CHAPTER 1

INTRODUCTION

In today's digital age, online shopping has become a common and convenient method for purchasing goods and services. However, individuals with visual impairments—particularly those who are blind—face major challenges in navigating traditional e-commerce platforms [1]. According to global estimates, there are around 161 million people who are visually impaired, out of which 36 million are completely blind [6]. These individuals often require continuous assistance from family members, friends, or caregivers to perform basic online tasks, including shopping.

Consider the scenario of a blind individual who lives independently. While they may have access to the internet, the process of shopping online becomes difficult, frustrating, and at times impossible without someone physically present to assist them[1]. Tasks that are trivial for sighted individuals—such as selecting the right product, identifying colors or sizes, and navigating through complex web interfaces—pose significant barriers for the visually impaired [3][6].

This project proposes the development of a voice-guided e-commerce web application specifically tailored to assist visually impaired users. The objective is to enable blind users to independently navigate an online shopping platform, make selections using audio prompts, and complete purchases without external help [5][7]. By using voice output for system instructions and input selection through simple button presses, the platform aims to be intuitive and accessible. The system also incorporates automatic clothing pattern and color recognition, allowing users to understand visual details of products that would otherwise be inaccessible to them [7][9].

1.1 Scope and Objective

- **Scope of the Project**

The scope of this project revolves around the development of a voice-assisted e-commerce web application that empowers visually impaired and blind individuals to perform online shopping independently, securely, and efficiently[4]. In today’s digital-first world, most services are designed with visual interfaces, unintentionally excluding millions of visually impaired users from full participation in online retail [1][6].

This system seeks to bridge the accessibility gap in e-commerce, offering a comprehensive solution that transforms the online shopping experience for blind users through audio guidance, voice interaction, and simplified navigation [3][4].

The platform will enable blind users to:

- Register, login, and authenticate securely using voice-guided interfaces[7].
- Navigate product categories and subcategories through audio prompts[5].

- Receive real-time spoken descriptions of products, including features like color, size, and design [7][9].
- Add products to cart or buy directly through intuitive keyboard/button interactions [2][7].
- Receive voice-based product recommendations [5].
- Complete the checkout process independently [3][6].

On the admin side, the system will allow e-commerce administrators to manage the store efficiently through conventional web-based controls for products, categories, and user management.

The application will be developed using ASP.NET and will be accessible via standard web browsers on desktops, laptops, and screen-reader-equipped devices. It is designed to comply with web accessibility standards (such as WCAG 2.1), ensuring usability for users with a wide range of visual impairments [6].

- **Key Objectives**

1. Empower Visually Impaired Users with Independence

The central goal of this project is to offer blind and visually impaired individuals the ability to shop online without needing external assistance. By using voice prompts and audio feedback, users can perform all major actions—browsing, selecting, and purchasing—independently [5][7].

2. Replace Visual Dependence with Voice-Based Interaction

Most websites depend heavily on visual elements for interaction, such as clickable images, drop-downs, and navigation menus. This project replaces these with audio alternatives:

- Every screen or action includes voice instructions [1].
- Users interact through keyboard shortcuts or button presses, making the process simple and consistent [2].

3. Provide Audio Descriptions of Visual Product Features

One of the biggest hurdles in online shopping for blind people is understanding visual product characteristics, such as clothing colors or patterns. This system integrates AI-based image recognition tools that analyze product visuals and translate them into meaningful spoken descriptions [7][9].

4. Simplify the User Journey for Accessibility

The project that we have build aims to create a clean, distraction-free, and responsive user experience tailored for blind users:

- Straightforward navigation
- No visual clutter
- No reliance on cursor or touchpad
- Clear voice feedback at every step [4][6]

5. Ensure Security and Privacy

As with any e-commerce platform, security is critical. The system will:

- Handle user data with encryption.
- Ensure secure login and password recovery [6].
- Provide voice feedback without compromising privacy.

6. Enable Admin-Level Management of Product Data

The admin module will be developed for e-commerce managers to:

- Add, update, and delete categories or products.
- Monitor user registrations and order histories.
- Maintain the inventory and recommend products via the platform's backend logic.

7. Comply with Accessibility Standards

The project targets compliance with international accessibility guidelines, particularly:

- WCAG 2.1 (Web Content Accessibility Guidelines)
- ARIA roles for screen readers
- Compatibility with text-to-speech systems like NVDA and JAWS [6][3]

8. Promote Digital Inclusivity

In the broader context, this project aims to contribute to inclusive technology practices by making digital commerce accessible to a marginalized community. It highlights the potential of human-centered design and innovation in solving real-world challenges [4][9].

1.2 Modules and Their Description

The proposed e-commerce web application consists of two primary modules: the Admin Module and the User Module, each containing sub-modules to manage their specific functionalities effectively.

i. **Admin Module**

The admin plays a central role in managing the overall functionality and content of the platform. The sub-modules available to the admin include:

ii. **Login**

The admin logs into the system using a secured login interface. Only authenticated admins can access the back-end functionalities of the application.

iii. **Manage Categories**

The admin can create and manage both main categories (such as Clothing, Electronics, etc.) and sub-categories (like Men's Wear, Women's Wear, Mobile Phones, etc.). This feature allows the admin to organize products efficiently, making it easier for users to navigate.

iv. **Manage Products**

This module allows the admin to add, update, or delete products from the inventory. Product details include name, description, price, color, size, and category. It is critical for maintaining a fresh and accurate product catalog for users.

v. View Users

The admin can access a list of all registered users, including their registration details. This information can help in analyzing user behavior, preferences, and feedback.

vi. View Orders

The admin can monitor all orders placed by users, including product details, order status, delivery address, and payment information. This module is essential for order management and fulfillment.

vii. User Module

The user module is designed with a focus on accessibility and ease of use for visually impaired individuals. It includes the following sub-modules:

- **Registration**

Users must register by providing essential details such as name, email, address, and creating a secure login ID and password. Each input field is accompanied by system-generated voice prompts that guide the user through the registration process. This ensures that blind users can enter information accurately and confidently.

- **Login**

Once registered, users can log into the system using their credentials. The system responds with audio confirmation indicating whether the login was successful or if there was an error (e.g., incorrect password). This immediate feedback is crucial for visually impaired users.

- **Category and Product Navigation**

After successful login, the system uses voice instructions to guide the user through the available product categories. Each category is assigned a specific numeric or alphabetic input key, which the user can press to select a category. The system then reads out the corresponding sub-categories and finally the product details.

For example, if the user presses "1" for Clothing, the system will further prompt with sub-categories like "Men’s Wear" or "Women’s Wear", and once a sub-category is selected, it will begin reading out the product names, colors, prices, and other important attributes.

- **Add to Cart and Buy Now**

Users have the option to add products to their cart for later checkout or buy a product immediately. Each option is navigated via voice guidance, and the corresponding action can be selected using a simple key press.

- **Product Recommendations**

On the cart page, the system will suggest recommended products based on the items already added. These recommendations are also read aloud, giving users additional shopping options without requiring visual input.

1.3 Existing System vs Proposed System

I. Existing System

Currently, most popular e-commerce websites cater exclusively to sighted users. These platforms rely heavily on graphical user interfaces (GUIs), mouse navigation, and visual content like images and videos. While some platforms offer limited screen reader support, they do not provide full accessibility for blind users, especially when it comes to:

- Understanding product visuals such as colors or patterns.
- Navigating complex page structures.
- Managing secure payments independently.
- Accessing meaningful product descriptions through audio alone.

As a result, visually impaired users often rely on companions for online shopping or avoid it altogether in favor of in-person purchases—again, requiring assistance.

II. Intented System

The proposed system revolutionizes this experience by:

- Providing a fully voice-interactive e-commerce interface.
- Enabling keyboard/button-based navigation for blind users.
- Incorporating voice prompts for every user action, from login to checkout.
- Implementing automatic visual content description, such as color and pattern recognition for clothing.

The system allows visually impaired users to shop independently, making the same purchases that sighted users can, on a platform designed specifically for their needs.

1.4 Literature Survey

The increasing reliance on e-commerce platforms has revolutionized the way consumers access goods and services. However, this transformation has disproportionately disadvantaged individuals with visual impairments, who face significant barriers in using traditional e-commerce websites. Numerous studies in the field of Human-Computer Interaction (HCI) and accessibility design have identified that most mainstream online shopping portals lack support for screen readers, voice navigation, and non-visual cues. This deficiency renders them inaccessible to blind users, who often require assistance to complete online purchases. According to research published by the World Health Organization, there are over 36 million blind individuals globally, yet the digital market rarely accommodates their needs. These findings emphasize the necessity for inclusive e-commerce platforms that enable independence in online shopping for all users, regardless of physical limitations.

Previous efforts in improving web accessibility have largely focused on implementing screen reader compatibility and basic keyboard navigation. While these features are helpful, they are not sufficient for fully independent e-commerce interactions. Researchers have shown that screen readers often misinterpret dynamic visual content such as carousels, dropdown menus, and product images. This creates an incomplete user experience where visually impaired individuals are unable to make informed decisions. Some assistive technologies, like voice-based search or text-to-speech plugins, have attempted to mitigate these issues, but integration into existing platforms is often inconsistent or non-intuitive. Studies highlight that effective solutions must be embedded into the design architecture from the start, rather than being retrofitted as external accessibility tools.

Emerging literature has proposed the use of Artificial Intelligence (AI) for image recognition and audio feedback to improve e-commerce accessibility. Projects like Microsoft’s Seeing AI and Google’s Lookout app demonstrate how computer vision can interpret visual product features like color, shape, and patterns. These technologies, when adapted to e-commerce environments, could enable blind users to better understand product appearances without assistance. In parallel, academic work on voice-guided navigation systems has shown that audio prompts can replace visual cues, enabling blind users to browse and interact with websites independently. The use of simple keyboard or button-based interactions alongside voice feedback creates a more accessible and intuitive user interface.

Despite these advancements, most commercial e-commerce systems still lack robust support for blind users. The Web Content Accessibility Guidelines (WCAG) 2.1 and ARIA standards provide a strong foundation for accessible design, yet compliance across e-commerce sites remains low. Scholarly articles point out that developers often overlook accessibility due to a lack of awareness or resources. As a result, there is a growing call for customized platforms built specifically for visually impaired users. These systems are expected to go beyond compliance, embedding accessibility into core functionalities such as secure login, voice-based browsing, AI-driven product description, and independent checkout processes. The literature survey supports the approach adopted in this project, which integrates AI-based color and pattern recognition, screen reader compatibility, and a fully voice-guided user interface developed using ASP.NET. By focusing on user-centered design principles and adhering to accessibility standards, the proposed platform aims to empower blind users with the ability to shop online independently. This aligns with the broader movement in academic and industry research toward digital inclusivity, highlighting the importance of designing technology that serves the needs of all users, including those with disabilities.

1.5 Drawbacks of the Existing System

The traditional systems have multiple shortcomings when it comes to accessibility for the blind:

- **Poor Accessibility**: Existing e-commerce sites are not designed with visually impaired users in mind.
- **High Dependence**: Users must rely on others to help complete even basic tasks online.
- **Inaccuracy**: Screen readers often misinterpret visual elements or skip critical details.
- **Low User-Friendliness**: Interfaces are overly complex for non-sighted users.
- **Time-Consuming**: Navigating through a site using assistive tools without tailored support is slow and frustrating.

1.6 System – Detailed Overview

To overcome the above limitations, a new, accessible e-commerce platform has been developed using ASP.NET within Visual Studio, targeted specifically at blind users.

Key Features of the Proposed System:

- **Admin and User Access Levels**: The system supports separate modules for admins and users, each with defined roles and permissions.
- **Accessible Registration & Login**: Every form field is accompanied by system-generated voice instructions, ensuring clarity and ease of input.
- **Voice-Guided Navigation**: From category selection to final checkout, the user is guided through **audio prompts** and input key options.
- **Voice-Based Recommendations**: Smart suggestions help users discover related products based on their cart.
- **Color and Pattern Recognition**: Automatically describes visual features like colors, textures, and patterns for clothing items, bridging the gap between sight and understanding.
- **Cart and Checkout Assistance**: Users can build a shopping cart or buy instantly with voice assistance at each step.
- **Order Tracking**: Admin can view and manage all orders, ensuring that fulfillment and user queries are handled efficiently.

Summary

This project focuses on developing a voice-guided e-commerce website specifically for visually impaired users, enabling them to shop online independently. It uses audio prompts and simple keyboard inputs to navigate products, hear descriptions of colors and patterns, and complete purchases without assistance. Built with ASP.NET and designed to meet accessibility standards, the platform aims to make online shopping secure, easy, and inclusive for visually impaired person while providing admins with tools to store effectively.

CHAPTER 2

METHODOLOGY

This chapter elaborates on the methodology employed in the development of an accessible, voice-enabled e-commerce website tailored for visually impaired users. The methodological approach integrates theories from Human-Computer Interaction (HCI), accessibility design standards, and software engineering paradigms, ensuring that the solution is both functional and inclusive [4][6].

2.1 Methodological Approach

The approach followed is grounded in the User-Centered Design (UCD) methodology, which emphasizes the involvement of end-users throughout the design and development process [4]. UCD is especially important when developing for users with disabilities, as their needs often differ significantly from the general population.

Additionally, the Iterative Software Development Life Cycle (SDLC) model was used. This model supports cyclic development, allowing for incremental builds and continuous refinement based on user feedback. The project lifecycle consisted of the following stages:

1. Requirement Gathering and Analysis
2. System Design and Modeling
3. Implementation
4. Integration and Testing
5. Deployment and Maintenance

Each phase was iterative, allowing developers to revisit earlier stages based on insights gained during testing and user interaction [4].

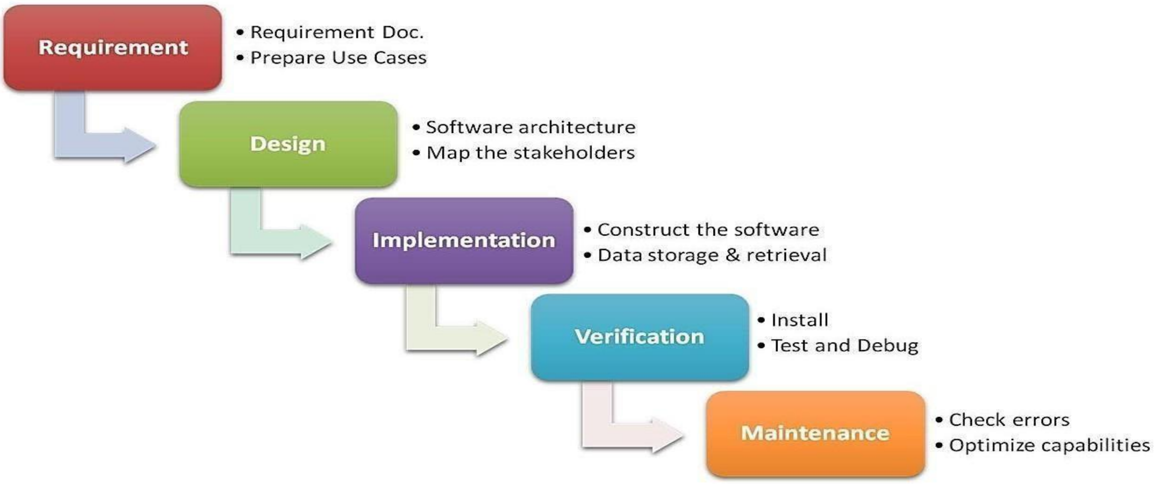


Fig 2.1 Methodology Flow Chart

2.1 Requirement Gathering and Analysis

Requirement gathering is a foundational step in any development lifecycle. It involves collecting functional and non-functional requirements to shape the design and development of the system.

Techniques Used

- **Literature Review:** An extensive survey of academic literature on accessibility, assistive technologies, and web usability was conducted.
- **Empirical Interviews:** Structured interviews with visually impaired individuals provided real-world insights into navigation challenges.
- **Observation:** Existing e-commerce websites were analyzed for accessibility shortcomings using screen readers and keyboard-only navigation.

Theoretical Basis

- **Norman’s Model of Interaction** was used to understand how visually impaired users interact with technology—focusing on stages like intention formulation, action execution, and evaluation of results.
- **WCAG 2.1 (Web Content Accessibility Guidelines)** served as the benchmark for accessibility requirements, emphasizing operability, perceptibility, and robustness.

