = "+itemcon+" Thank You :)"
            to      = winnermail
            res     = send_mail(subject, msg, "spotsmartweb@gmail.com", [to])
            if res ==1:
               print ("mail sended to winner")
            else:
               print("something wrong for sending mail to winner")


            #To owner
            #send winner contact
            subject = "Online Bidding"
            msg     = "Congratulations your item "+i.name+"'s higgest bidder's email id is "+winnermail+" ,  contact him for further informations. phone no = "+wincon +" Thank You :)"
            to      = i.ownermail
            res     = send_mail(subject, msg, "spotsmartweb@gmail.com", [to])
            if res ==1:
               print ("mail sended to seller")
            else:
               print("something wrong for sending mail to seller")
            i.sendwinmail="sended"
            i.save()
        except:
            pass


@login_required(login_url='login')
def pastConfigurations(request):
```

```
    # cuser =request.user
    # cmail = cuser.email
    # item = Item.objects.filter(ownermail=cmail)
    item = Item.objects.all()
    for i in item:
        try:
            hb = i.highest_bidder
            if hb is not None:
                i.sold="sold"
                i.save()
            else:
                i.sold="unsold"
                i.save()
        except:
            pass
    # print("hy")


@login_required(login_url='login')
def home(request):
    items = Item.objects.all()
    today = date.today()
    yesterday = today - datetime.timedelta(days=1)
    # print(today)
    # print(yesterday)
    for i in items:
        # print (i.start_date)
        if(today > i.start_date):
            i.status = "past"
            # print("past")
```

```
        if(today < i.start_date):
            i.status="future"
            # print("future")
        if(today == i.start_date):
            i.status="live"
            # print("live")
        i.save()
        # print("-------")
    pastConfigurations(request)
    sendMailTowinners(request)
    items = Item.objects.filter(status="live")
    return render(request,"home.html",{'items':items})

def logout(request):
    auth.logout(request)
    return redirect("login")

def ilogout(request):
    auth.logout(request)
    return redirect("login")

@login_required(login_url='login')
def myprofile(request):
    bidder = request.user
    # item_obj = Item.objects.get(highest_bidder=bidder.id)
    details = bidder
    cuname = details.username
    # print(cuname)
    # ,"item_obj":item_obj
```

```
    obj = Detail.objects.filter(username=cuname)
    contact=""
    for i in obj:
        contact = i.contact
    return render(request,"myprofile.html",{"details":details,"contact":contact})


@login_required(login_url='login')
def log(request):
    cuser =request.user
    cmail = cuser.email
    cid = cuser.id
    item_obj = Item.objects.filter(highest_bidder=cid)


    biddeditem = item_obj
    # item = Item.objects.filter(ownermail=cmail)
    pitem = Item.objects.filter(ownermail=cmail).filter(status="past")
    litem = Item.objects.filter(ownermail=cmail).filter(status="live")
    fitem = Item.objects.filter(ownermail=cmail).filter(status="future")
    return
render(request,"log.html",{'pitem':pitem,'litem':litem,'fitem':fitem,"biddeditem":bidde
ditem})


@login_required(login_url='login')
def future(request):
    items = Item.objects.filter(status="future")
    return render(request,"future.html",{"items":items})
```

urls.py

```
from django.contrib import admin
from django.urls import path
from . import views
from django.contrib.auth import views as auth_views

from django.views.generic import TemplateView

urlpatterns = [
    path('',views.login,name="login"),
    path('register',views.register,name="register"),
    path('home',views.home,name="home"),
    path('logout',views.logout,name="logout"),
    path('items/logout',views.ilogout,name="ilogout"),
    path('myprofile',views.myprofile,name="myprofile"),
    path('future',views.future,name="future"),
    path('log',views.log,name="log"),

    # forgot
    path('password-reset/',
        auth_views.PasswordResetView.as_view(
            template_name='commons/password-reset/password_reset.html',
            subject_template_name='commons/password-
reset/password_reset_subject.txt',
            email_template_name='commons/password-
reset/password_reset_email.html',
            success_url='done/'
        ),
```

```
        name='password_reset'),
    path('password-reset/done/',
        auth_views.PasswordResetDoneView.as_view(
            template_name='commons/password-reset/password_reset_done.html'
        ),
        name='password_reset_done'),
    path('password-reset-confirm/<uidb64>/<token>/',
        auth_views.PasswordResetConfirmView.as_view(
            template_name='commons/password-reset/password_reset_confirm.html'
        ),
        name='password_reset_confirm'),
    path('password-reset-complete/',
        auth_views.PasswordResetCompleteView.as_view(
            template_name='commons/password-reset/password_reset_complete.html'
        ),
        name='password_reset_complete'),
]
```

Base.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Bid Mafia</title>
        <link rel="stylesheet" href="../static/css/nav.css">
        <link
href="https://fonts.googleapis.com/css?family=Audiowide&display=swap"
rel="stylesheet">
```

