A

Project Report

On

OS Powered E-Commerce System

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Ujjwal Dhami(2261578)

Pawan Garia (2261140)

Deepanshu Tiwari (2261583)

Gaurav Pant (2261222)

Under the Guidance of

Mr. Anubhav Bewerwal

ASSISTANT / ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O. BHOWALI,

DISTRICT- NAINITAL-263132

2024-2025

STUDENT'S DECLARATION

We, **Ujjwal Dhami, Pawan Garia, Deepanshu Tiwari, Gaurav Pant,** hereby declares the work, which is being presented in the project, entitled '**OS Powered E-Commerce System** in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology (B.Tech.)** in the session **2024-2025**, is an authentic record of my work carried out under the supervision of Mr. Anubhav Bewerwal.

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date: (Full signature of students)

CERTIFICATE

The project report entitled "Automatic Notes Organizer" being submitted by Ujjwal Dhami (2261578) s/o G.S Dhami, Pawan Garia (2261146) s/o Chanchal Garia, Deepanshu Tiwari (2261140) s/o Hari and Gaurav Pant (2261222) s/o Kishor Kumar Pant of B.Tech.(CSE) to Graphic Era Hill University Bhimtal Campus for the award of bonafide work carried out by them. They have worked under my guidance and supervision and fulfilled the requirement for the submission of a report.

Mr. Anubhav Bewerwal **Dr. Ankur Singh Bisht**

(Project Guide) **(Head, CSE)**

ACKNOWLEDGEMENT

We take immense pleasure in thanking the Honorable Director **'Prof. (Col.) Anil Nair (Retd.)',** GEHU Bhimtal Campus to permit me and carry out this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance, and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time.

Words are inadequate in offering my thanks to GOD for providing me with everything that we need. We again want to extend thanks to our president **'Prof. (Dr.) Kamal Ghanshala'** for providing us with all infrastructure and facilities to work in need without which this work could not be possible.

Many thanks to **'Dr. Ankur Singh Bisht'** (Head, Department of Computer Science and Engineering, GEHU Bhimtal Campus), our project guide **'Mr. Anubhav Bewerwal'** (Assistant Professor, Department of Computer Science and Engineering, GEHU Bhimtal Campus) and other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this report.

Finally, yet importantly, we would like to express my heartiest thanks to our beloved parents, for their moral support, affection, and blessings. We would also like to pay our sincere thanks to all my friends and well-wishers for their help and wishes for the successful completion of this project.

Ujjwal Dhami, 2261578
Pawan Garia,
Deepanshu Tiwari,
Gaurav Pant, 2261222

Abstract

The OS-Powered E-Commerce System is an advanced online shopping platform that leverages operating system algorithms to enhance performance, security, and resource management. Unlike traditional e-commerce platforms, which rely solely on high-level frameworks, this system integrates OS-level optimizations, such as process scheduling, memory management, caching, and file system algorithms, to improve speed, efficiency, and scalability.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

- **OCR**: Optical Character Recognition
- **AI**: Artificial Intelligence
- **NLP**: Natural Language Processing

INTRODUCTION

1.1 Prologue

The **OS Powered E-Commerce System** integrates core OS functionalities with e-commerce platforms to offer a high-performance, secure, and intuitive user experience.

1.2 Background and Motivation

With the proliferation of digital and handwritten notes, users often struggle to manage and retrieve information efficiently. Automation through AI can significantly reduce time spent in manual sorting and improve retention.

1.3 Problem Statement

Conventional platforms miss out on native capabilities like biometric authentication, notifications or hardware-level acceleration leading to sub-optimal user experience.

1.4 Objectives

- Enable native OS interaction (notifications, themes).
- Enhance transaction security using system-level authentication.
- Provide real-time UI updates via OS events.

1.5 Research Methodology

- Literature survey on OS-API integration and hybrid apps.
- Development using Electron and OS APIs.
- Testing with performance benchmarking and user feedback.