2.1.2 Extracted Requirements

Functional Requirements:

- Voice prompts for all system actions and inputs.
- Keyboard navigation without any mouse dependency.
- Screen reader compatibility for all web pages.
- Voice-based product search and checkout process.

Non-Functional Requirements:

- **Accessibility:** High adherence to WCAG 2.1 standards.
- **Usability:** Simple navigation, audible feedback.
- **Performance:** Fast response to user input and voice commands.
- **Scalability:** Ability to expand product listings and voice features.

2.2 System Design

The design of the system was based on **modular architecture**, ensuring separation of concerns and maintainability. The **Model-View-Controller (MVC)** architectural pattern was employed to separate user interface, data, and control logic.

2.2.1 Admin and User Modules

Admin Module:

- Structured using role-based access control (RBAC) for security.
- Built for CRUD operations on categories, subcategories, and products.
- Allows monitoring of user activity and transactions.

User Module:

- Incorporates **Voice Interaction Design (VID)** principles.
- Navigation is based on **linear and hierarchical models** suited for non-visual interfaces.
- Supports audio cues for confirmation, navigation, and assistance.

Theoretical Models and Diagrams

- **Use Case Diagrams:** Modeled using UML to show interactions between actors (user/admin) and the system.
- **ER Diagram:** Designed to represent relational database structure, capturing entities like Users, Products, Orders.
- **Activity and Sequence Diagrams:** Depict the control flow and communication between objects and actors.
- **DFD (Data Flow Diagram):** Shows data transformation and movement within the system.
- **System Architecture Diagram:** Displays a three-tier architecture involving:
  - **Presentation Layer** (UI and audio output),
  - **Application Layer** (business logic),
  - **Data Layer** (database operations).

2.3 Implementation Strategy

The project used ASP.NET with C# for backend logic and Microsoft SQL Server as the database. The development process was divided into sprints using Agile methodology, enabling frequent testing and feedback cycles.

Justification for Technology Stack

- **ASP.NET:** Chosen for its scalability, security features, and integration with Microsoft tools.
- **C#:** Object-oriented programming allows modular and reusable code.
- **JavaScript Web Speech API:** Provides real-time voice synthesis capabilities for browser-based audio feedback.
- **MS SQL Server:** Offers reliable, secure data storage with relational capabilities.
- **HTML5 & ARIA:** Enhance semantic structure and accessibility of web interfaces.

Accessibility Integration Techniques

- **Semantic HTML:** Ensures that screen readers can interpret content correctly.
- **ARIA (Accessible Rich Internet Applications) roles:** Enhance screen reader capabilities.
- **Keyboard Event Handling:** JavaScript functions capture key events (e.g., Tab, Enter, Arrow keys) to enable smooth navigation.
- **Voice Output Mapping:** Each UI element is associated with a specific voice prompt.

2.4 Testing Methodology

Testing involved both traditional software testing techniques and specialized **accessibility validation methods**.

Testing Techniques

- **Black-Box Testing:** Verifying functionality without internal code analysis.
- **White-Box Testing:** Ensuring internal logic correctness through code inspection.
- **Manual Accessibility Testing:** Using screen readers (NVDA, JAWS) to evaluate usability.
- **Automated Testing Tools:** AXE, WAVE, and Lighthouse for accessibility audits.

Types of Testing Conducted

- **Unit Testing:** Verified individual functions (e.g., login, add to cart).
- **Integration Testing:** Checked communication between voice module and database.
- **System Testing:** Evaluated complete platform for bugs and performance.
- **Usability Testing:** Observed how visually impaired users interacted with the system.
- **Validation Testing:** Ensured that functional and non-functional requirements were met.

2.5 Deployment and Maintenance

Deployment was carried out on a Windows platform due to native support for .NET applications.

Deployment Environment

- **OS:** Windows 10
- **Web Server:** IIS (Internet Information Services)
- **Browser Compatibility:** Optimized for Google Chrome
- **Hardware:** Minimum i3 CPU, 1 GB RAM, 50 GB HDD

Maintenance Activities

- **Routine Data Backups:** To ensure data safety.
- **Voice Script Updates:** As user needs evolve, voice commands and feedback scripts are periodically refined.
- **Security Patches:** Updated based on latest vulnerabilities.
- **User Feedback Loop:** Continuous improvement based on user experience.

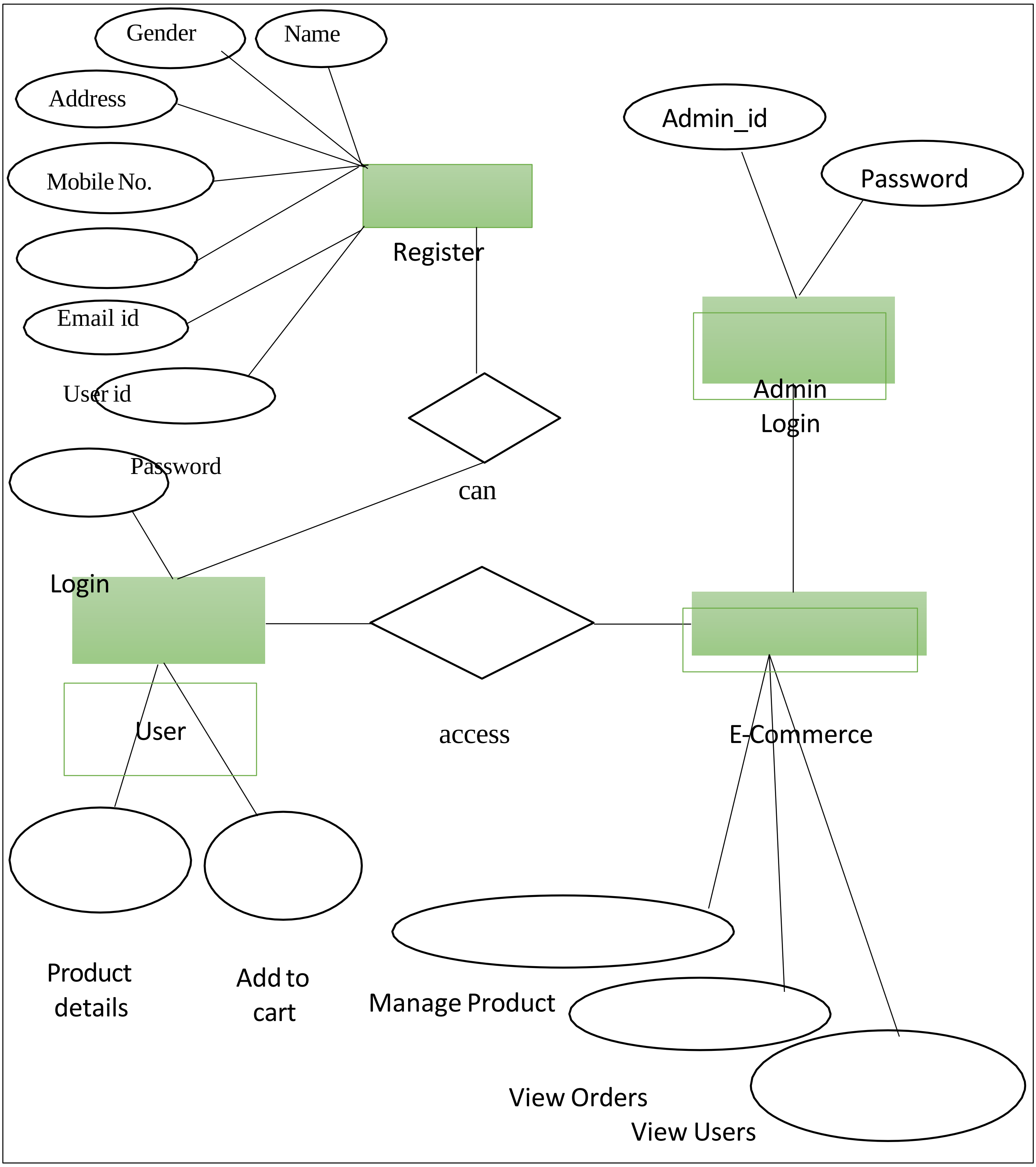
Summary

The methodology used in this project is a robust blend of design theory, accessibility research, and engineering discipline. By grounding the development process in UCD and accessibility standards, the platform ensures that visually impaired users are not only accommodated but empowered. The choice of tools, frameworks, and testing models further reinforces the platform’s commitment to inclusivity, usability, and performance.

Chapter 3

PROJECT DESIGN

3.1 E-R Diagram



The diagram represents an Entity-Relationship (ER) model for an e-commerce system, illustrating how different users interact with the platform. It includes two main types of users: regular users and administrators. Users begin by registering on the platform, providing details such as name, gender, address, email ID, and login credentials. Once registered, they can log in and access the e-commerce platform, where they are able to view product details and add items to their cart.

On the other hand, administrators log in using an admin ID and password. After successful authentication, they gain access to the e-commerce system where they can manage products. This management capability includes viewing user information and order details. The system includes relationships such as "register," "login," and "can," which define how users and admins gain access and interact with the platform. Overall, the ER diagram provides a clear structure for designing the backend database of an e-commerce application, ensuring the logical flow of user registration, product interaction, and administrative control.

3.2 Use Case Diagram

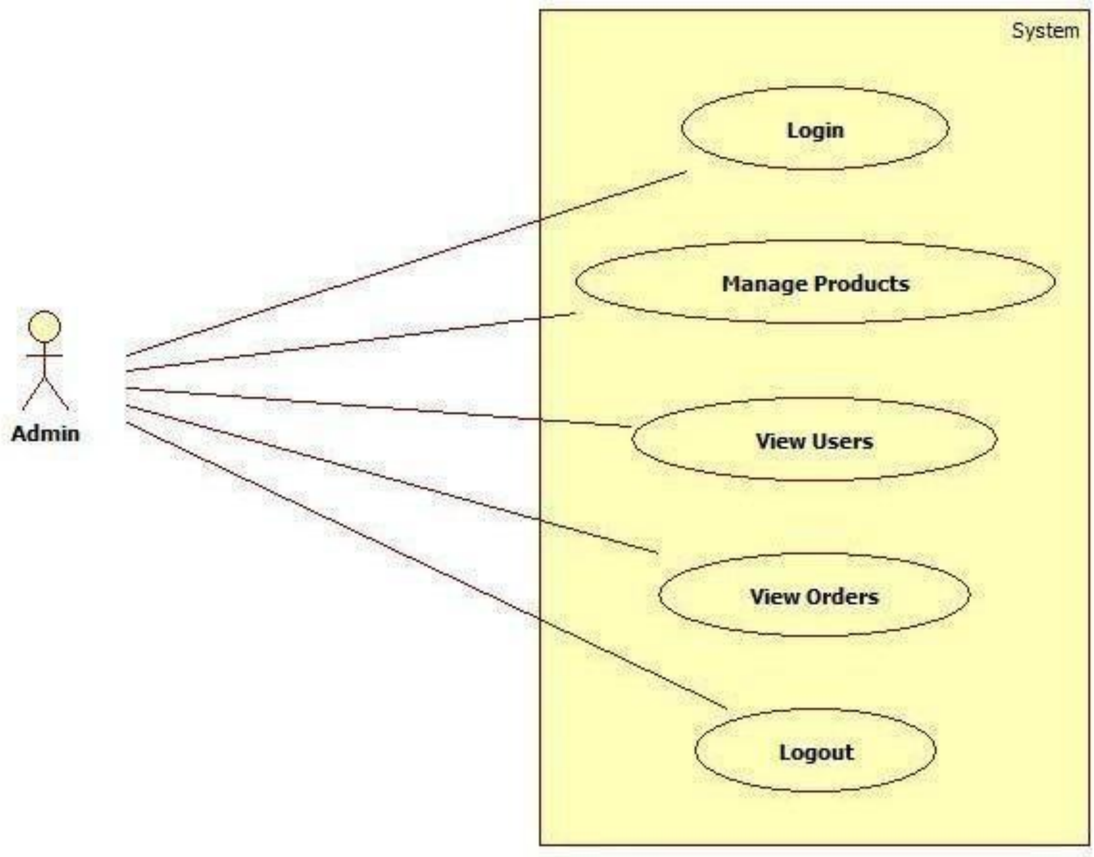


Fig. 3.2 (a) Admin Use Case Diagram

The diagram is a use case diagram that illustrates the interactions between an Admin and the E-commerce system. It shows the various actions that an admin user can perform within the system. The admin is represented by a stick figure on the left, and the system functionalities are enclosed within a yellow box labeled “System” on the right.

The use cases available to the admin include: Login, which allows the admin to access the system; Manage Products, where the admin can add, update, or remove products from the platform; View Users, enabling the admin to see user information and activity; View Orders, which provides access to order details for monitoring and processing; and Logout, which securely ends the admin's session. Each action is connected to the admin, indicating that they are the actor responsible for performing these tasks. This diagram helps visualize the scope of admin privileges and the key functionalities provided to manage the e-commerce system effectively.

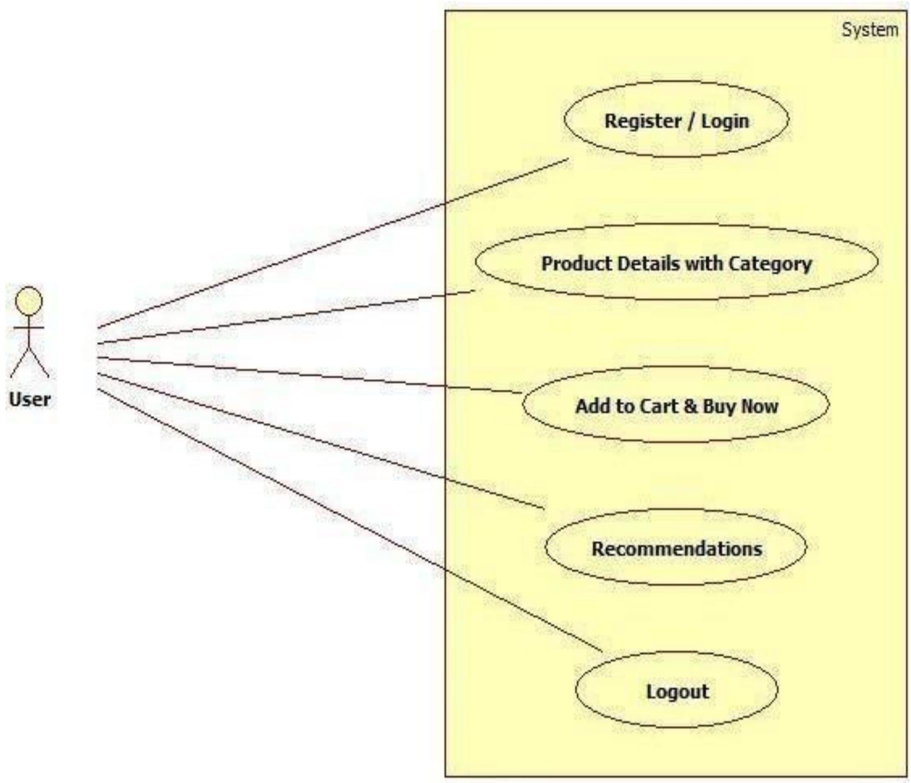


Fig.3.2 (b) User Use Case Diagram.

The diagram shown is a use case diagram that represents the interactions between a User and the E-commerce system. It highlights the core functionalities available to a registered or guest user on the platform. The user is depicted as a stick figure on the left side, while the right side, enclosed in a yellow box labeled "System," lists the various use cases or actions that a user can perform.

The available use cases for the user include Register / Login, which allows users to either create a new account or access an existing one. After logging in, users can view Product Details with

Category, which enables browsing and filtering of products based on categories. They also have the option to Add to Cart & Buy Now, offering flexibility to either save items for later purchase or proceed to instant checkout. The Recommendations use case provides personalized product suggestions based on user activity or preferences, enhancing the shopping experience. Lastly, the Logout function allows users to securely exit the system.

This diagram is useful in understanding the main features offered to users and how they interact with the system to browse, shop, and manage their accounts efficiently.