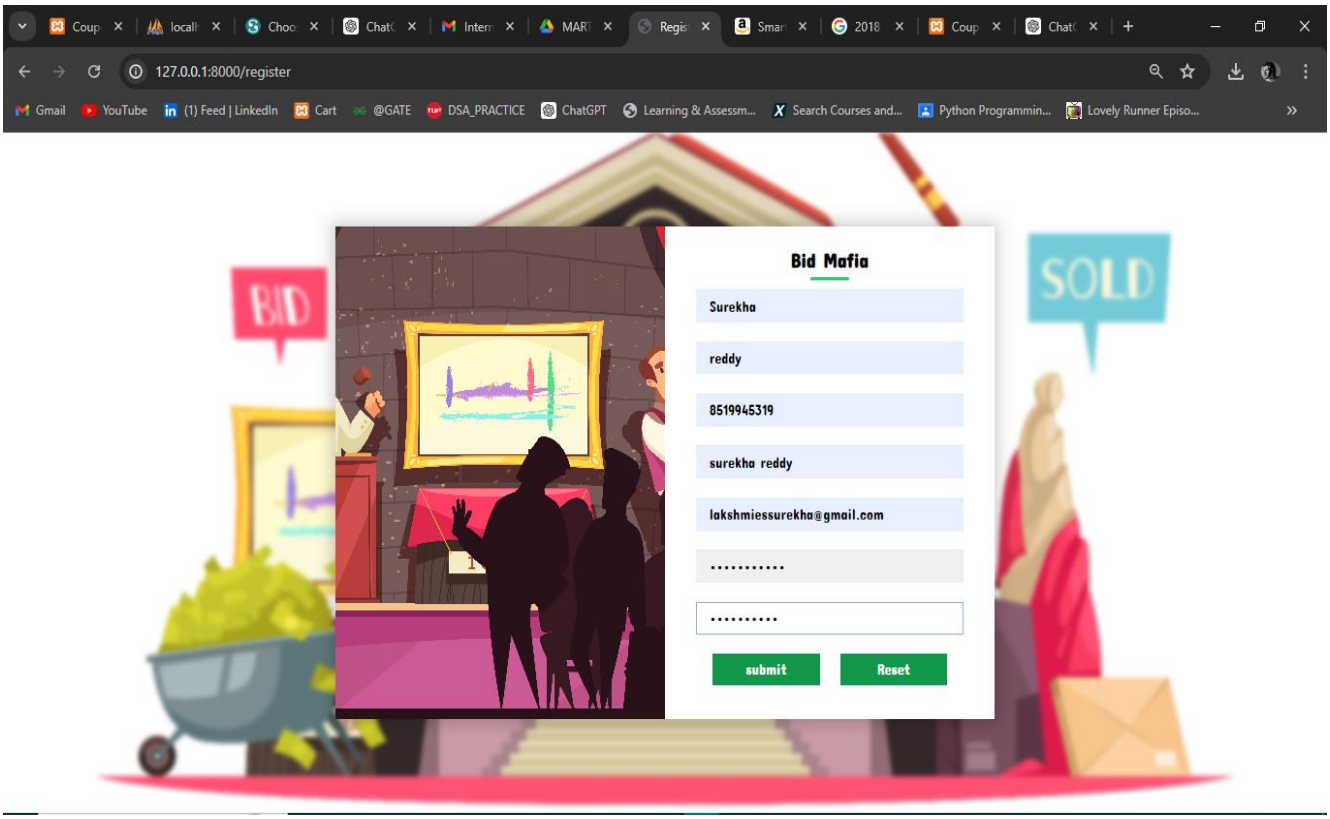
```
    <link
href="https://fonts.googleapis.com/css2?family=Roboto+Condensed&display=swap"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@500&display=s
wap" rel="stylesheet">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link href="https://fonts.googleapis.com/css?family=Open+Sans&display=swap">
  </head>
  <body>
    <p class="hhhh">
      Where the Wolrld goes to shop
      <span>
       bid mafia
      </span>
      &mdash; don't miss just bid &mdash;
     </p>
    <div class="khali">
    <div class="wrapper">
      <div class="ffuk">
        <ul>
          <li><a href="http://127.0.0.1:8000/home">Home</a></li>
          <li><a href="items/additem">Add Item</a></li>
          <li><a href="future">Future Auction</a></li>
          <li><a href="log">Log</a></li>
```