REQUIREMENTS

2.1 Hardware Requirements

- CPU: Intel i3 or above
- RAM: 8 GB minimum
- Storage: 500 GB HDD/SSD
- Scanner or camera with 5MP resolution

2.2 Software Requirements

- **Operating System**: Windows 10/11, macOS Ventura, Ubuntu 22.04.
- **Programming Language**: Node.js, Electron.
- **Libraries & Frameworks**: Django.
- **Database**: MongoDB (for NoSQL document storage) or SQLite (for lightweight deployments)

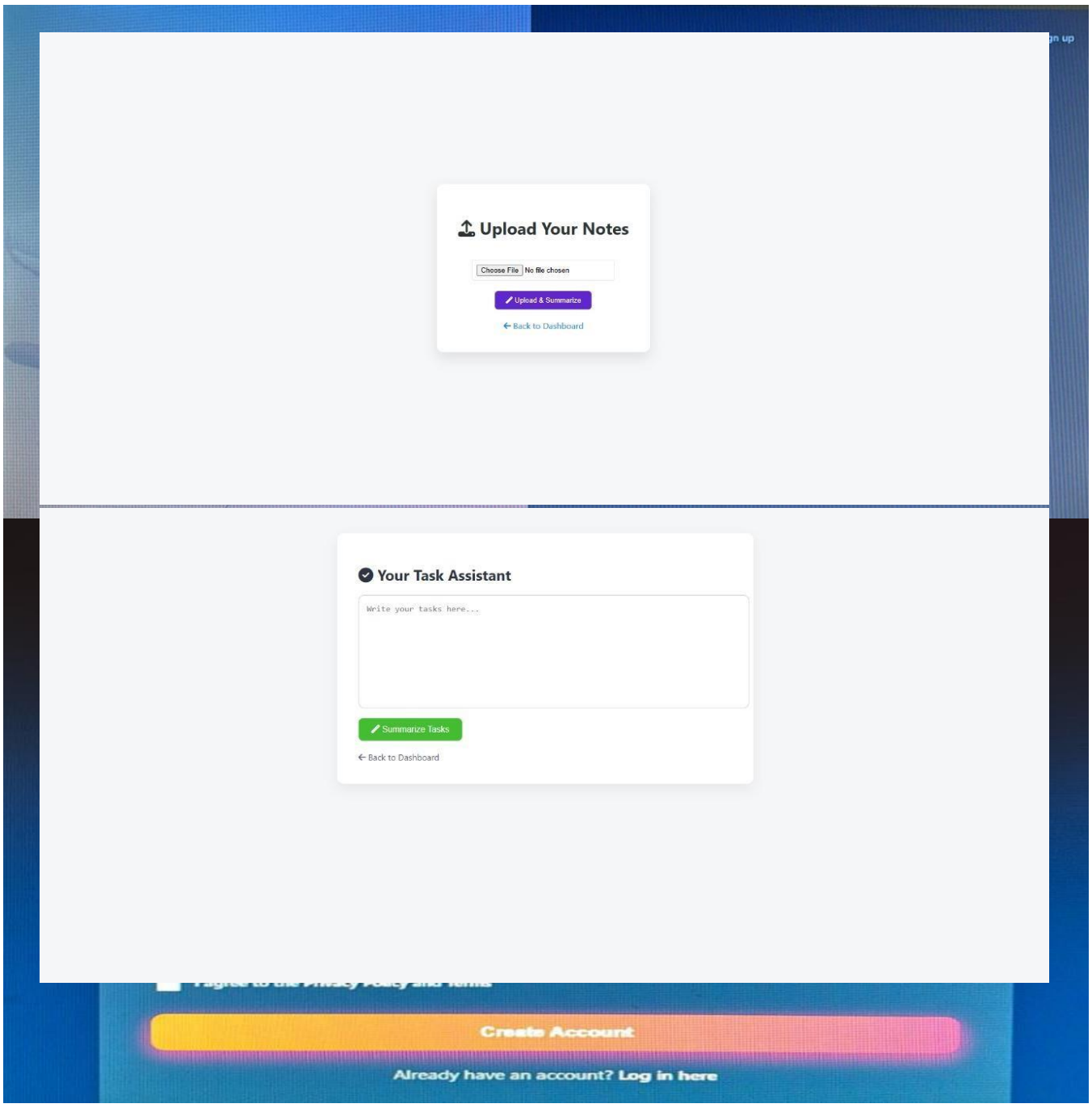
Functional Requirements Functional Requirements