3.3 Sequence Diagram

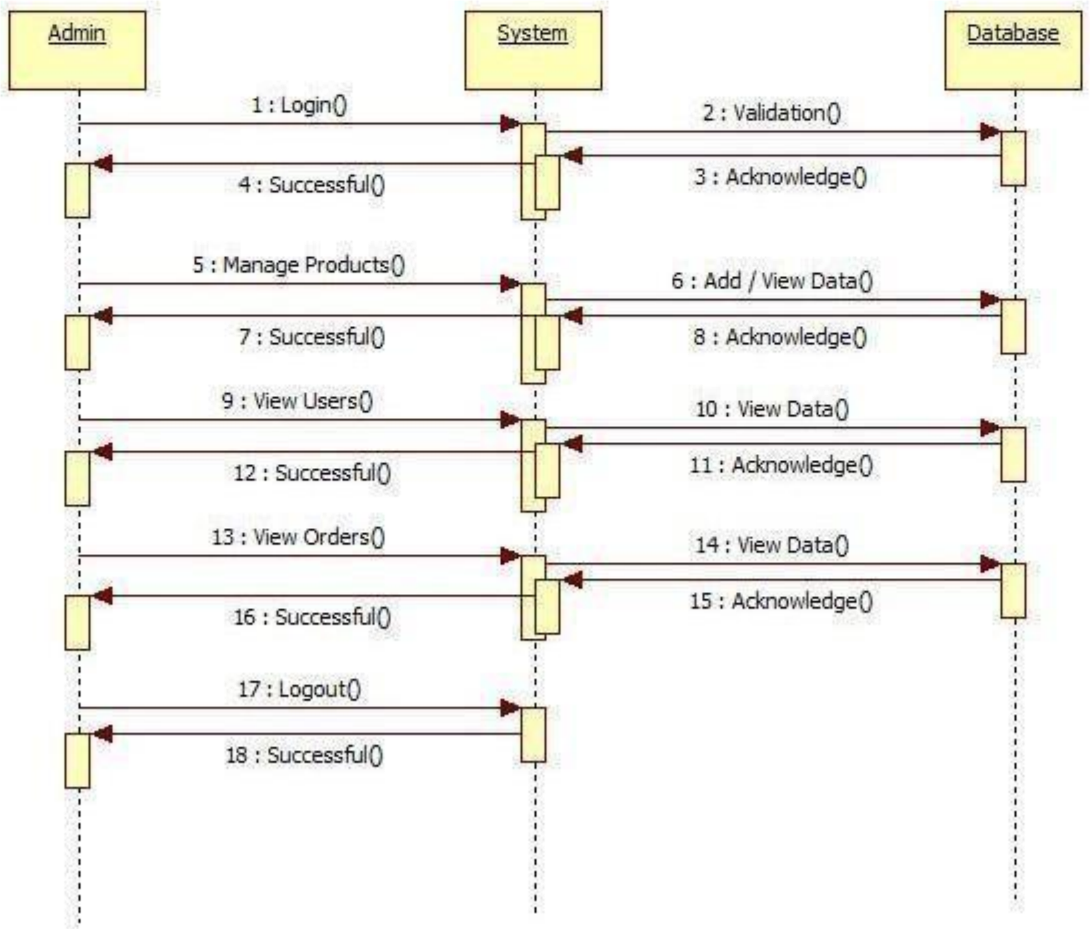


Fig. 3.3. Admin Sequence Diagram

The UML sequence diagram illustrates the interaction between an Admin, the System, and a Database within an e-commerce application. It captures the sequential flow of operations performed by the Admin, including logging in, managing products, viewing users and orders, and logging out. The process begins with the Admin initiating a login request to the System, which in turn validates the credentials through the Database. Upon successful validation, the Database acknowledges the request, and the System confirms the login to the Admin. For managing products, the Admin sends a request to the System, which processes the operation by communicating with the Database to add or view data. Acknowledgment from the Database is followed by a success message to the Admin. Similarly, when the Admin chooses to view users or orders, the System retrieves the respective data from the Database and sends back a confirmation upon successful acknowledgment. Finally, when the Admin logs out, the System ends the session with a success response. This diagram effectively models the end-to-end communication flow, emphasizing clear, bidirectional messaging and system reliance on the Database for validation and data retrieval, ensuring a smooth and traceable transaction process.

3.4 Activity Diagram

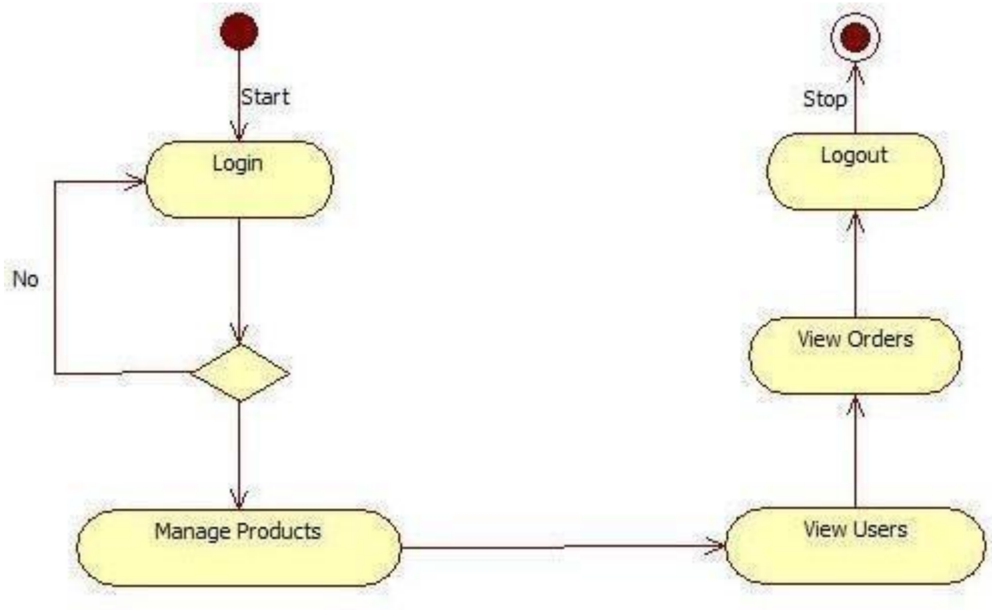


Fig. 3.4 (a) Admin Activity Diagram

The UML Activity Diagram provides a high-level overview of a user's journey within an e-commerce system, capturing the complete workflow from the initial interaction to the termination of the session. The process begins at the start node, where the user is prompted to either register or log in. A decision point then determines the validity of the login or registration attempt. If unsuccessful, the user is redirected to try again, ensuring proper authentication. Upon successful login or registration, the user proceeds to browse product details, organized by categories for easier navigation. The activity continues as users can add selected products to their cart or choose to purchase them immediately. Based on the user's interactions and preferences, the system may also offer personalized product recommendations to enhance the shopping experience. Finally, the user has the option to log out, concluding the session at the stop node. This diagram effectively captures the essential steps of user interaction in an e-commerce environment, offering insights into both user behavior and system design considerations.

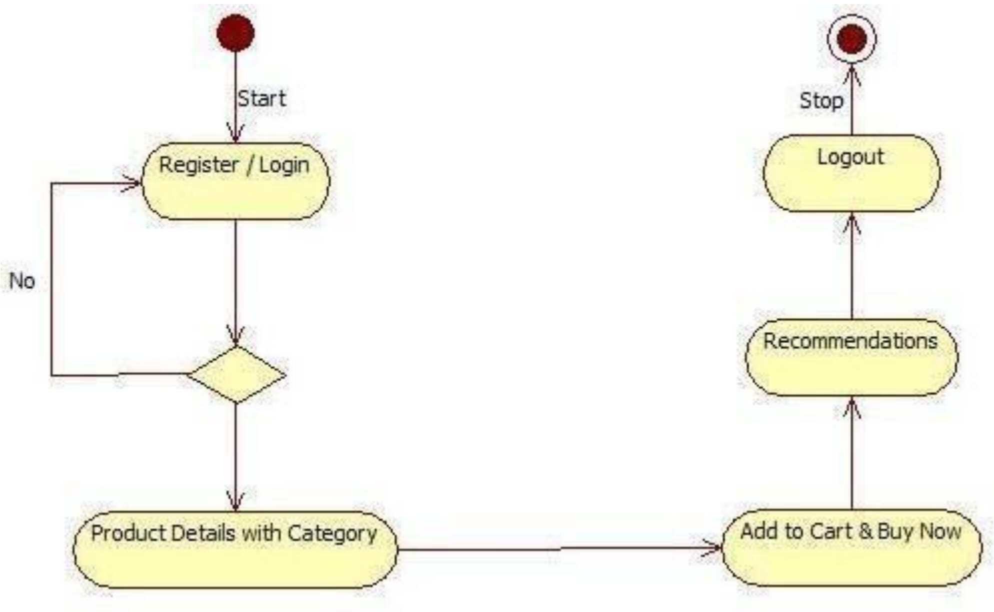


Fig. 3.4(b) User Activity Diagram

The UML Activity Diagram illustrates the workflow of an Admin within an e-commerce system, detailing the sequence of actions from login to logout. The process initiates at the start node, where the Admin is prompted to log in. A decision point evaluates the success of the login attempt—if unsuccessful, the Admin is directed to retry. Upon successful authentication, the Admin gains access to system functionalities, beginning with managing products, which may include adding, updating, or deleting product listings. Following this, the Admin can view details of registered users to oversee platform activity and user management. The workflow continues with the option to view customer orders, enabling the Admin to monitor and manage transactions. The session concludes when the Admin logs out, leading to the stop node that marks the end of the activity. This diagram offers a clear and structured representation of typical administrative operations, making it valuable for designing efficient and user-friendly backend systems or admin interfaces.

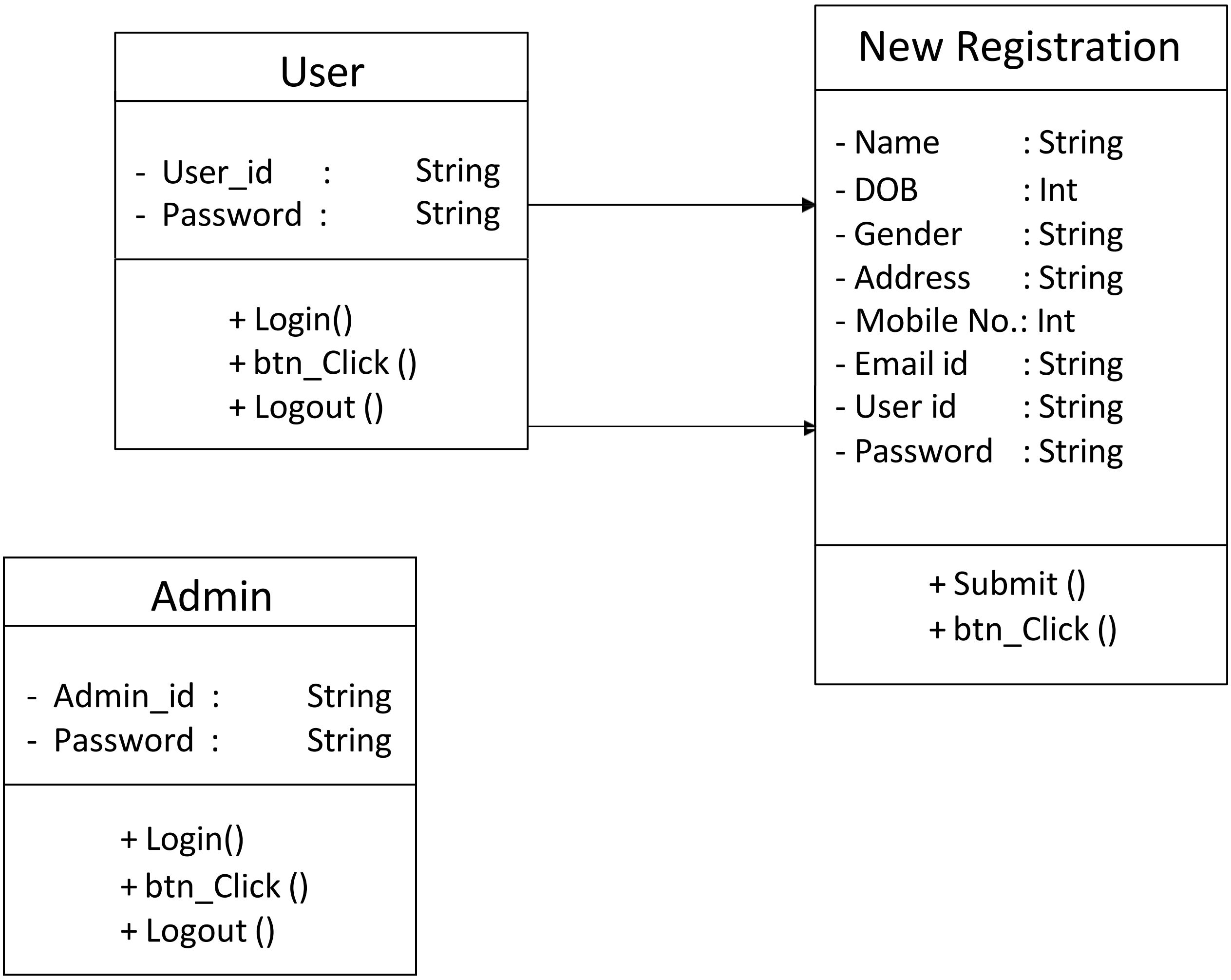


Fig: 3.4(c) class diagram

The image shows a computer screen displaying a UML (Unified Modeling Language) diagram, effectively illustrating the structure and relationships among various components of a software system. The diagram features three main classes—"User," "Admin," and "New Registration"—each represented by a rectangular box containing clearly labeled attributes and methods that define the characteristics and behaviors of the classes. Relationships such as inheritance and composition are also depicted, providing insight into how the classes interact within the system. Rendered in black text on a white background, the diagram maintains a clean and uncluttered appearance, enhancing readability and comprehension. The inclusion of a toolbar at the top suggests that the diagram is part of a broader document or presentation, likely intended for communicating system design to stakeholders or development team members. Overall, the image serves as an effective visual aid for conveying the structural design of a software system in a clear, organized, and accessible manner.

3.5 Data Flow Diagram (DFD’s)

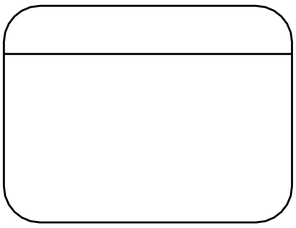
A Data Flow Diagram (DFD) is a visual tool used to illustrate how data moves within a system, helping analysts and developers understand the logical flow of information. It maps out how data is processed from input to output, without referencing the physical setup of the system, making it an essential part of system analysis. This type of diagram, often referred to as a logical DFD, highlights how information flows between various entities such as departments, individuals, or workstations. Two common notations used in creating DFDs are the Yourdon and the Gane-Sarson notations. Each element in the diagram is clearly labeled for clarity, with processes being numbered for easy identification. The design process for DFDs is done in multiple layers, starting with the top-level diagram known as the context diagram. This overview includes a single process that represents the entire system and is later expanded into more detailed levels, such as the Level 1 DFD.

This layered approach continues until enough detail is gathered to understand the system fully. The concept of DFDs was first introduced by Larry Constantine as a method for graphically representing system requirements, contributing significantly to modular system design. Often called “bubble charts,” DFDs play a crucial role in both analyzing and designing systems by outlining significant data transformations. The four basic symbols used in a DFD include a square to represent external entities (sources or destinations of data), an arrow to show the flow of data, a circle or bubble to denote processes that transform data, and an open-ended rectangle to indicate data stores where information is held temporarily or permanently. Overall, DFDs provide a clear, structured, and logical view of how data flows and is managed within a system, serving as a foundation for designing and understanding information systems.

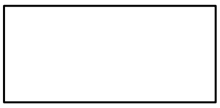
3.5.1DFD SYMBOLS:

There are four symbols in the DFD

- Square – Represents an external entity, which could be the source of incoming data or the destination where data is sent outside the system.
- Arrow – Indicates the flow of data between different parts of the system, showing how and where information moves.
- Circle or Bubble – Depicts a process that takes input data, performs a transformation or operation, and produces output data.
- Open-ended Rectangle – Represents a data store, where information is either temporarily held or stored for future use within the system.

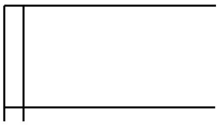


Process that transforms data flow.



Source or Destination of data

Data flow



Data Store

3.5.1 CONSTRUCTING A DFD:

When designing a Data Flow Diagram (DFD), there are several key guidelines that help ensure clarity and consistency. Every process in the diagram should be given a meaningful name and assigned a unique number, making it easy to identify and reference. The general flow of data is shown from top to bottom and left to right, representing the natural movement of information from sources to destinations. However, if data needs to loop back to a source, this can be shown with a long backward arrow or by duplicating the source symbol and marking it with a small diagonal line to indicate it appears more than once.