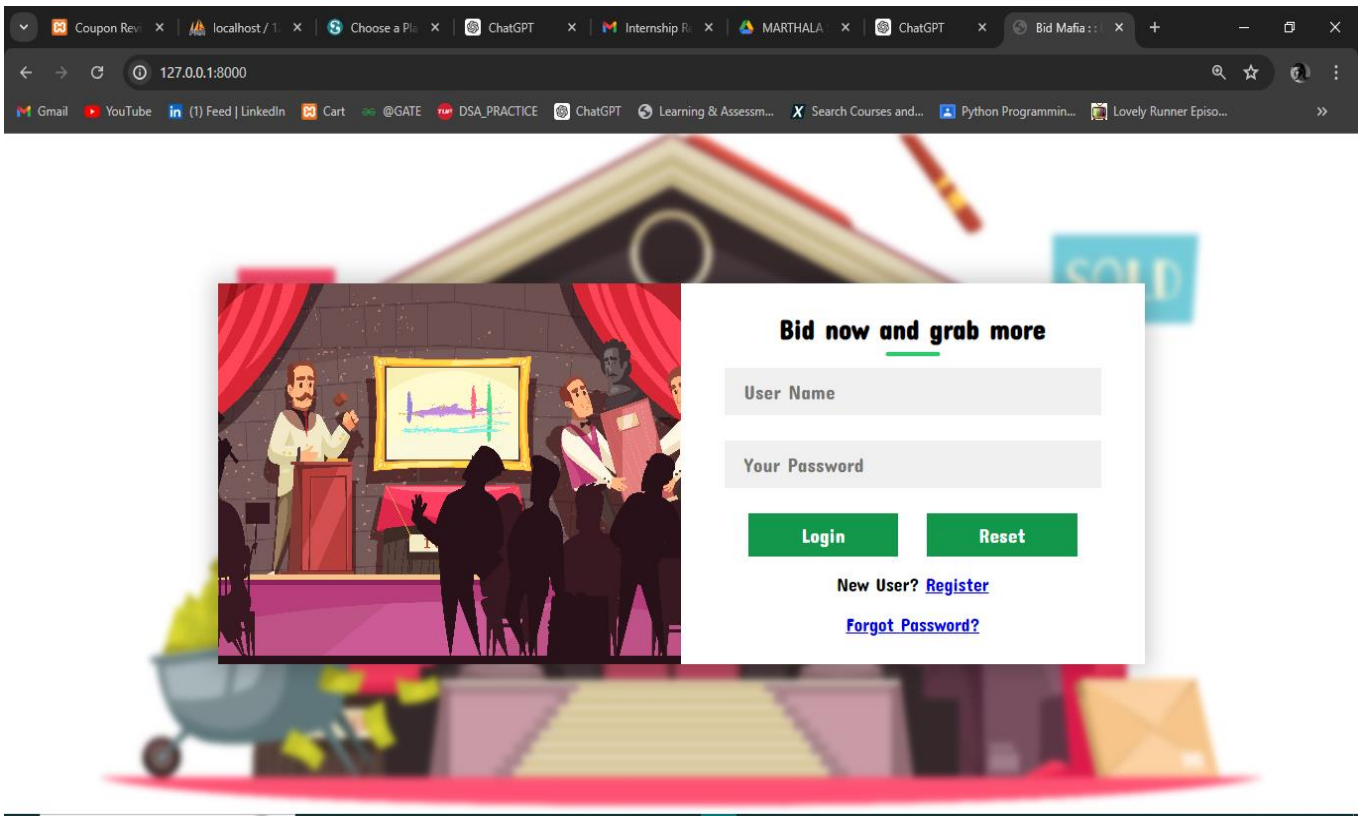
```
                <li><a href="myprofile">My Profile</a></li>
                <li><a href="logout">Logout</a></li>
                <!-- <li><a href="#"></a></li> -->
              </ul>
            </div>
        </div>
        </div>
        <div class="container">
            {% block content %}
            {% endblock %}
        </div>
    </body>
</html>
```

CHAPTER 6
TESTING

6.1. BLACK BOX TESTING

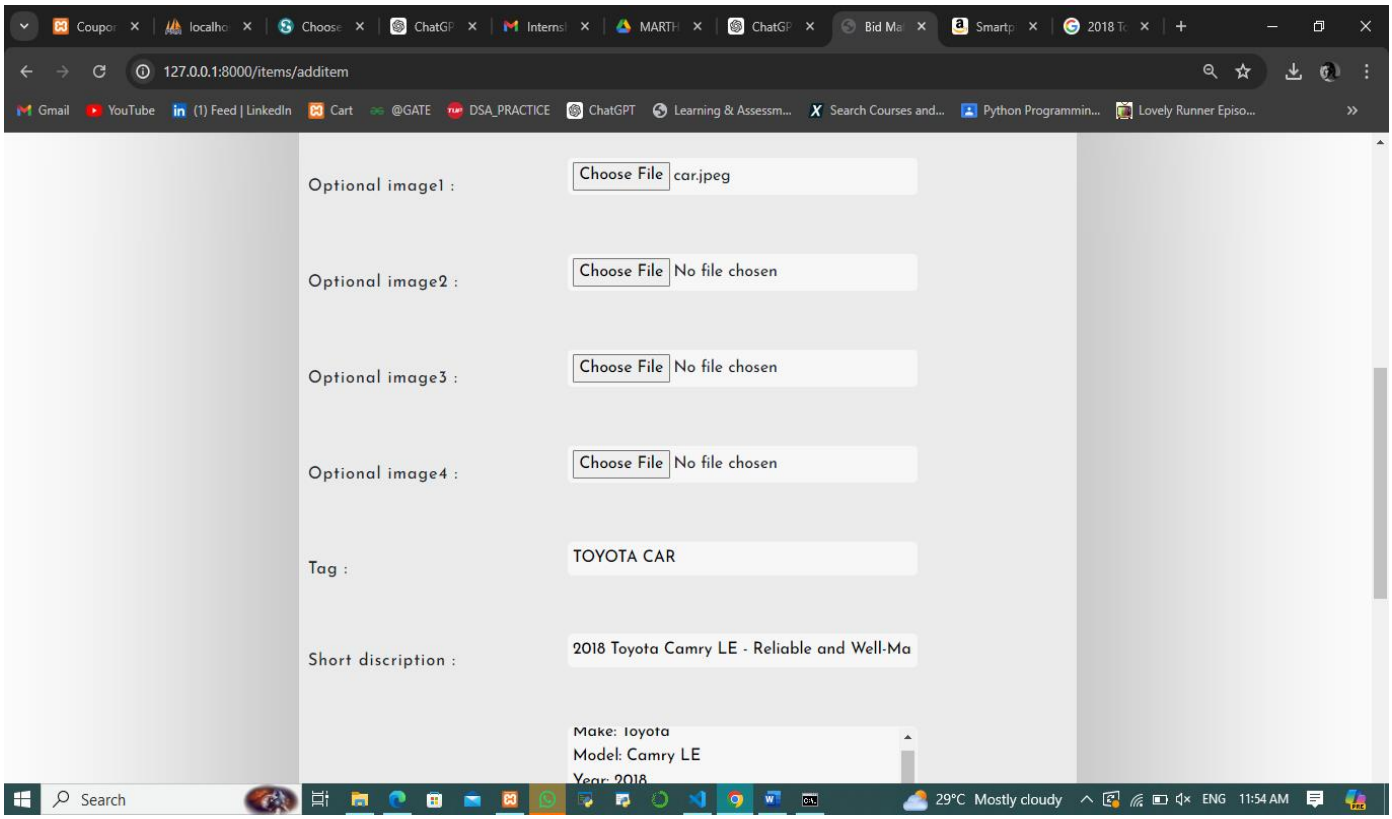
Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues. Black box testing is a powerful testing technique because it exercises a system end to end. Just like end users “Don’t care” how a system is coded or architected, and expect to receive an appropriate response to their requests, A tester can stimulate user activity and see if the system delivers on it’s promises. Along the way, a black test evaluates all relevant subsystems, including UI/UX, web server application server, database dependencies and integrated systems.