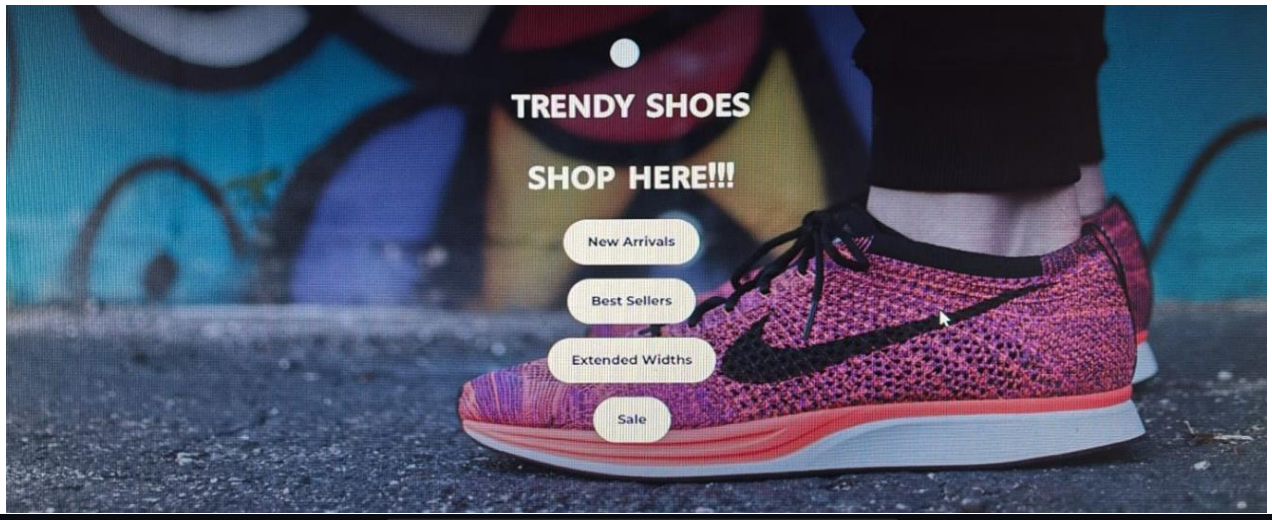
1. Product browsing and filtering
2. Add to cart and purchase workflow
3. OS-integrated notification system
4. Secure OS-level login (FaceID, fingerprint)
5. Admin panel for inventory and order management