As processes are broken down into more detailed levels, each sub-process should also be numbered to maintain a clear structure. For naming conventions, data stores and external entities are written in all capital letters, while process names and data flow labels use capital letters at the beginning of each word to improve readability. DFDs should also show only the essential contents of each data store, making sure all data elements going in and out are accurately represented. To avoid missing any data elements or interfaces, tools like questionnaires should be thorough. If there are gaps or redundancies, these are usually addressed through interviews or further investigation during the analysis phase.

3. 5.3 BUILDING OF DFD:

When developing Data Flow Diagrams (DFDs), it’s important to follow certain guidelines to ensure they are clear, structured, and easy to understand. Each process in the diagram should have a descriptive title and a unique number to make it easy to identify and reference. The standard flow of data moves from top to bottom and left to right, representing a natural, logical sequence. While data usually flows from a source to a destination, it can sometimes loop back to the source. This can be shown either by drawing a long arrow back to the original source or by duplicating the source symbol and marking it with a small diagonal line to indicate it's a repeated element. When a process is broken down into more detailed parts—known as a process explosion—the lower-level components must also be numbered accordingly. Naming conventions are also important: names of data stores and external destinations are written in uppercase letters, while names of processes and data flows begin with capital letters for each word. Each data store in the DFD should contain only the essential data elements, and it must accurately reflect all inputs and outputs. To ensure completeness, tools like questionnaires should include all incoming and outgoing data elements. If any details are missed or if there are redundancies, these are usually identified and resolved through follow-up interviews or further system analysis.

3.5.4 TYPES OF DATA FLOW DIAGRAMS

Current Physical

This type represents how the current system operates in a physical sense. It includes the names of people, positions, or computer systems involved in processing, as well as the specific technologies used. Data stores and flows are often labeled with the names of physical media such as paper files, computer databases, forms, or storage devices like tapes.

Current Logical

In this version, physical elements are minimized or removed. The focus is on the essential processes and data flows within the existing system, regardless of the actual hardware or software used. It highlights what the system does rather than how it does it.

New Logical

This type outlines an improved version of the logical system. It's typically used when users are satisfied with what the system does but not with how it functions. The new logical model may introduce additional features, remove redundant processes, or optimize data flows for better performance and efficiency.

New Physical

This diagram provides a visual representation of how the new system will physically function. It details the actual implementation, including new technologies, devices, and infrastructure that willbe part of the redesigned system.

Rules for Designing DFDs

To maintain accuracy and clarity in your DFDs, certain guidelines must be followed:

- **Process Rules:**

A process must have both inputs and outputs; it cannot exist with only one of them.

If a component only receives data and does nothing else, it is considered a sink, not a process.

Each process should be labeled using a verb phrase, indicating the action it performs (e.g., “Process Order”).

- **Data Store Rules:**

Data cannot move directly from one data store to another. A process must act as a bridge between them.

Likewise, data cannot flow straight from a source to a data store; it must first pass through a process.

Every data store should be labeled using a noun phrase, describing the type of data stored (e.g., “Customer Records”).

- **Source or Sink Rules:**

Sources and sinks represent points where data enters or exits the system.

Data cannot flow directly from a source to a sink without being processed. A process must handle the data in between.

These elements are labeled with noun phrases, such as “Customer” or “Inventory System”.

- **Data Flow Rules:**

Data flows only in one direction between components, even though two-way communication is possible between a process and a data store (e.g., read and update operations). When both actions occur, they are typically shown with two separate arrows.

A data flow cannot immediately loop back into the process it came from. If data needs to return to the original process, it must first go through at least one other process.

A data flow directed into a data store usually represents an update, such as adding, deleting, or modifying records.

When multiple processes or components send the same data to one destination, a join can be shown, indicating that any of those sources can deliver the data.

3.6System Architecture

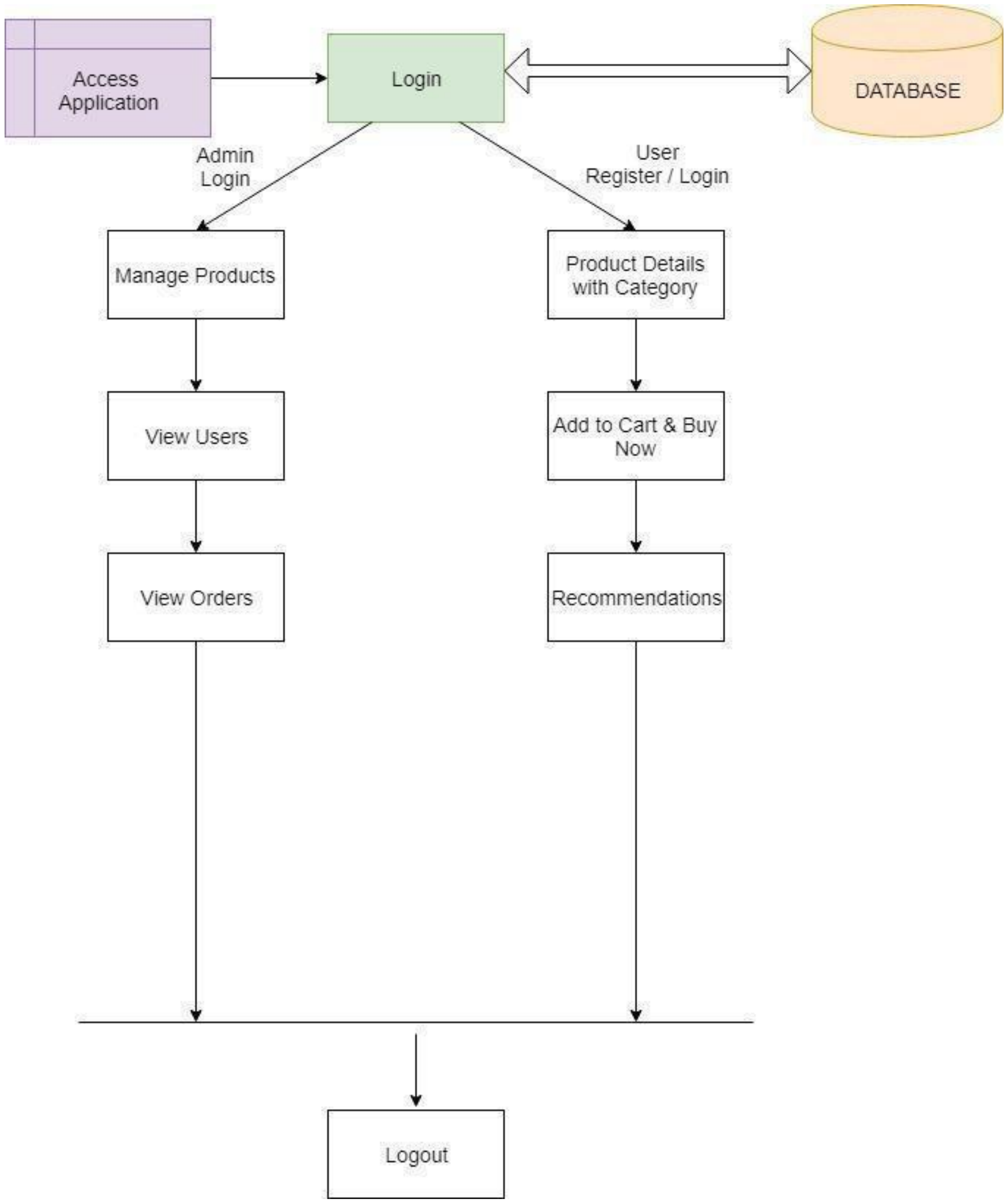


Fig 3.6 System Architecture

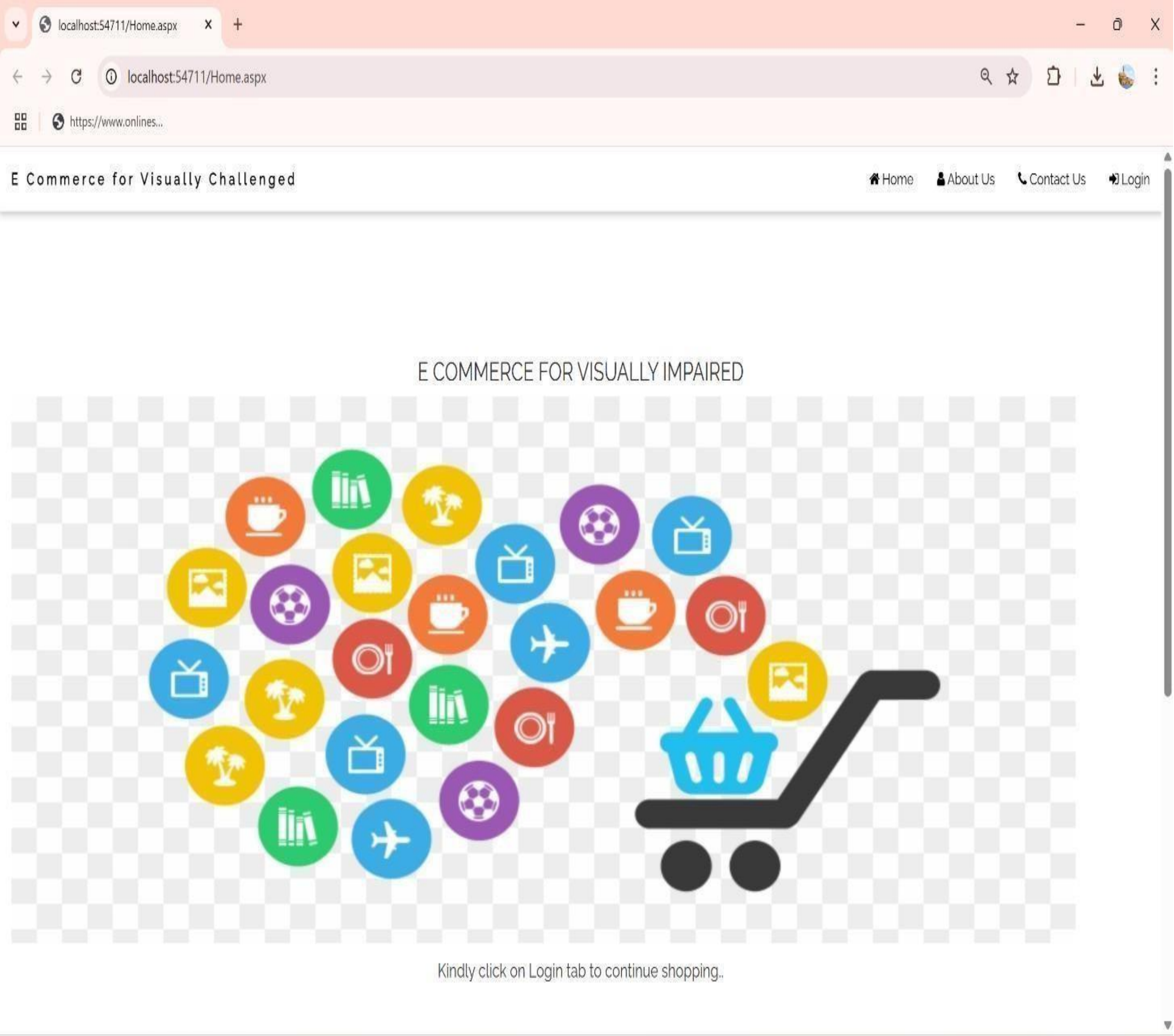
The image presents a system architecture or flowchart diagram that visually outlines the functional flow of an e-commerce web application, specifically distinguishing between Admin and User roles. The process begins with users or admins accessing the application and proceeding through a centralized login module that communicates with the database for authenticating credentials. After logging in, the flow splits into two different routes. Admins have access to features like product management—adding, editing, or deleting listings—listing details of registered users, and order management for customers. At the same time, frequent users can view product information sorted by type, place items in the cart, buy products via the "Buy Now" feature, and get product recommendations tailored to their activity or preference. Both groups terminate their sessions with a logout action. The diagram clearly conveys the double-role process, focusing on integration of the system, logical order, and interaction with users, and therefore is a useful element for system design documentation or Software Requirements Specification (SRS) reports.

Summary

Chapter 3 presents a detailed design and modeling framework for an e-commerce web application, focusing on both user and admin roles. It begins with an Entity-Relationship (E-R) diagram that defines key entities such as User, Admin, and Product, along with their attributes and relationships, highlighting how users register, log in, browse products, and how admins manage products and monitor user activity [3]. The use case diagrams further illustrate the specific interactions each actor (user and admin) has with the system, such as login, product management, and order viewing for admins, and product browsing, cart actions, and personalized recommendations for users [4]. The sequence diagram describes the step-by-step communication flow between the admin, the system, and the database, emphasizing processes like login, managing products, and viewing data. Activity diagrams for both users and admins visualize the overall workflow, from authentication to completing tasks like purchases or administrative management, ending with logout [4]. The class diagram shows the static structure of the system using classes like User and Admin, their properties, methods, and relationships, offering insight into the object-oriented structure of the application [9]. Data Flow Diagrams (DFDs) are introduced to map the logical and physical movement of data through the system. The chapter explains DFD symbols, construction rules, types (current and new logical/physical models), and key principles governing data flow and process management [6]. Finally, the system architecture diagram provides a comprehensive visual flow of the entire application, distinguishing between user and admin paths, starting from login to task execution and logout, and showcasing system integration and functional separation. Overall, the chapter offers a complete visualization of the system’s design, ensuring clarity in functionality, data handling, and role-based access within the e-commerce platform [3][4][6][9].

Chapter 4

Snapshots


localhost:54711/Home.aspx
localhost:54711/Home.aspx
https://www.onlines...
E Commerce for Visually Challenged
Home
About Us
Contact Us
Login

E COMMERCE FOR VISUALLY IMPAIRED

Kindly click on Login tab to continue shopping..

Fig 4.1 This is the Main Display Page of E-Commerce for Visually Impaired

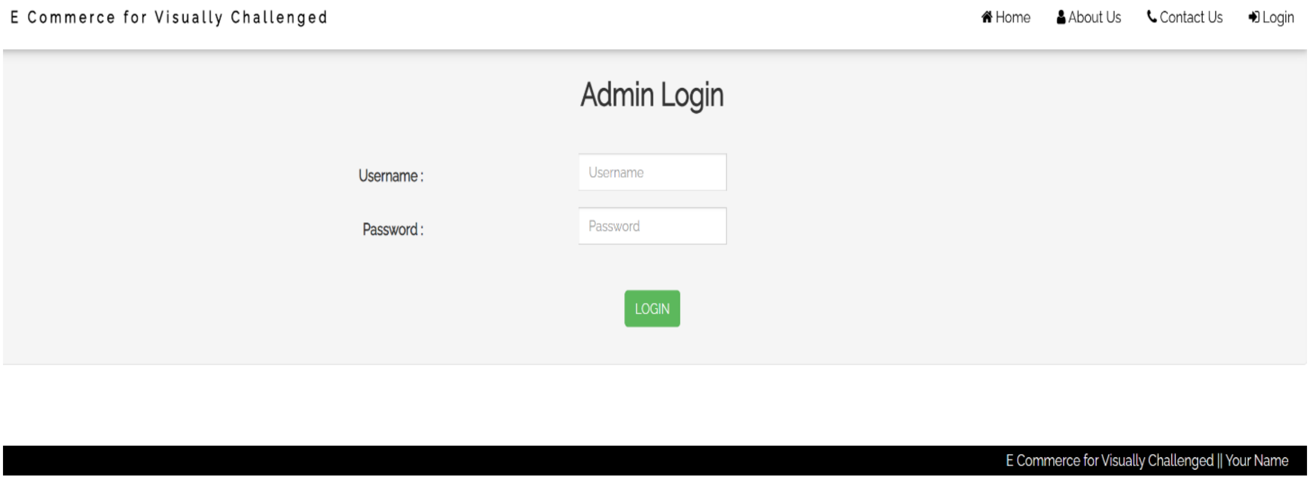

E Commerce for Visually Challenged
Home
About Us
Contact Us
Login
Admin Login
Username :
Username
Password :
Password
LOGIN
E Commerce for Visually Challenged || Your Name