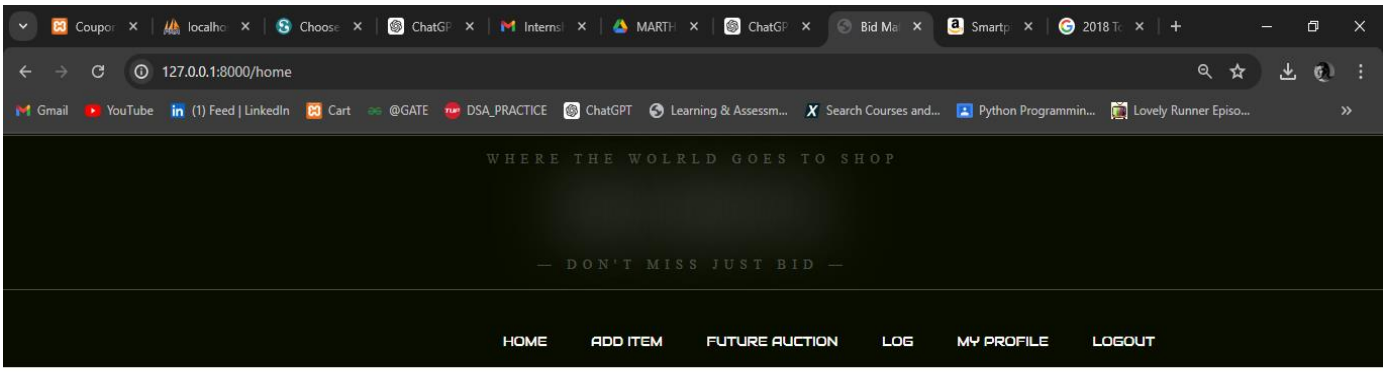




6.2. WHITE BOX TESTING

White box testing is an approach that allows testers to inspect and verify the innerworkings of a software system—its code, infrastructure, and integrations with external systems. White box testing is an essential part of automated build processes in a modern Continuous Integration/Continuous Delivery (CI/CD) development pipeline. White box testing is often referenced in the context of Static Application Security Testing (SAST), an approach that checks source code or binaries automatically and provides feedback on bugs and possible vulnerabilities.