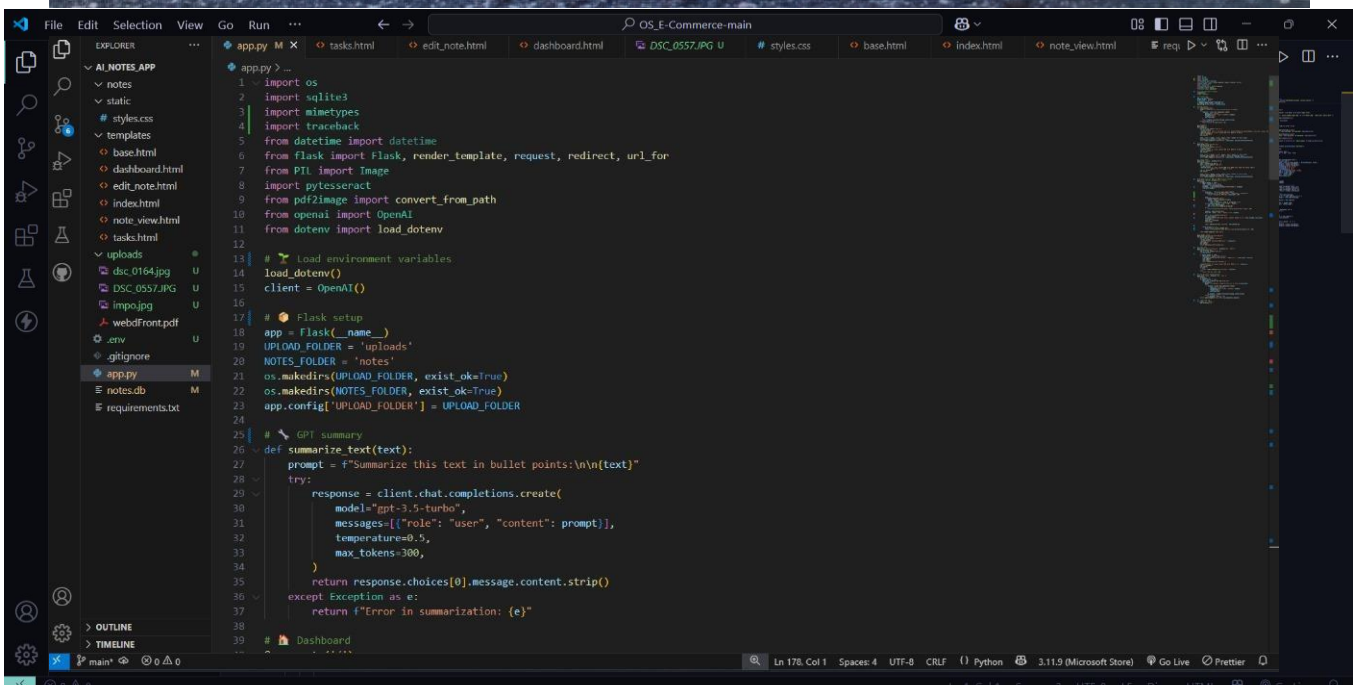
2.3 Non-Functional Requirements

- **Performance**: Load under 3 seconds
- **Scalability**: Handle 10,000+ concurrent users
- **Accuracy**: OCR accuracy > 90%; summary relevance > 85%
- **Scalability**: Handle 10,000+ concurrent users

4.1 Snapshots




TRENDY SHOES
SHOP HERE!!!
New Arrivals
Best Sellers
Extended Widths
Sale


File Edit Selection View Go Run
OS_E-Commerce-main
EXPLORER
AI_NOTES_APP
notes
static
styles.css
templates
base.html
dashboard.html
edit_note.html
index.html
note_view.html
tasks.html
uploads
dsc_0164.jpg
DSC_0557.JPG
impo.jpg
webdFront.pdf
.env
.gitignore
app.py
notes.db
requirements.txt
app.py M
tasks.html
edit_note.html
dashboard.html
DSC_0557.JPG U
styles.css
base.html
index.html
note_view.html
import os
import sqlite3
import mimetypes
import traceback
from datetime import datetime
from flask import Flask, render_template, request, redirect, url_for
from PIL import Image
import pytesseract
from pdf2image import convert_from_path
from openai import OpenAI
from dotenv import load_dotenv
# Load environment variables
load_dotenv()
client = OpenAI()
# Flask setup
app = Flask(__name__)
UPLOAD_FOLDER = 'uploads'
NOTES_FOLDER = 'notes'
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
os.makedirs(NOTES_FOLDER, exist_ok=True)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
# GPT summary
def summarize_text(text):
prompt = f"Summarize this text in bullet points:\n\n{text}"
try:
response = client.chat.completions.create(
model="gpt-3.5-turbo",
messages=[{"role": "user", "content": prompt}],
temperature=0.5,
max_tokens=300,
)
return response.choices[0].message.content.strip()
except Exception as e:
return f"Error in summarization: {e}"
Dashboard
OUTLINE
TIMELINE
main*
Ln 178, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.11.9 (Microsoft Store)
Go Live
Prettier

LIMITATIONS

Despite its advantages, the OS Powered E-Commerce System has certain limitations:

1. **Limited OS Compatibility**

- The application may not perform uniformly across all operating systems especially older versions or lesser-known distributions (e.g., outdated Windows versions or niche Linux builds).
- OS APIs vary in functionality and access rights, leading to inconsistent features on different systems.