Fig 4.2 This is the Admin Login page

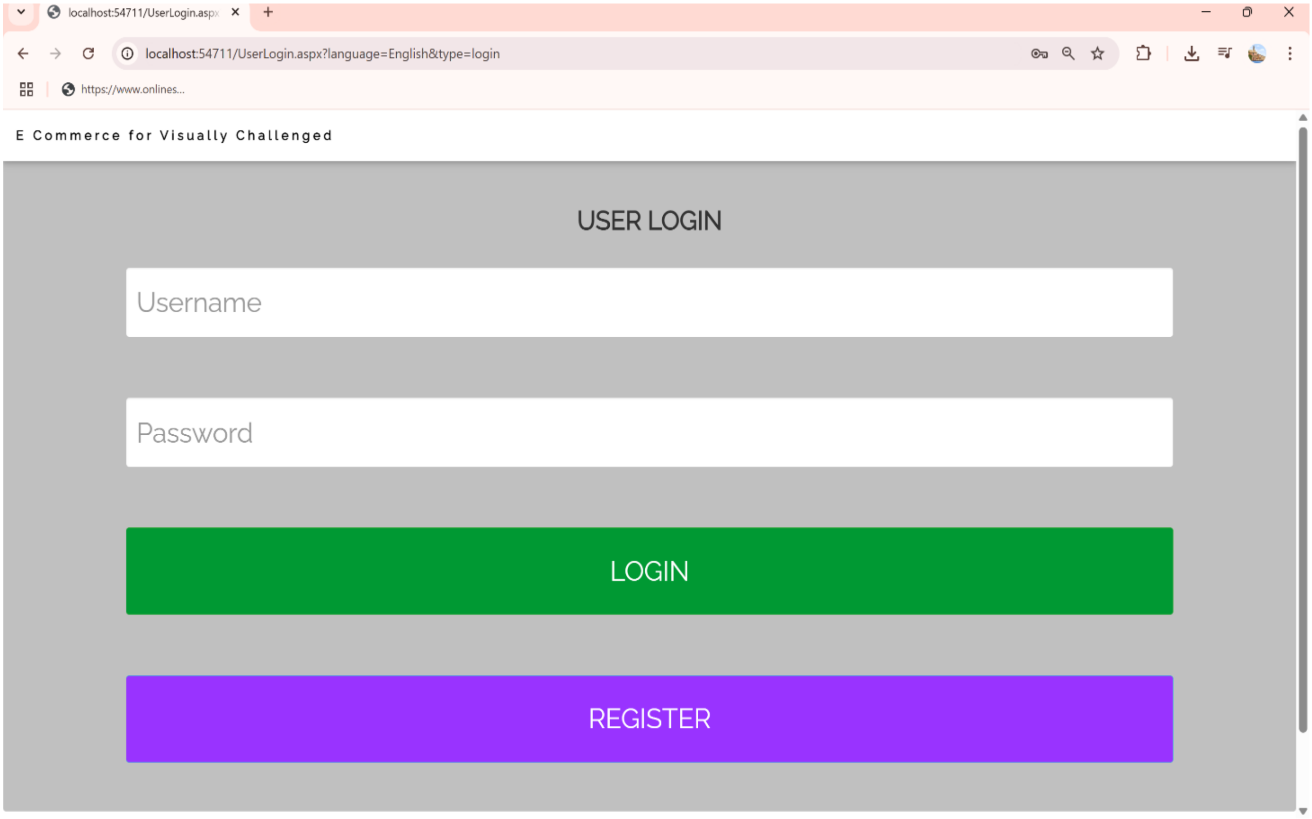
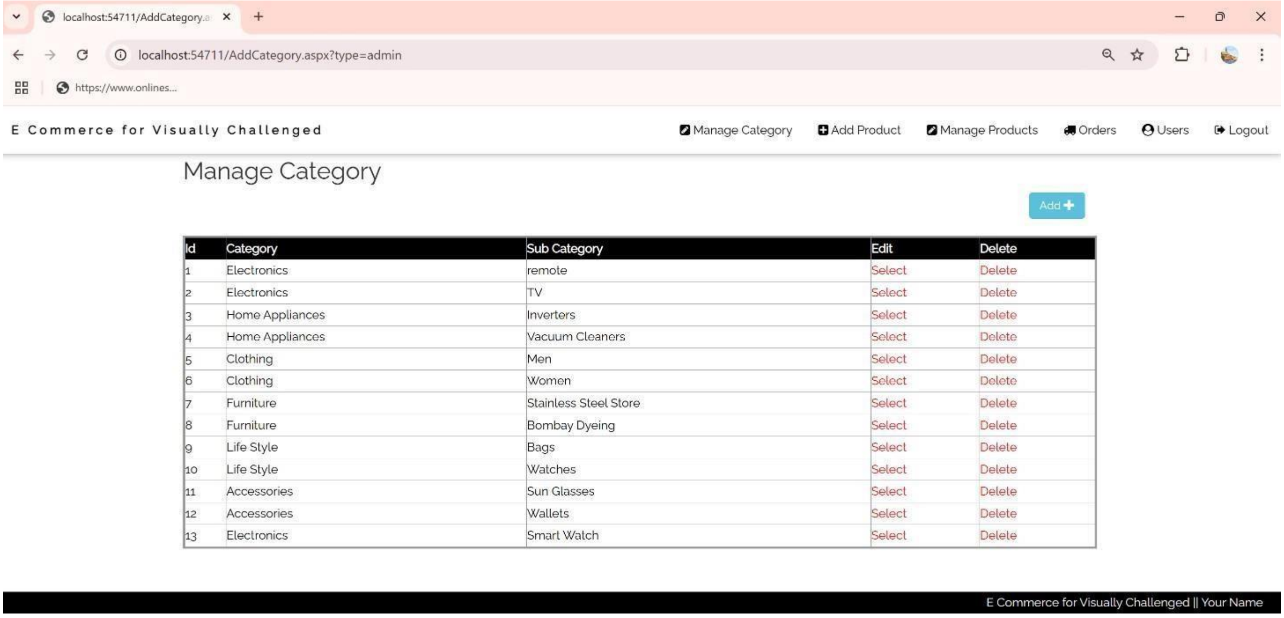

localhost:54711/UserLogin.aspx
localhost:54711/UserLogin.aspx?language=English&type=login
https://www.onlines...
E Commerce for Visually Challenged
USER LOGIN
Username
Password
LOGIN
REGISTER

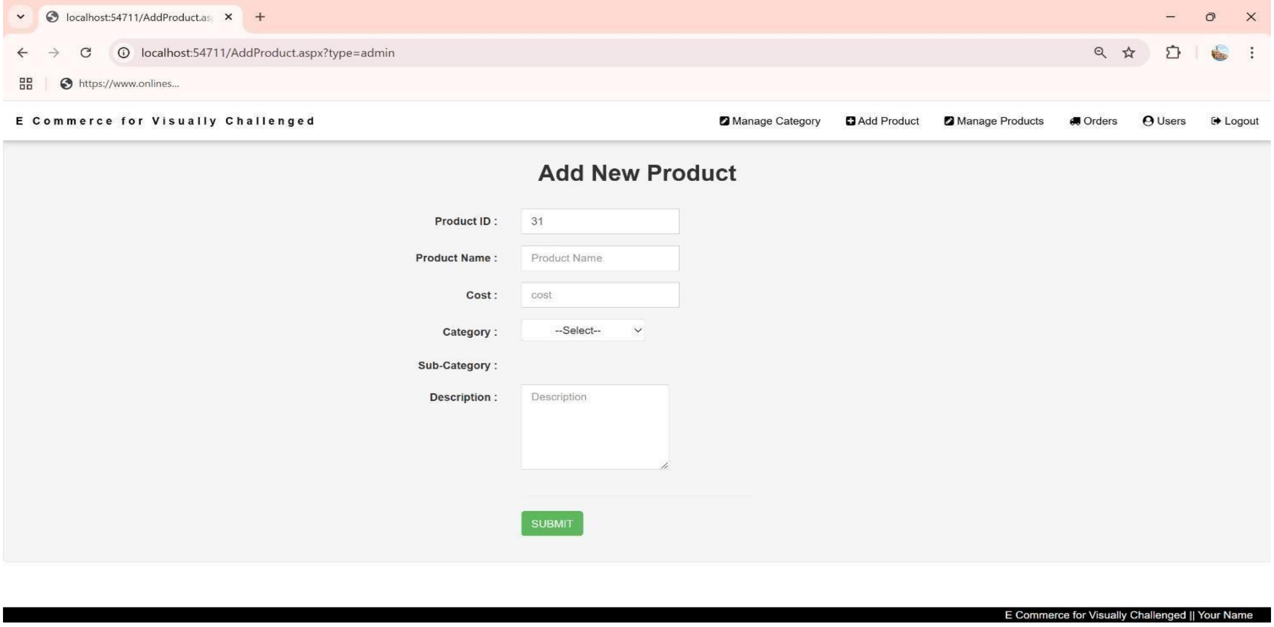
Fig 4.3 This is the User Login page


localhost:54711/AddCategory.a
localhost:54711/AddCategory.aspx?type=admin
https://www.onlines...
E Commerce for Visually Challenged
Manage Category
Add Product
Manage Products
Orders
Users
Logout
Add
E Commerce for Visually Challenged || Your Name

Manage Category

Id	Category	Sub Category	Edit	Delete
1	Electronics	remote	Select	Delete
2	Electronics	TV	Select	Delete
3	Home Appliances	Inverters	Select	Delete
4	Home Appliances	Vacuum Cleaners	Select	Delete
5	Clothing	Men	Select	Delete
6	Clothing	Women	Select	Delete
7	Furniture	Stainless Steel Store	Select	Delete
8	Furniture	Bombay Dyeing	Select	Delete
9	Life Style	Bags	Select	Delete
10	Life Style	Watches	Select	Delete
11	Accessories	Sun Glasses	Select	Delete
12	Accessories	Wallets	Select	Delete
13	Electronics	Smart Watch	Select	Delete

Fig 4.4 This is the List of all the products in the Manage Category page


localhost:54711/AddProduct.as
localhost:54711/AddProduct.aspx?type=admin
https://www.onlines...
E Commerce for Visually Challenged
Manage Category
Add Product
Manage Products
Orders
Users
Logout
SUBMIT
E Commerce for Visually Challenged || Your Name

Add New Product

Product ID : 31

Product Name : Product Name

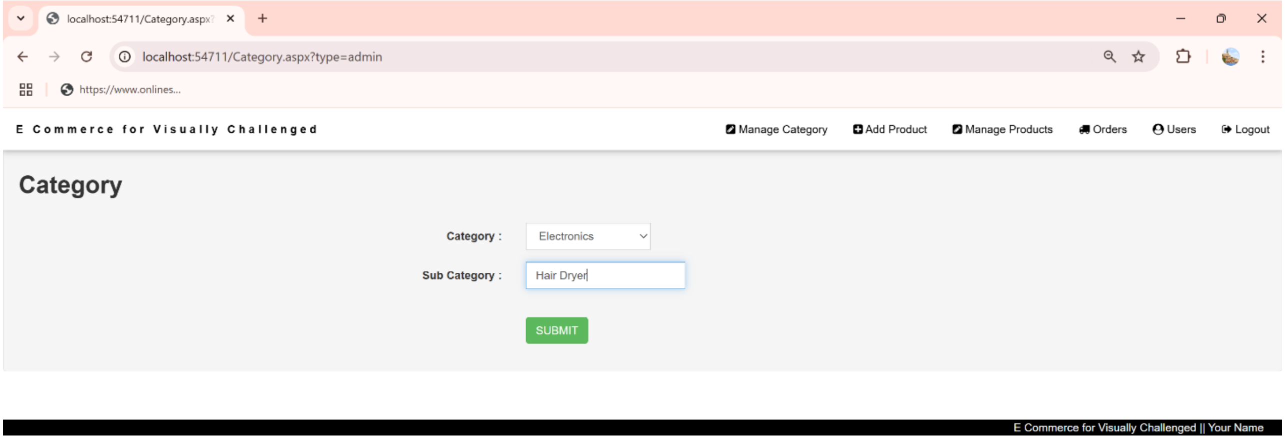
Cost : cost

Category : --Select--

Sub-Category :

Description : Description

Fig 4.5 This is the Add New Product page

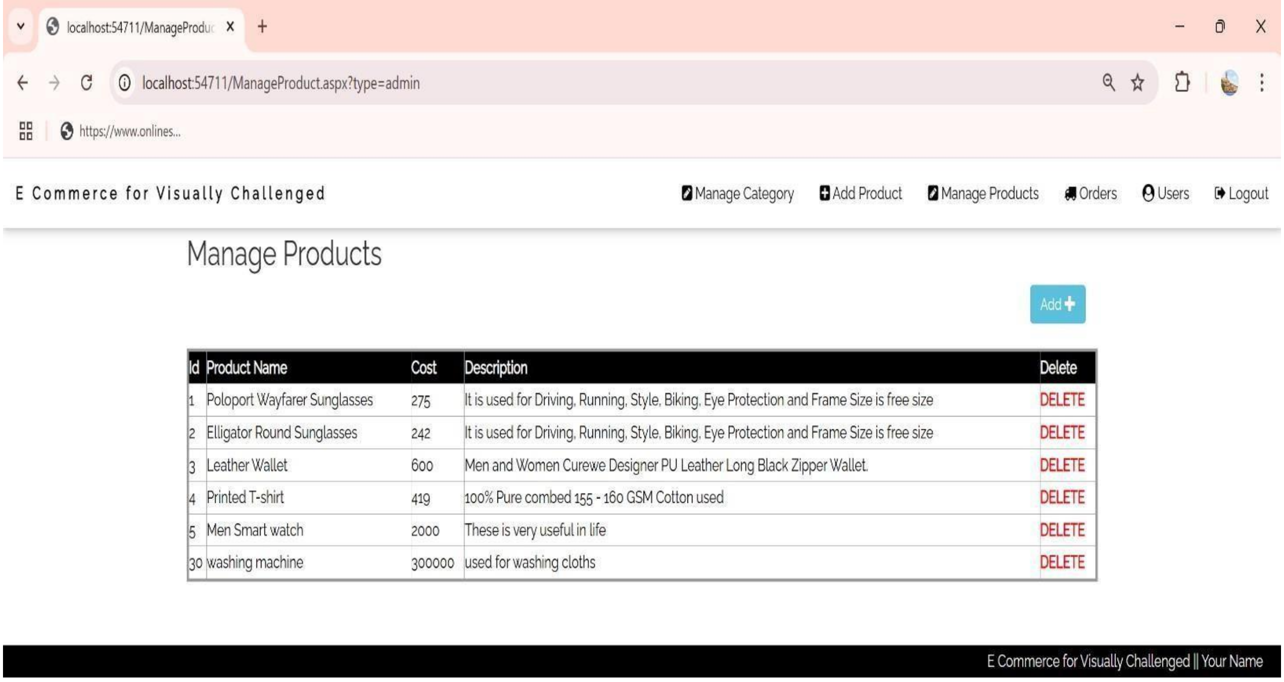

localhost:54711/Category.aspx
localhost:54711/Category.aspx?type=admin
https://www.onlines...
E Commerce for Visually Challenged
Manage Category
Add Product
Manage Products
Orders
Users
Logout
SUBMIT
E Commerce for Visually Challenged || Your Name