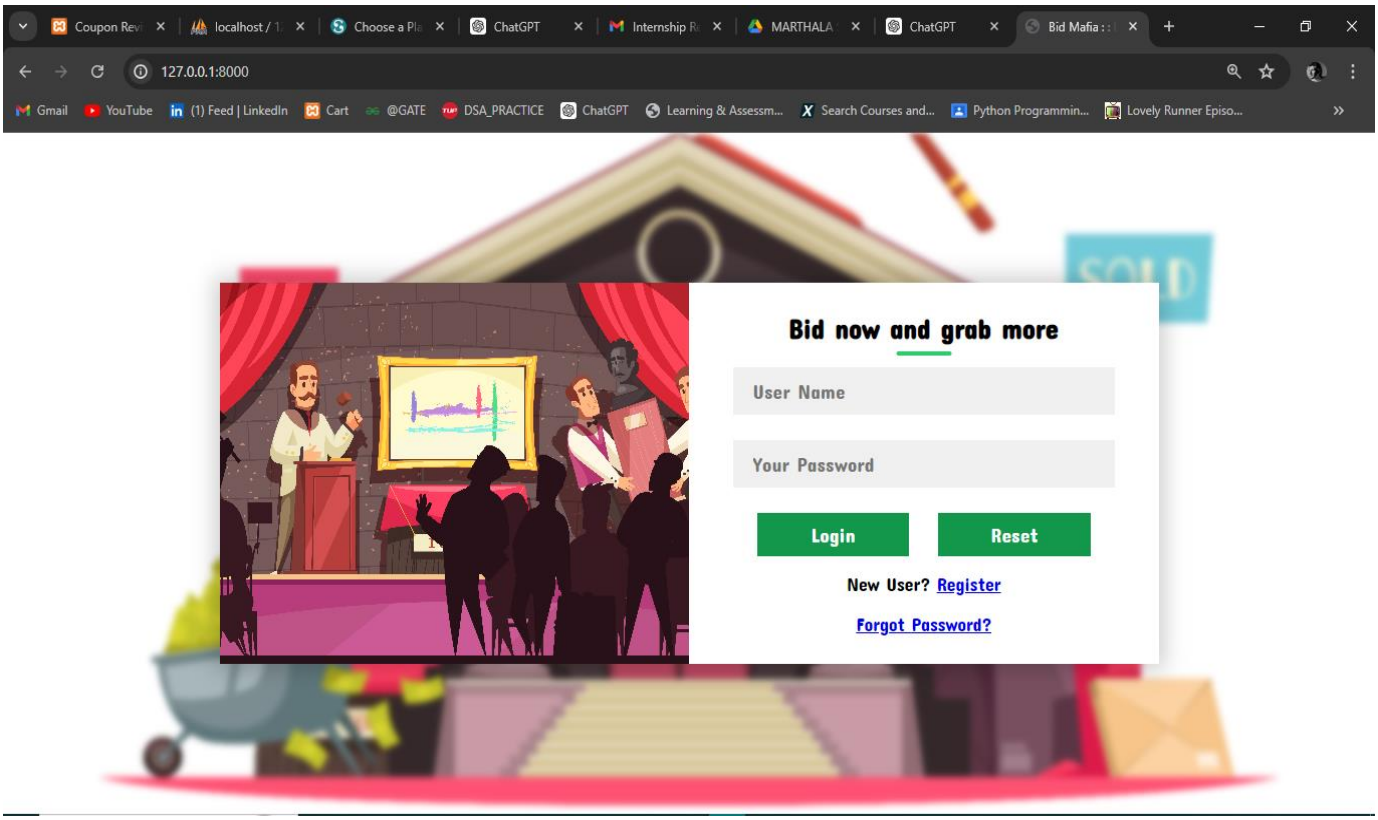

127.0.0.1:8000/items/additem
Optional image1 :
Choose File car.jpeg
Optional image2 :
Choose File No file chosen
Optional image3 :
Choose File No file chosen
Optional image4 :
Choose File No file chosen
Tag :
TOYOTA CAR
Short discription :
2018 Toyota Camry LE - Reliable and Well-Ma
Make: Toyota
Model: Camry LE
Year: 2018

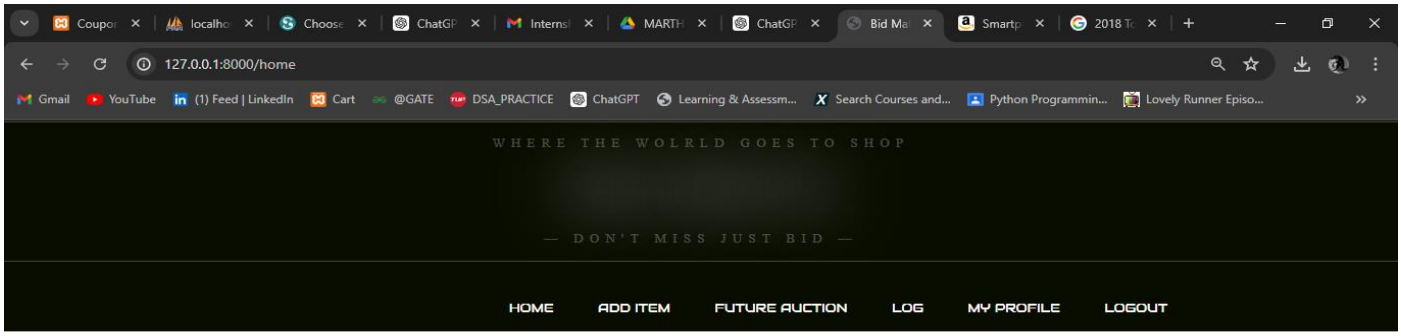

127.0.0.1:8000/home
WHERE THE WOLRLD GOES TO SHOP
— DON'T MISS JUST BID —
HOME
ADD ITEM
FUTURE AUCTION
LOG
MY PROFILE
LOGOUT