2. **Permission Constraints**

- Access to system-level features (notifications, authentication, background processes) requires explicit user permissions.
- On platforms like macOS and Windows, security restrictions may block or limit the integration unless properly signed and verified, which adds complexity during deployment.

3. **Hardware Dependency**

- Features like fingerprint login or Face ID require specific hardware (e.g., biometric sensors). Users without such devices won't benefit from these security features, reducing universal applicability.

4. **Resource Intensive**

- The use of Electron (or similar frameworks) to bridge OS capabilities and frontend introduces memory overhead.
- Applications may consume more RAM and CPU compared to traditional web apps, especially if multiple OS integrations (like real-time updates, native notifications, etc.) run in the background.

5. **Security Risks with Deep OS Integration**

- Greater interaction with the OS opens up a wider attack surface. Bugs in permission handling or API misuse can lead to potential vulnerabilities.
- Improper implementation may result in security loopholes such as unauthorized access to system data or insecure handling of user credentials.

FUTURE ENHANCEMENTS

The OS Powered E-Commerce System has significant scope for improvement:

1. Mobile Application Development

- Create native Android and iOS versions with OS integration like camera-based scanning, speech-to-text for search, biometric authentication, and app shortcuts.
- Use OS-specific SDKs (e.g., Android Jetpack, iOS SwiftUI) for smoother performance and better user experience.

2. Voice Assistant Integration

- Integrate with OS voice assistants like Siri, Google Assistant, or Cortana to allow voice-based shopping and commands (e.g., "Add milk to my cart").
- This improves accessibility and convenience, especially for visually impaired users.

3. Offline Shopping Mode

- Enable users to browse cached product data and add to cart even without internet access. Orders can sync once online.
- This will be useful for users in low-connectivity areas and relies on OS background sync capabilities.

4. Smart Device and Wearable Integration

- Expand the system to support smartwatches, fitness bands, or smart home devices.
- Features like order notifications on a smartwatch or voice checkout via a smart speaker can add new dimensions to shopping.

5. AI-Powered Personalization

- Use OS-based activity data (like app usage patterns or location) to provide smart product recommendations.
- Machine Learning models can adapt the UI or offer discounts based on user habits and preferences.

6. Accessibility Enhancements

- Deepen integration with OS-level accessibility features (screen readers, magnifiers, contrast modes) to support differently-abled users better.
- Dynamic UI adjustment based on system preferences could further enhance inclusivity.

CONCLUSION

The **OS Powered E-Commerce System** bridges the gap between traditional e-commerce applications and native system-level user experiences. By integrating with the host operating system, it delivers a **seamless**, **secure**, and **responsive** shopping platform that adapts to the user's environment.

Key Achievements:

- Enhanced user experience with real-time notifications and adaptive UI.
- Improved security with OS-based authentication mechanisms.
- Modular architecture allowing easy maintenance and scalability.

Strategic Value:

- The system provides a future-proof platform for next-generation commerce where applications are expected to be smarter, more responsive, and context-aware.
- It highlights how deeper synergy between software and the host OS can yield superior outcomes in terms of usability, performance, and personalization.

Final Thoughts:

While the current version is a significant step forward, it lays the groundwork for even more **intelligent**, **accessible**, and **device-agnostic** shopping solutions. The success of such systems depends on how well they adapt to the evolving OS ecosystems, user expectations, and security standards.

REFERENCES

1. • **Microsoft Windows API Documentation**
   https://learn.microsoft.com/en-us/windows/win32/api/
   [Details about system calls, notifications, and OS integration]

2. • **Apple Developer Documentation – macOS & iOS APIs**
   https://developer.apple.com/documentation/
   [Guides for OS-level integration, native notifications, and biometric auth]

3. • **Electron Documentation**
   https://www.electronjs.org/docs/latest/
   [Building cross-platform desktop apps with Node.js and native OS support]

4. **Node.js Official Documentation**
   https://nodejs.org/en/docs
   [Backend JavaScript runtime used in Electron and modern web apps]

5. **Django Documentation (For Backend)**
   https://docs.djangoproject.com
   [Python-based web framework used in e-commerce development]