Category

Category : Electronics

Sub Category : Hair Dryer

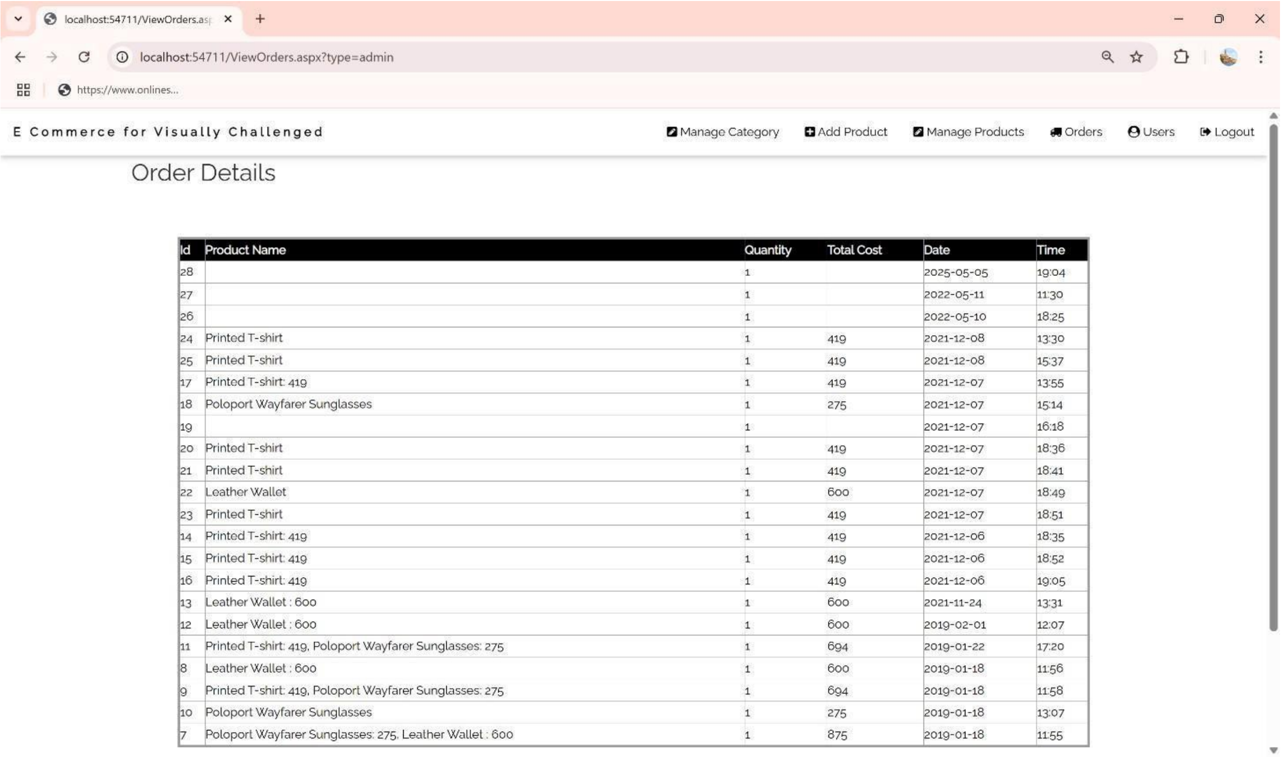
Fig 4.6 This is the Add New Product in particular Sub Category page


localhost:54711/ManageProduc
localhost:54711/ManageProduct.aspx?type=admin
https://www.onlines...
E Commerce for Visually Challenged
Manage Category
Add Product
Manage Products
Orders
Users
Logout
Add
E Commerce for Visually Challenged || Your Name

Manage Products

Id	Product Name	Cost	Description	Delete
1	Poloport Wayfarer Sunglasses	275	It is used for Driving, Running, Style, Biking, Eye Protection and Frame Size is free size	DELETE
2	Elligator Round Sunglasses	242	It is used for Driving, Running, Style, Biking, Eye Protection and Frame Size is free size	DELETE
3	Leather Wallet	600	Men and Women Curewe Designer PU Leather Long Black Zipper Wallet.	DELETE
4	Printed T-shirt	419	100% Pure combed 155 - 160 GSM Cotton used	DELETE
5	Men Smart watch	2000	These is very useful in life	DELETE
30	washing machine	300000	used for washing cloths	DELETE

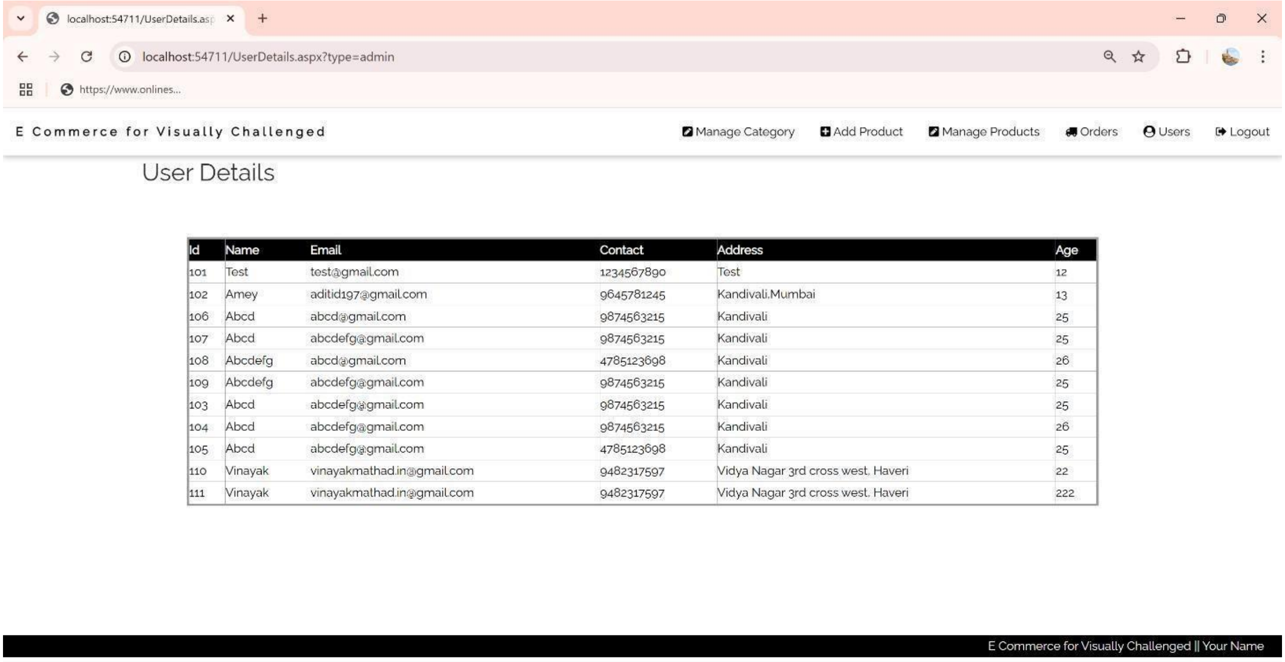
Fig 4.7 This is the Manage Product page


localhost:54711/ViewOrders.asp
localhost:54711/ViewOrders.aspx?type=admin
https://www.onlines...
E Commerce for Visually Challenged
Manage Category
Add Product
Manage Products
Orders
Users
Logout

Order Details

Id	Product Name	Quantity	Total Cost	Date	Time
28		1		2025-05-05	19:04
27		1		2022-05-11	11:30
26		1		2022-05-10	18:25
24	Printed T-shirt	1	419	2021-12-08	13:30
25	Printed T-shirt	1	419	2021-12-08	15:37
17	Printed T-shirt: 419	1	419	2021-12-07	13:55
18	Poloport Wayfarer Sunglasses	1	275	2021-12-07	15:14
19		1		2021-12-07	16:18
20	Printed T-shirt	1	419	2021-12-07	18:36
21	Printed T-shirt	1	419	2021-12-07	18:41
22	Leather Wallet	1	600	2021-12-07	18:49
23	Printed T-shirt	1	419	2021-12-07	18:51
14	Printed T-shirt: 419	1	419	2021-12-06	18:35
15	Printed T-shirt: 419	1	419	2021-12-06	18:52
16	Printed T-shirt: 419	1	419	2021-12-06	19:05
13	Leather Wallet : 600	1	600	2021-11-24	13:31
12	Leather Wallet : 600	1	600	2019-02-01	12:07
11	Printed T-shirt: 419, Poloport Wayfarer Sunglasses: 275	1	694	2019-01-22	17:20
8	Leather Wallet : 600	1	600	2019-01-18	11:56
9	Printed T-shirt: 419, Poloport Wayfarer Sunglasses: 275	1	694	2019-01-18	11:58
10	Poloport Wayfarer Sunglasses	1	275	2019-01-18	13:07
7	Poloport Wayfarer Sunglasses: 275, Leather Wallet : 600	1	875	2019-01-18	11:55

Fig 4.8 This is the Order Details of all the users


localhost:54711/UserDetails.asp
localhost:54711/UserDetails.aspx?type=admin
https://www.onlines...
E Commerce for Visually Challenged
Manage Category
Add Product
Manage Products
Orders
Users
Logout
E Commerce for Visually Challenged || Your Name

User Details

Id	Name	Email	Contact	Address	Age
101	Test	test@gmail.com	1234567890	Test	12
102	Amey	aditid197@gmail.com	9645781245	Kandivali,Mumbai	13
106	Abcd	abcd@gmail.com	9874563215	Kandivali	25
107	Abcd	abcdefg@gmail.com	9874563215	Kandivali	25
108	Abcdefg	abcd@gmail.com	4785123698	Kandivali	26
109	Abcdefg	abcdefg@gmail.com	9874563215	Kandivali	25
103	Abcd	abcdefg@gmail.com	9874563215	Kandivali	25
104	Abcd	abcdefg@gmail.com	9874563215	Kandivali	26
105	Abcd	abcdefg@gmail.com	4785123698	Kandivali	25
110	Vinayak	vinayakmathad.in@gmail.com	9482317597	Vidya Nagar 3rd cross west, Haveri	22
111	Vinayak	vinayakmathad.in@gmail.com	9482317597	Vidya Nagar 3rd cross west, Haveri	222

Fig 4.9 This is the Users Details of the different user’s page

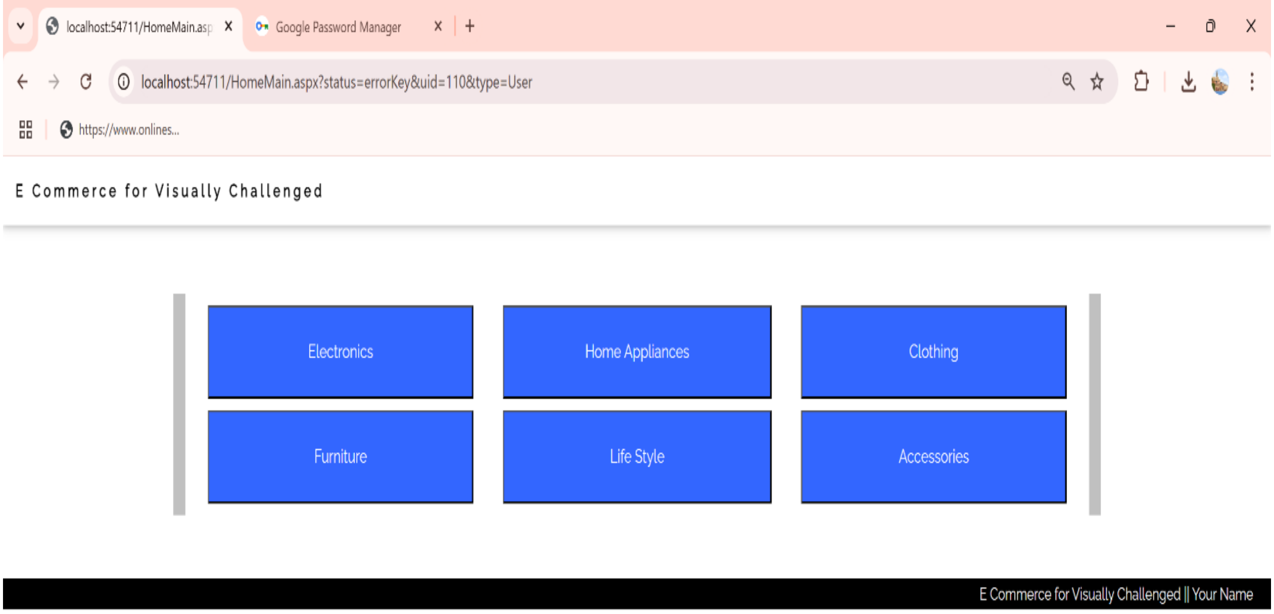

localhost:54711/HomeMain.asp
Google Password Manager
localhost:54711/HomeMain.aspx?status=errorKey&uid=110&type=User
https://www.onlines...
E Commerce for Visually Challenged
Electronics
Home Appliances
Clothing
Furniture
Life Style
Accessories
E Commerce for Visually Challenged || Your Name

Fig 4.10 This is the List of all the products which is further divided into many Sub Category page

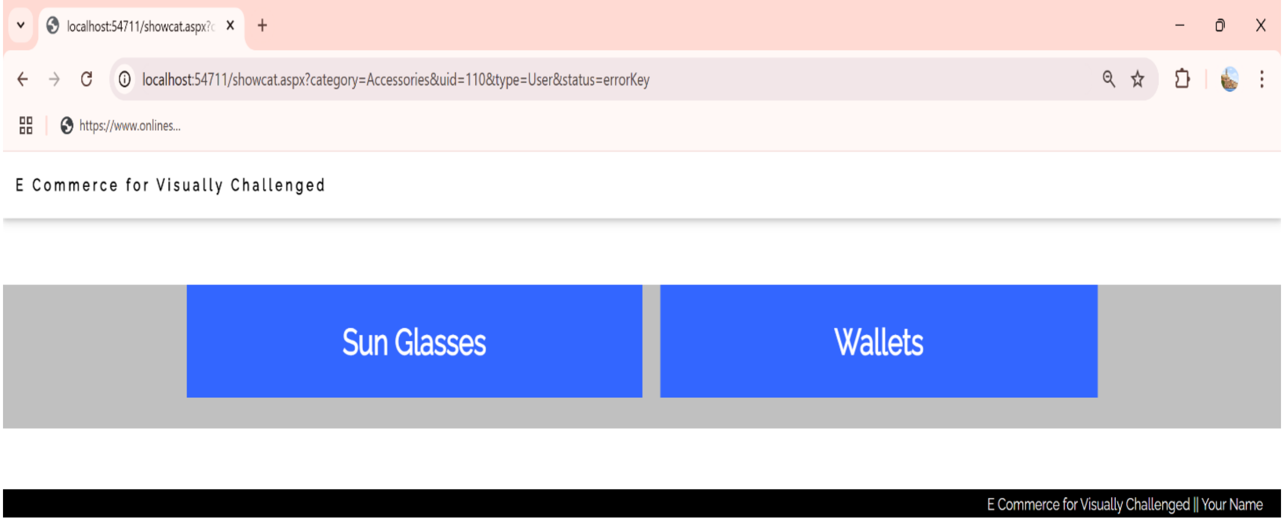
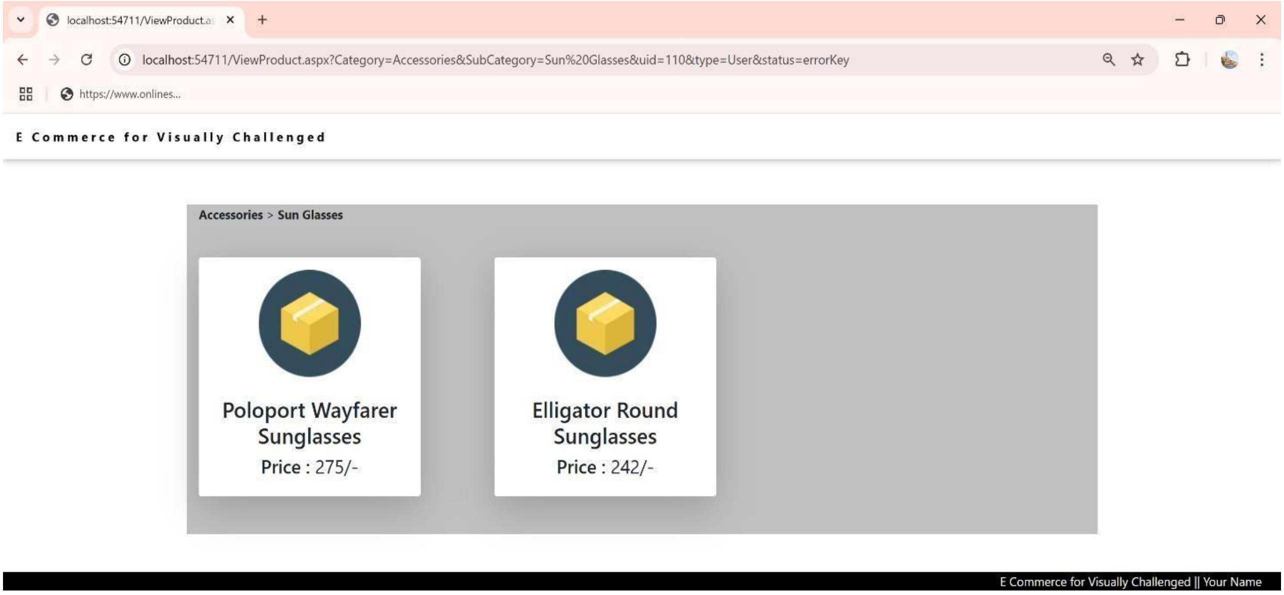

localhost:54711/showcat.aspx?c
localhost:54711/showcat.aspx?category=Accessories&uid=110&type=User&status=errorKey
https://www.onlines...
E Commerce for Visually Challenged
Sun Glasses
Wallets
E Commerce for Visually Challenged || Your Name