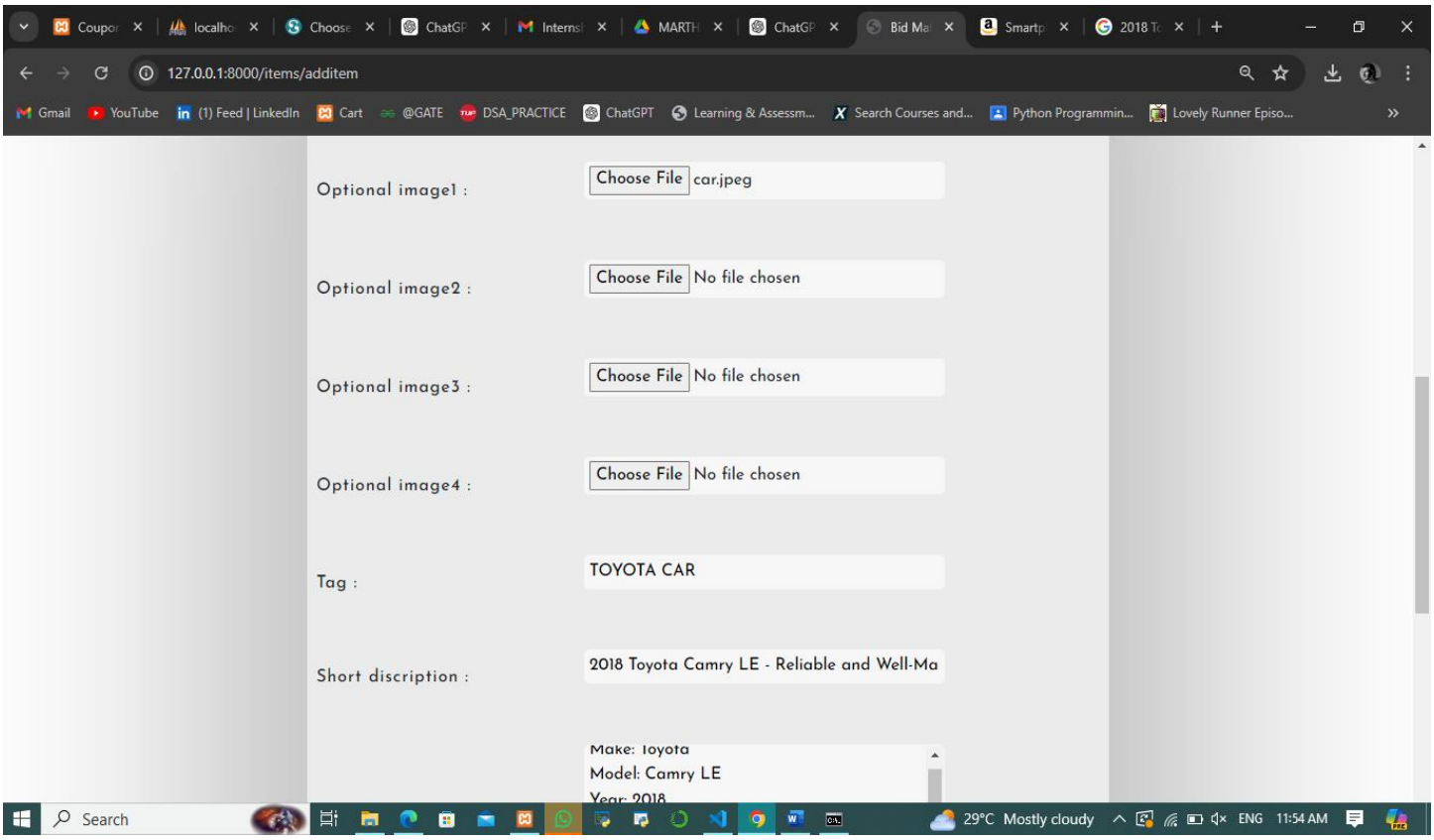

Current Auction
CAR
CURRENT-PRICE : 600000 ₹

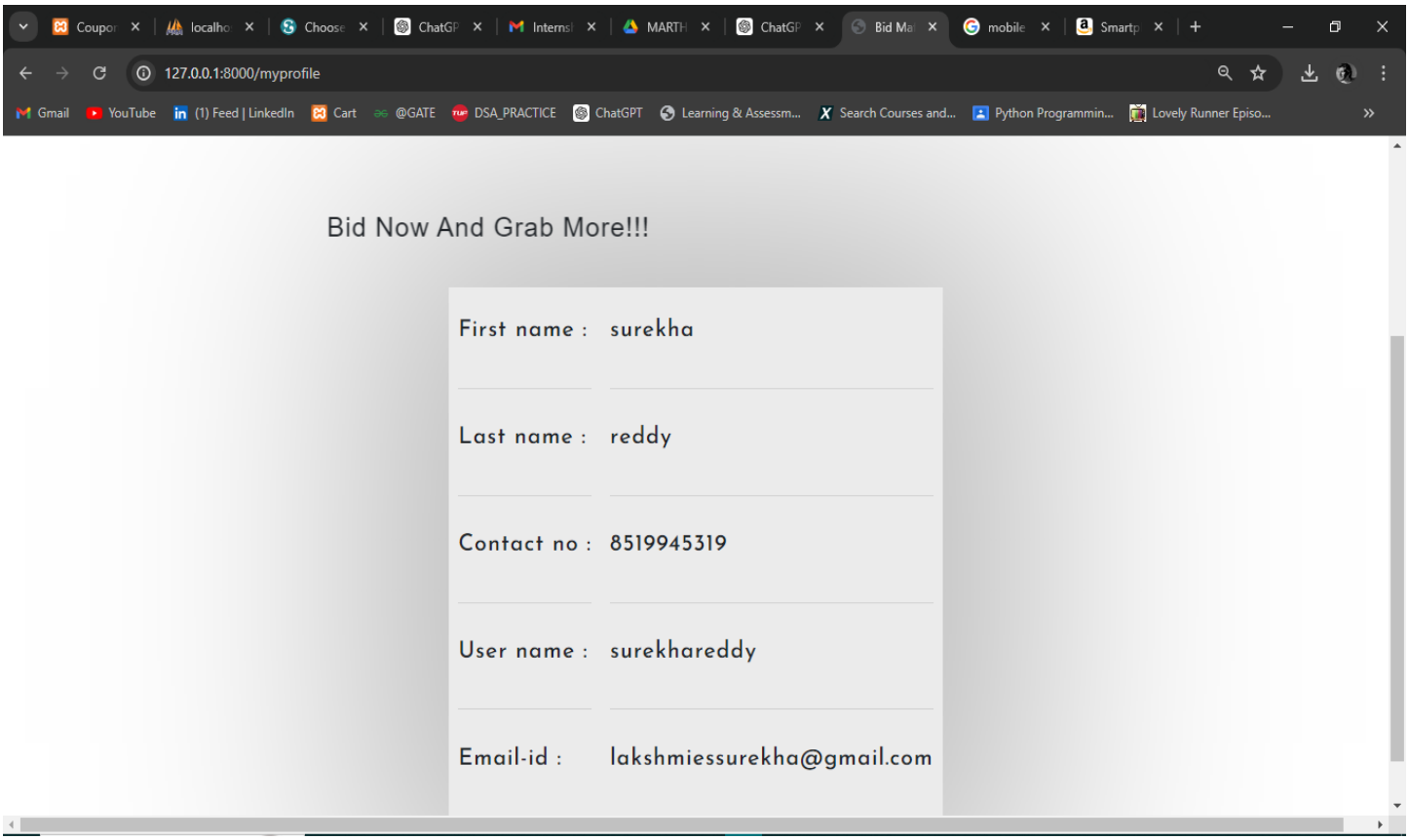
CHAPTER 7

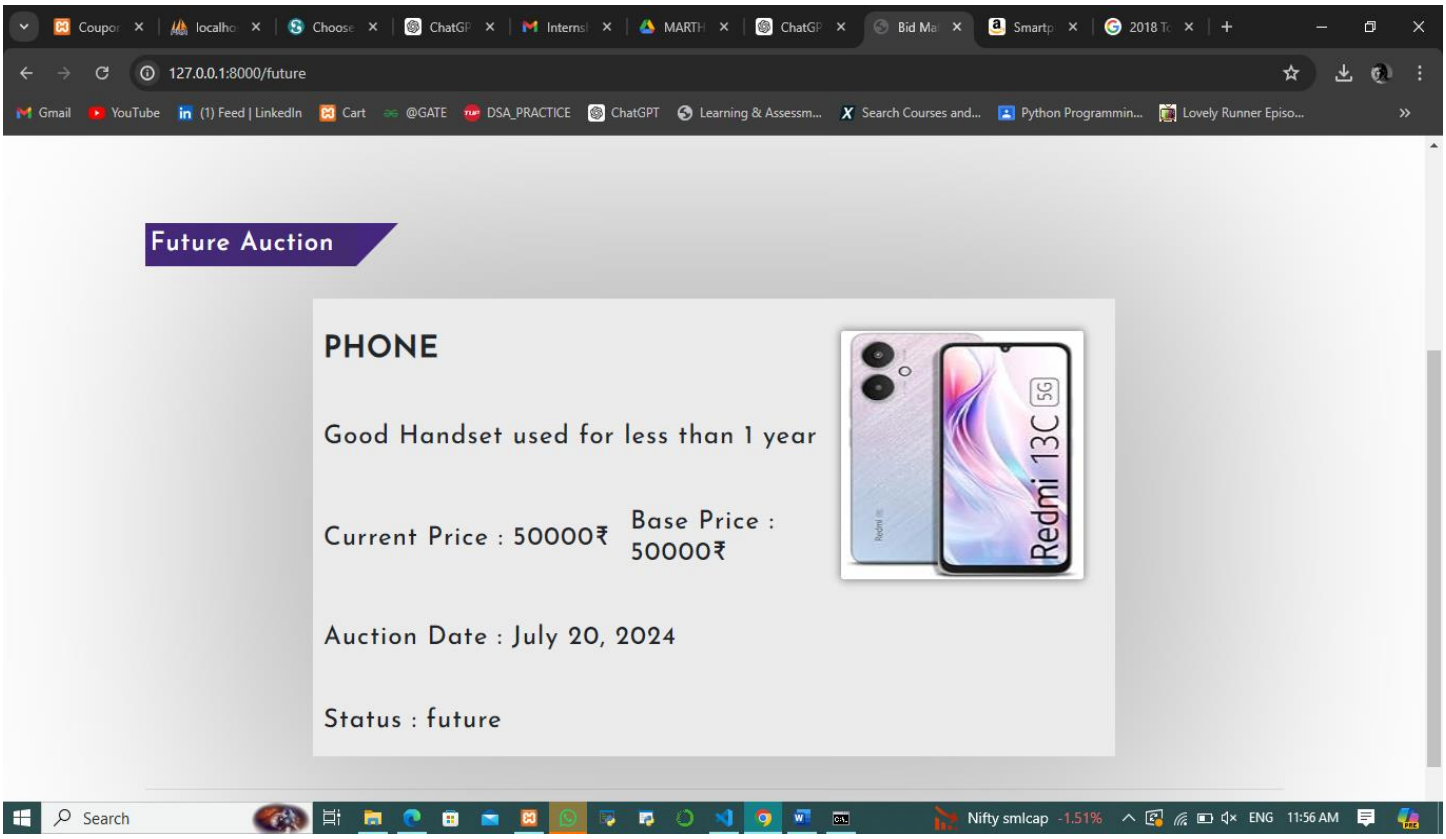
SAMPLE SCREENSHOTS

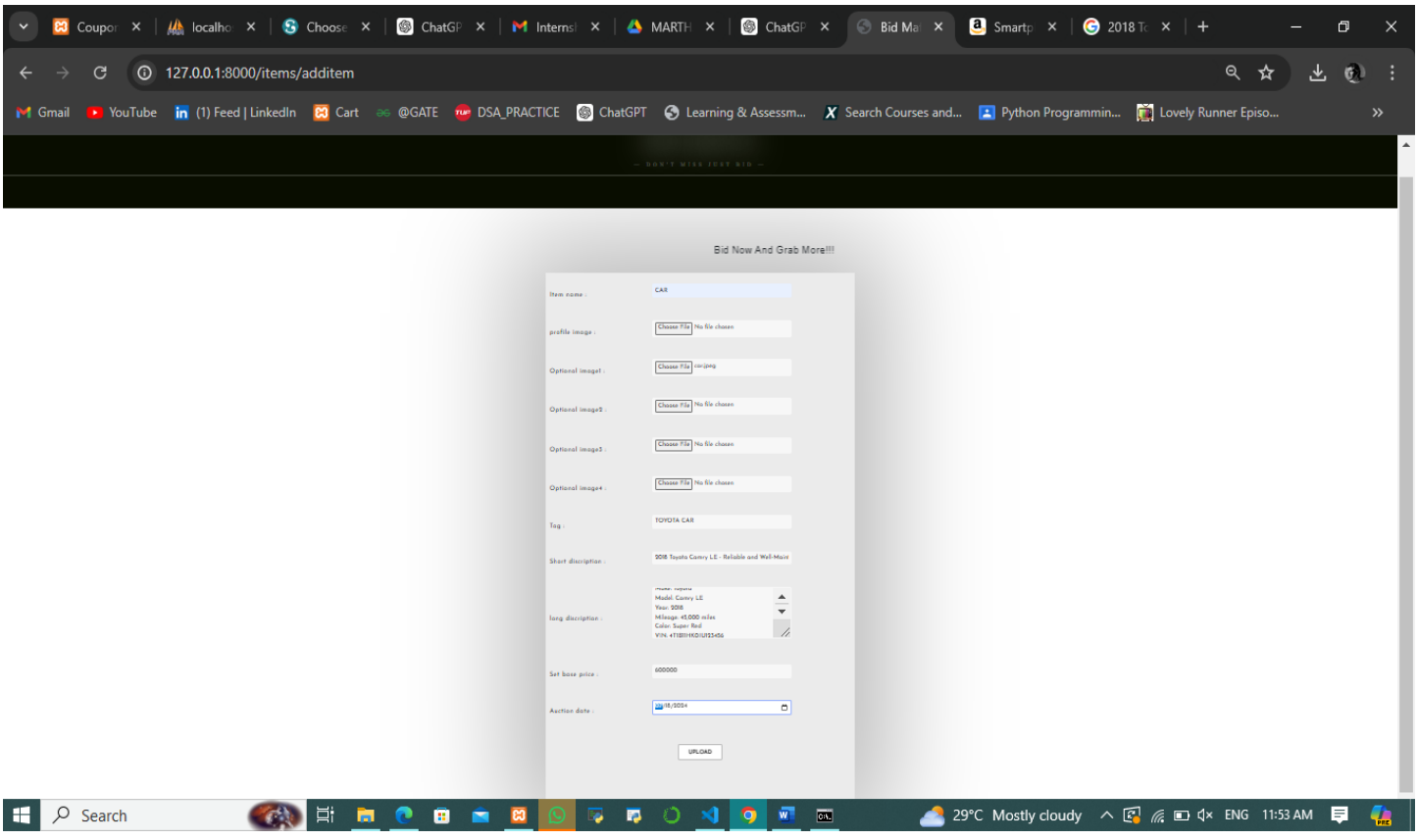





127.0.0.1:8000/items/additem
Optional image1 :
Choose File car.jpeg
Optional image2 :
Choose File No file chosen
Optional image3 :
Choose File No file chosen
Optional image4 :
Choose File No file chosen
Tag :
TOYOTA CAR
Short discription :
2018 Toyota Camry LE - Reliable and Well-Ma
Make: Toyota
Model: Camry LE
Year: 2018
Search
29°C Mostly cloudy
ENG 11:54 AM


127.0.0.1:8000/myprofile
Bid Now And Grab More!!!
First name : surekha
Last name : reddy
Contact no : 8519945319
User name : surekhareddy
Email-id : lakshmiessurekha@gmail.com