Fig 4.11 This is the Sub Category List inside the Main Category List



localhost:54711/ViewProduct.aspx?Category=Accessories&SubCategory=Sun%20Glasses&uid=110&type=User&status=errorKey

E Commerce for Visually Challenged

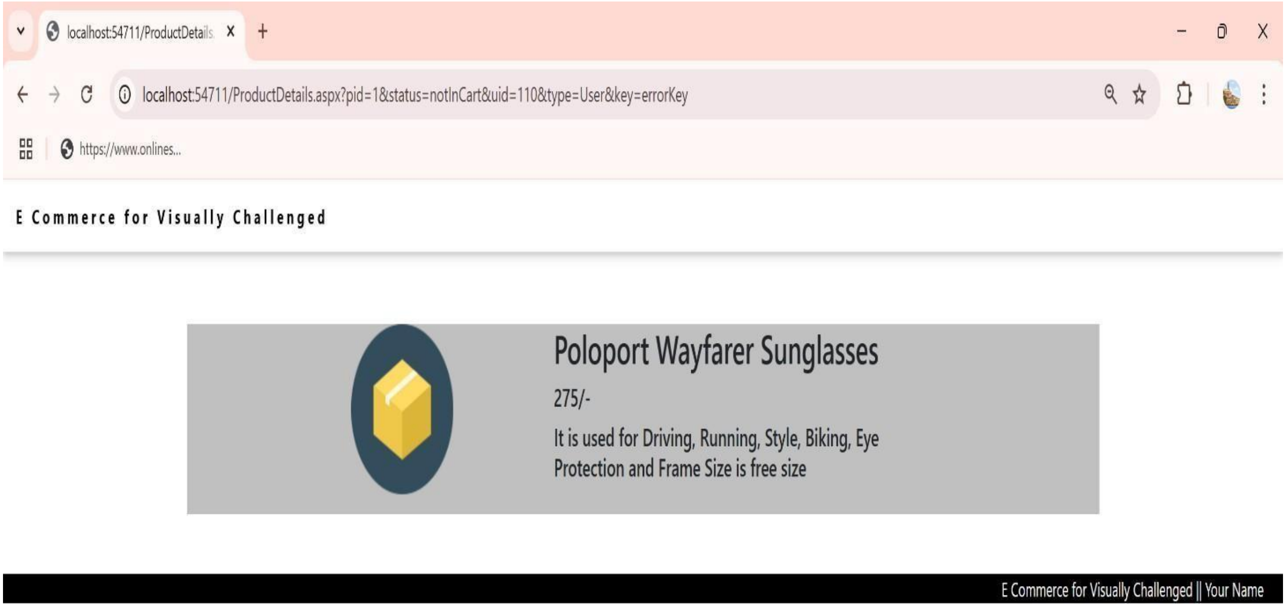
Accessories > Sun Glasses

Poloport Wayfarer Sunglasses
Price : 275/-

Elligator Round Sunglasses
Price : 242/-

E Commerce for Visually Challenged || Your Name

Fig 4.12 These are the Different Products of the Accessories



localhost:54711/ProductDetails.aspx?pid=1&status=notInCart&uid=110&type=User&key=errorKey

E Commerce for Visually Challenged

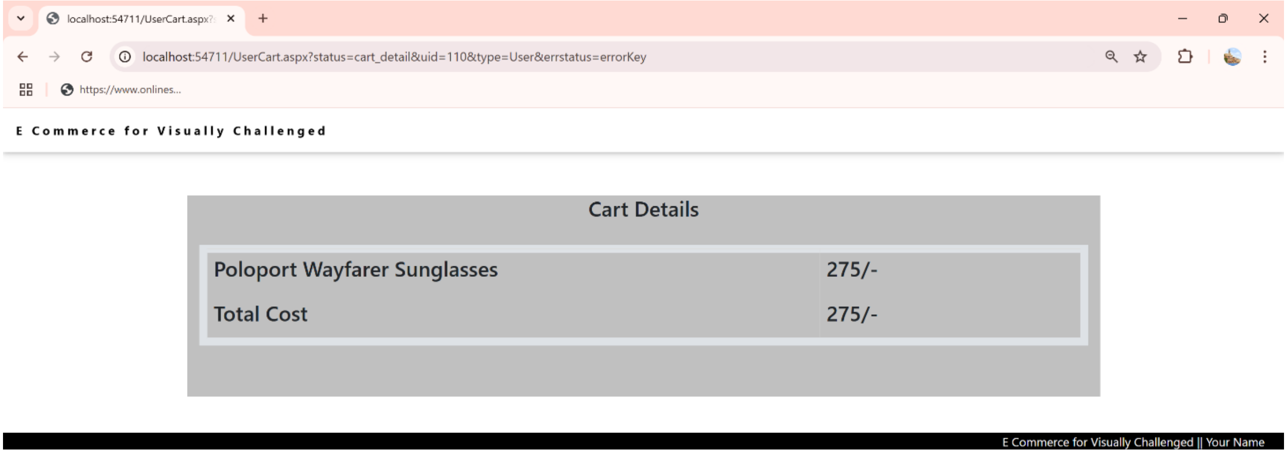
Poloport Wayfarer Sunglasses

275/-

It is used for Driving, Running, Style, Biking, Eye Protection and Frame Size is free size

E Commerce for Visually Challenged || Your Name

Fig 4.13 This is the Description of the Products

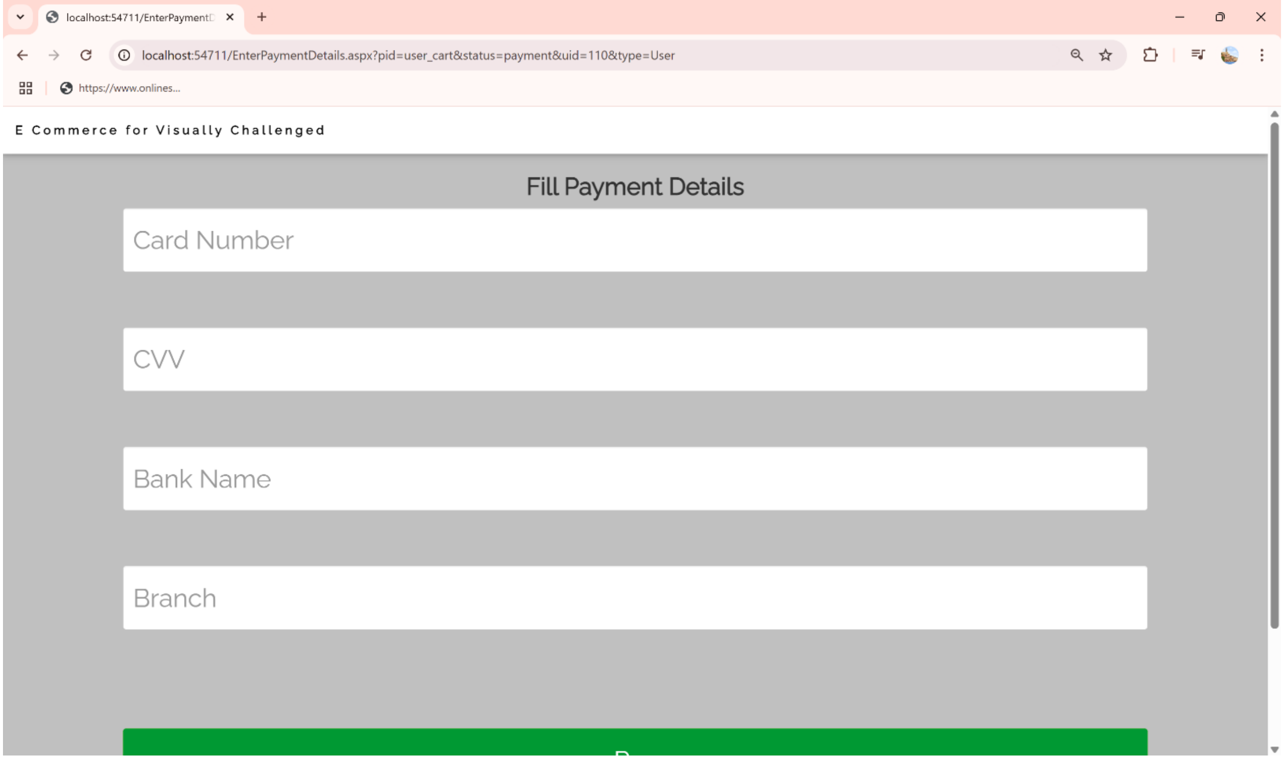

localhost:54711/UserCart.aspx?
localhost:54711/UserCart.aspx?status=cart_detail&uid=110&type=User&errstatus=errorKey
https://www.onlines...
E Commerce for Visually Challenged

Cart Details

Poloport Wayfarer Sunglasses	275/-
Total Cost	275/-

E Commerce for Visually Challenged || Your Name

Fig 4.14 This is the Adding of the Product to Cart


localhost:54711/EnterPaymentD
localhost:54711/EnterPaymentDetails.aspx?pid=user_cart&status=payment&uid=110&type=User
https://www.onlines...
E Commerce for Visually Challenged

Fill Payment Details

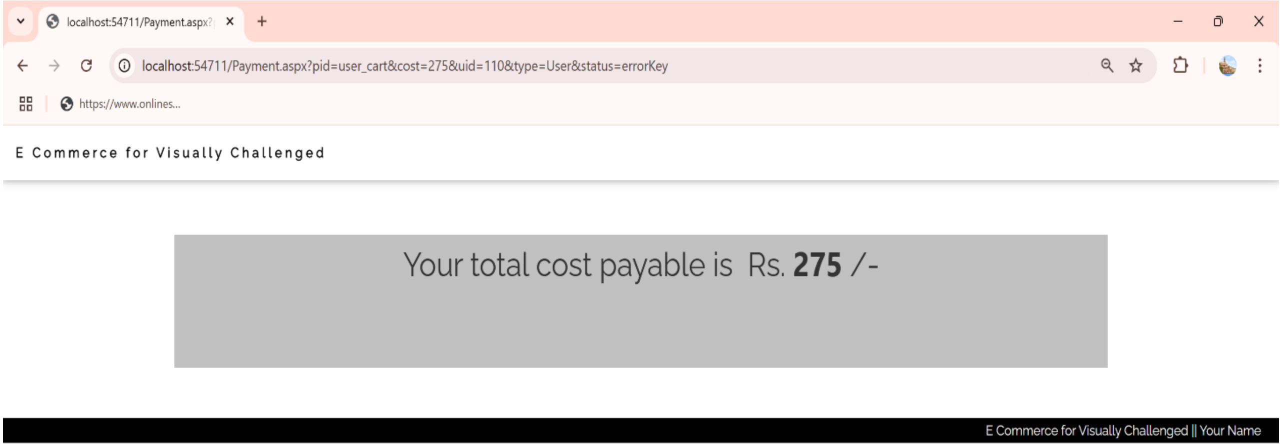
Card Number

CVV

Bank Name

Branch

Fig 4.15 This page is about entering the Card Details for Payment Method



localhost:54711/Payment.aspx?
localhost:54711/Payment.aspx?pid=user_cart&cost=275&uid=110&type=User&status=errorKey
https://www.onlines...

E Commerce for Visually Challenged

Your total cost payable is Rs. **275** /-

E Commerce for Visually Challenged || Your Name

Fig 4.15 This page is about total amount Payable from Card

Chapter 5

Project Implementation

5.1 Project Implementation Technology

The Project is designed and developed in Visual Studio. We used Visual Studio for coding of the project. Created and maintained all databases into SQL Server, in that we create tables, write query for store data or record of project.

❖ **Hardware Requirement:**

- ➢ i3 Processor Based Computer or higher
- ➢ Memory: 1 GB
- ➢ Hard Drive: 50 GB
- ➢ Monitor
- ➢ Internet Connection

❖ **Software Requirement:**

- ➢ Windows 7 or higher
- ➢ Visual Studio
- ➢ SQL Server
- ➢ Google Chrome Browser

5.2 OVERVIEW OF TECHNOLOGIES USED

1. Front End Technology

a)Microsoft .NET Framework

The Microsoft .NET Framework is a powerful development platform created to simplify application development in today’s distributed and Internet-connected world. It provides a consistent and unified programming model for building various types of applications, whether they are desktop-based, web-based, or distributed over a network.

One of its key strengths is its ability to support object-oriented programming across different environments. Whether your application runs locally on a user’s device or is hosted and accessed over the Internet, .NET provides a seamless experience for developers.

The platform also addresses common challenges in software development, such as deployment issues and version conflicts. By offering a managed runtime environment, it ensures that applications run securely—even if the code comes from less trusted sources. This environment also enhances performance by avoiding the typical drawbacks of scripting or interpreted languages.

Developers benefit from a consistent experience across various types of applications. Whether you’re building a traditional Windows application or a modern web application, .NET provides the tools and structure needed. It also supports industry standards, making it easier to integrate applications written in .NET with other technologies.

The .NET Framework is made up of two main components:

Common Language Run Time

(CLR)

This is the core engine of the .NET Framework. It handles critical tasks such as memory management, thread execution, and error handling. The CLR ensures that applications run securely and efficiently. Code that runs under the CLR is known as *managed code*, whereas code that doesn’t is referred to as *unmanaged code*.

.**NET FRAMEWORK CLASS LIBRARY**

(NFCL)

This is a comprehensive collection of reusable classes and types that developers can use to build their applications. It supports a wide range of programming needs—from creating simple command-line apps to building advanced graphical user interfaces and web services using technologies like ASP.NET.

One of the unique features of the .NET Framework is its flexibility in working with unmanaged components. This means that existing code written in older or different programming models can still interact with the .NET environment. For example, ASP.NET is a web application framework that hosts the CLR, allowing developers to build scalable, server-side web applications using managed code.

the .NET Framework, another powerful capability it offers is the ability to host the runtime within unmanaged applications. A good example is Internet Explorer, which can act as a host through a MIME type extension. This setup enables developers to embed managed components, such as Windows Forms controls, directly into HTML pages. Essentially, this means you can include interactive .NET components on a webpage and have them run within the browser.

Unlike traditional methods such as ActiveX controls, which often pose security risks, managed components in this setup benefit from the .NET Framework’s enhanced security features. These include controlled execution environments and isolated file storage. This ensures that even when code runs from the web, it operates in a semi-trusted mode, preventing unauthorized access to sensitive parts of the user’s system.

A visual representation (as referred to in the original documentation) shows how the Common Language Runtime (CLR) and the .NET class library are interconnected with applications and the broader system architecture. It also highlights where managed code fits into the picture, showcasing its role within the .NET ecosystem.

a) Features of the Common Language Runtime (CLR)

The Common Language Runtime is the heart of the .NET Framework. It takes care of several core system functions automatically, such as memory allocation, thread management, and code execution. Additionally, it verifies the safety of the code before running it and handles its compilation into native machine code.

A major benefit of this runtime environment is its security model. Managed code is assigned different trust levels depending on its origin—whether it comes from the internet, a local machine, or a company network. This trust level determines what actions the code can perform. For example, an application downloaded from the internet might be allowed to play a sound or show a graphic animation, but it won’t be permitted to access files on your computer or connect to your network unless explicitly granted permission.

This security enforcement is known as Code Access Security (CAS). It protects users by ensuring that even if they run code from external sources, that code can't harm their system or compromise their data.

The CLR also focuses heavily on code reliability. It uses a feature called the Common Type System (CTS), which ensures that all managed code is self-explanatory and follows strict type rules. This means that code written in one language (like C#) can safely interact with code written in another .NET-compatible language (like VB.NET), as long as they both adhere to CTS standards

Another major advantage for developers is cross-language integration. Developers are free to write applications in the programming language they’re most comfortable with. Regardless of the language used, they can still take full advantage of the .NET runtime, class libraries, and third-party components. This flexibility simplifies the process of building or upgrading applications, especially when migrating from older systems.

Even though the .NET runtime is designed with future technologies in mind, it remains fully compatible with older software. One of its strong points is its ability to work seamlessly with both managed and unmanaged code. This means developers can still use existing components like COM objects and DLLs without having to completely rewrite their applications. This level of interoperability ensures that legacy systems can evolve alongside modern development practices.

In terms of performance, the runtime is designed to be highly efficient. One of the key performance features is the Just-In-Time (JIT) compiler. Instead of interpreting code at runtime (which is slower), the JIT compiler converts managed code into native machine code right before execution. This results in faster execution speeds.