127.0.0.1:8000/future
Future Auction
PHONE
Good Handset used for less than 1 year
Current Price : 50000₹
Base Price : 50000₹
Auction Date : July 20, 2024
Status : future
Redmi 13C 5G


127.0.0.1:8000/items/additem
Bid Now And Grab More!!!
CAR
TOYOTA CAR
UPLOAD

CHAPTER 8
CONCLUSION

The scope of the Online Bidding System encompasses the essential features and functionalities required to conduct secure and efficient online auctions. By focusing on usability, performance, and security, the project aims to deliver a robust platform that meets the needs of both users and administrators. Future enhancements will further expand the system’s capabilities and improve the overall user experience.

CHAPTER 9
FUTURE ENHANCEMENT

Integration with payment gateways for secure transactions.

Development of a mobile application to provide a more accessible user experience.

Implementation of advanced analytics to provide insights into user behavior and auction trends.

Integration with social media platforms for easier sharing of auctions and increased user engagement

CHAPTER 10
BIBLIOGRAPHY

Official Django Documentation

Comprehensive guide and reference: https://docs.djangoproject.com/en/stable/

Corey Schafer's Django Tutorials on YouTube

Video tutorials that cover everything from basics to advanced topics: https://www.youtube.com/playlist?list=PLLxk3TkuAYnrbajtAUBtXg4f0iOIZrKYN

Build an E-commerce Website with Django

A detailed tutorial series on building an e-commerce site: https://www.youtube.com/playlist?list=PLXmMXHVSvS-CoYS177-UvMAQYRf8toS3G