Additionally, the built-in memory manager plays a crucial role in maintaining system efficiency. It helps prevent memory fragmentation and optimizes the way memory is used, leading to improved performance. By managing memory allocation and garbage collection automatically, it ensures that applications run smoothly and reliably.

Moreover, the .NET runtime can be hosted by powerful server-side platforms like Microsoft SQL Server and Internet Information Services (IIS). This enables developers to write their business logic using managed code while leveraging the high performance and scalability offered by these enterprise-level servers.

b) .NET Framework Class Library

The .NET Framework Class Library is a comprehensive collection of reusable components that work hand-in-hand with the runtime environment. These components, or types, are designed using object-oriented principles. Developers can easily extend or inherit these types to build their own features, making development faster and more intuitive.

Because these classes are so well-integrated, the learning curve for working with the .NET Framework is relatively smooth. Developers can also combine their own custom-built classes with third-party components without facing compatibility issues.

A great example is the way .NET handles collections. It offers a variety of interfaces that allow developers to build their own custom collection types, which still work seamlessly with existing .NET collection classes.

The class library provides support for a wide range of programming tasks. These include:

- Handling strings and text processing
- Managing collections of data
- Working with files and directories
- Establishing database connections

In addition to these general capabilities, the .NET Class Library includes components for building more specialized applications. Developers can use it to create:

- Console applications for command-line tasks
- Scripted or hosted applications
- Graphical user interface (GUI) applications using Windows Forms
- Web applications using ASP.NET
- XML-based web services
- Windows services for background tasks

This flexibility makes the .NET Framework a highly versatile toolset for building all kinds of software—from simple desktop utilities to complex enterprise solutions.

C). Client Application Development

Client applications refer to software programs that users typically interact with on their desktops. These applications are designed to open graphical windows or forms that enable users to perform everyday tasks. Examples include tools like text editors, spreadsheets, and customized software for business activities such as data entry, report generation, and more. These applications rely heavily on graphical user interface (GUI) elements like buttons, windows, menus, and dialogs, and often work directly with local system resources such as the file system, printers, or other hardware.

In the past, a common way to build interactive applications that ran on a user’s machine was through ActiveX controls—components that could be downloaded and embedded in web pages. However, these have now been replaced with a more secure and powerful system: Windows Forms controls, which are part of the .NET Framework. These controls behave similarly to traditional desktop applications, providing access to the user interface and limited local system features, but with better safety and reliability.

Historically, client applications were written using languages like C++ with Microsoft Foundation Classes (MFC) or developed using Visual Basic, a popular tool for rapid application development. The .NET Framework modernizes this approach by combining these older technologies into a single, consistent development environment. This makes it easier for developers to create rich, responsive desktop applications without needing to manage the complexities of legacy code or tools.

The .NET Framework provides a specialized set of tools for building user interfaces, known as Windows Forms. These tools allow developers to quickly create application windows and interface elements like text boxes, labels, buttons, and toolbars. Developers can also easily modify visual properties such as size, color, layout, and behavior to match business needs. One of the advantages of the framework is that it automatically handles certain operating system limitations—such as recreating a form if its visual settings are not natively supported.

Security is also a major benefit. Unlike older technologies, Windows Forms controls run in a managed environment with restricted access to system resources. This means the application can use certain features of the local system—such as interacting with files or displaying windows—while remaining limited in terms of what it can access. As a result, applications can be distributed more safely, even over the internet, without exposing the system to unnecessary risk.

a).Developing Server-Based Applications

In the .NET ecosystem, server-side applications are typically hosted by external applications that integrate the .NET runtime. These hosts, often unmanaged, allow managed .NET code to dictate server behavior, blending the flexibility of the runtime with the robustness of the host system.

This model offers several advantages: you can harness all the features of the .NET class library and runtime—such as security, memory management, and exception handling—while maintaining the high performance and scalability of powerful server platforms like Internet Information Services (IIS) and SQL Server. For example, business logic for web or database applications can be written in .NET and executed within these servers, making the most of their reliability and throughput.

b) Managed Code on the Server Side

ASP.NET is the core framework used for building web applications using managed .NET code. It’s more than just a runtime—it’s a complete system for developing dynamic web pages and services. Whether you're creating traditional web forms or more modern XML web services, ASP.NET provides the architecture needed to build, deploy, and run applications securely and efficiently.

One of the significant components of ASP.NET is its support for Web Forms and XML Web Services. Web Forms allow developers to create interactive, data-driven web pages using server-side logic. XML Web Services, on the other hand, are reusable components without a graphical interface, designed to be consumed by other applications—making them ideal for integration in large, distributed systems.

Unlike older scripting technologies, ASP.NET applications are compiled rather than interpreted. This leads to faster execution and more efficient resource management. Code-behind files, for example, allow developers to separate business logic from HTML content, making web pages easier to manage and maintain.

The framework also embraces modern standards such as SOAP, WSDL, and XML, enabling communication and data sharing across different platforms and programming languages. This ensures interoperability, allowing .NET-based services to interact with systems built using non-Microsoft technologies.

c) ASP.NET and Web Forms

ASP.NET revolutionized web development by introducing a powerful framework built around the Common Language Runtime (CLR). This framework provides a performance boost over traditional web scripting methods. Since ASP.NET applications are compiled, they benefit from just-in-time compilation, early binding, and other runtime optimizations—resulting in faster and more efficient applications right from the start.

Visual Studio, the primary development environment for ASP.NET, enhances productivity through features like WYSIWYG (What You See Is What You Get) editing, drag-and-drop components, and streamlined deployment tools. These help developers build applications quickly without sacrificing flexibility.

ASP.NET is also highly flexible—it supports multiple programming languages and integrates seamlessly with existing codebases. Developers can reuse components, interact with COM-based systems, and utilize features like automatic garbage collection and reference management.

One of the standout features of ASP.NET is its simplified development experience. Common tasks such as user input handling, form submissions, and authentication are handled more efficiently, reducing the amount of boilerplate code.

In terms of deployment and management, ASP.NET applications use a plain-text configuration system, allowing for quick updates without restarting the server. This "zero-touch deployment" ensures smooth updates and easy scalability, making it ideal for enterprise-level applications.

ASP.NET Web Forms

Web Forms is a key component of ASP.NET that provides a structured, event-driven programming model for building interactive web applications. It bridges the gap between traditional web pages and desktop applications by enabling developers to create reusable UI components and organize logic cleanly.

Web Forms pages use the .aspx extension and can be deployed under any IIS virtual directory. When accessed by a browser, the server compiles the .aspx file into a .NET class and executes it to generate the page dynamically. The compiled class is reused for future requests, improving performance.

There are two main development approaches for Web Forms:

1. Inline Code – where HTML and server-side code coexist in the same file.
2. Code-Behind – where the logic is separated into a dedicated file, promoting cleaner architecture.

To further streamline development, ASP.NET provides server controls—components like buttons, labels, and input fields that run on the server and maintain their state between requests. These controls simplify user interaction handling and eliminate the need for complex client-side scripting.

ASP.NET also supports rich customization through templates and third-party controls. Developers can extend the interface or logic easily, while ensuring compatibility across browsers without relying on cookies or external script libraries.

d) Crystal Reports Integration

Crystal Reports is the default reporting solution within Visual Studio .NET. It provides developers with a powerful tool to design and generate professional-looking reports that can be embedded within both desktop and web applications.

Instead of writing complex code to generate data summaries or visualizations, developers can use the Crystal Report Designer to define layouts, apply formatting, and create data groupings or charts visually. The report engine handles the processing and rendering, saving time and reducing errors.

Crystal Reports comes with built-in Report Experts that guide users through report creation. These tools support advanced features like:

- Drilling into chart data for more detailed views.
- Generating summary fields, subtotals, and percentage breakdowns.
- Highlighting data using conditional formatting and rotating text.

This integration simplifies the reporting process in .NET applications, especially for business and analytics dashboards, where visual data representation is critical.

2.BACK END TECHNOLOGY:

a. About Microsoft SQL Server

Microsoft SQL Server is a powerful and widely-used relational database management system (RDBMS) that utilizes Structured Query Language (SQL) for managing and accessing data. Designed to support various applications and services, SQL Server provides a robust and scalable platform that can handle anything from small desktop applications to large-scale enterprise systems.

At its core, SQL Server is built upon the client/server model and offers comprehensive tools for data storage, manipulation, and retrieval. It ensures data integrity, security, and accessibility while offering tools to handle data relationships, transactions, and user concurrency.

b. Understanding Databases in SQL Server

A database in SQL Server is essentially a structured collection of data stored in tables. It doesn’t directly show data to users but serves as a backend data source accessed via applications.

Databases consist of two main elements:

- Physical files – These contain the actual data.
- DBMS software – This manages access, maintains rules, and ensures consistency and security.

The Database Management System (DBMS) plays a critical role in ensuring that:

- Data relationships are correctly maintained.
- Data storage rules are enforced.
- The system can recover data to a consistent state after a failure.

c. Relational Database Model

The relational model is the most effective approach to organizing data. In this model, data is arranged into tables (also called relations) composed of rows (records) and columns (attributes). The strength of this model lies in its mathematical foundation, specifically set theory.

To ensure optimal structure and remove redundancy, relational databases go through a process called normalization, where data is broken down into multiple related tables while preserving integrity.

d. Client/Server Architecture

SQL Server operates on the client/server architecture, where the server is a central, high-powered machine that holds the database and handles requests from multiple client machines. These clients connect over a network to access shared resources on the server.

The server-side houses both the database files and the DBMS software. SQL Server’s communication components allow applications running on client machines to connect to and interact with the server. One of SQL Server’s strengths is its ability to handle thousands of concurrent client connections while maintaining data consistency and concurrency control.

e. Structured Query Language (SQL)

SQL (Structured Query Language) is the primary language used to interact with relational databases. It allows users to perform operations such as inserting, updating, deleting, and querying data.

SQL Server adheres to the standards defined by ANSI and ISO. The widely accepted standard implemented is SQL-92, which provides a robust foundation for database operations. Through SQL, developers can create powerful and flexible applications that interact efficiently with the data layer.

f. Database Normalization

Normalization is a technique used to design database structures that reduce redundancy and improve data integrity. The various normal forms include:

- 1st Normal Form (1NF): Ensures each field contains only atomic values and no repeating groups.
- 2nd Normal Form (2NF): Builds on 1NF by ensuring that non-key attributes depend fully on the primary key.
- 3rd Normal Form (3NF): Further refines 2NF by removing transitive dependencies (non-key attributes depending on other non-key attributes).

- Boyce-Codd Normal Form (BCNF): A stricter version of 3NF, ensuring every determinant is a candidate key.
- 4th Normal Form (4NF): Ensures that multivalued dependencies are properly addressed.
- 5th Normal Form (5NF): Also called Projection Join Normal Form, it handles complex join dependencies, ensuring data can be split and rejoined without loss.

g. Features and Advantages of SQL Server

Microsoft SQL Server offers a wide range of features that make it suitable for businesses of all sizes: Ease of Use: Installation, setup, and administration are simplified with built-in tools.

- Scalability: SQL Server runs efficiently on both small systems (like laptops) and large enterprise servers.
- Concurrency: Handles multiple users and applications accessing the database simultaneously without conflict.
- Data Warehousing and OLAP: SQL Server supports business intelligence with tools for online analytical processing and data aggregation.
- Visual Tools: It offers graphical interfaces for designing, querying, and managing databases.

h. Databases in Practice

A SQL Server instance can host multiple databases, each storing specific data sets such as employee records, inventory, or customer orders. These databases can be isolated or interrelated based on the system's design.

Before creating a database, it is vital to understand how to structure its components, such as tables, views, stored procedures, triggers, and indexes, to ensure optimal performance and maintainability.

3.Middleware Technology

Active Data Objects.Net Overview

ADO.NET represents a significant evolution from the classic ADO (ActiveX Data Objects) model, specifically tailored to meet the demands of scalable, web-based applications. It was designed with core principles like statelessness, scalability, and XML integration in mind. While it builds upon some familiar ADO components—like Connection and Command—ADO.NET introduces several powerful new elements, most notably the Dataset, Data Reader, and Data Adapter.

The Dataset: A Disconnected Data Container

At the heart of ADO.NET lies the Dataset, which sets it apart from previous data access models. Unlike earlier connection-based approaches, the Dataset is completely disconnected from the underlying data source. This independence means that once data is loaded into a Dataset, it can be manipulated without maintaining an active connection to the database. Inside a Dataset, you can find familiar structures such as tables, columns, relationships, constraints, and even views, mimicking the design of a traditional database but operating in-memory.

The Role of the Data Adapter

The Data Adapter serves as the bridge between the Dataset and the data source.

It performs two critical tasks:

Fetching data from the database into the Dataset.

Pushing updates made in the Dataset back to the database.

These actions are executed through SQL commands, which the Data Adapter manages under the hood. This message-based architecture supports modern, multi-tier applications by handling data in disconnected chunks, which promotes efficiency and scalability.

Unified Programming Through XML

Another key feature of the Dataset is its XML-based structure. This enables a uniform programming model that works seamlessly across various data formats—whether flat files, relational databases, or hierarchical data. Regardless of where the data comes from, the Dataset treats all information the same way, providing developers with a consistent set of APIs to interact with.

Managed Providers: The Connection Gateways

While the Dataset itself is agnostic about its data source, Managed Providers handle the specifics. Their role is to establish connections, load data into the Dataset, and save changes back to the database. The .NET Framework includes two primary managed providers:

SQL Server .NET Data Provider (System.Data.SqlClient): Optimized for connecting directly with Microsoft SQL Server.

OLE DB .NET Data Provider (System.Data.OleDb): Used to interact with other databases via OLE DB.

Each provider includes four essential objects:

- Connection – Opens a channel to the database.
- Command – Executes SQL queries and statements.
- Data Reader – Offers a fast, forward-only read of data.
- Data Adapter – Manages data transfers to and from the Dataset.

SUMMARY

It explains how the project was implemented using Visual Studio for development and SQL Server for database management. The system was built to run on basic hardware with Windows 7 or higher. Technologies like the .NET Framework and ASP.NET were used to create a secure and accessible web application. The platform supports voice-guided features, making it user-friendly for visually impaired users.

CHAPTER 6

FEASIBILITY REPORT

FEASIBILITY REPORT

A feasibility study serves as a high-level overview of a project, aimed at answering important initial questions such as: What problem are we trying to solve? Is there a viable solution? And is it worth pursuing? This kind of study is typically conducted once the problem has been clearly understood, with the primary goal being to assess whether the proposed system is achievable from a technical, operational, and economic standpoint. By carrying out a detailed feasibility analysis, decision-makers gain a well-rounded understanding of the proposed solution and its practicality. For this particular project, the feasibility study focused on three core areas to ensure the solution is realistic and doesn’t run into major obstacles.

Technical Feasibility

Technical feasibility involves evaluating whether the necessary technical tools, resources, and expertise are available to successfully develop and run the system. It assesses whether the chosen technology stack is appropriate not just for the current scope of the project, but also for future expansion and maintenance. In this case, all the required technologies are easily accessible and well-supported. The system is developed using ASP.NET with C# for building the web application, and Microsoft SQL Server is used to manage the backend database. The application is compatible with Windows 7 or newer, making it widely deployable on most systems.

The project also benefits from a modular design, which means it can be scaled up or modified in the future without major rework. Additionally, user-centric features like voice navigation, keyboard-based interaction, and real-time audio feedback have been incorporated to ensure the system is inclusive, especially for users with visual impairments. These tools make the system not only functional but also user-friendly. The technology used is known for its security, stability, and accessibility, which reinforces the fact that the project is technically feasible.

Technologies Used:

Component	**Specification**
Operating System	Windows 7 or higher
Programming Language	ASP.NET with C#
Database	Microsoft SQL Server
Documentation Tool	Microsoft Word

6.2 Economic Viability

Economic feasibility focuses on determining whether the proposed system is cost-effective—essentially weighing the total development and maintenance costs against the expected benefits. This includes a detailed look at both direct expenses (like tools and infrastructure) and indirect costs (such as time and labor). In the case of this project, the financial outlook is very promising. It was developed using existing tools and software that were already part of the academic environment, which meant no additional spending was needed for new software licenses or hardware purchases. Tools like Visual Studio, SQL Server, and the required operating system were either open-source or provided through institutional licensing agreements.

Moreover, the project was undertaken by a team of four students as part of their academic coursework, effectively eliminating labor costs. The project timeline was set at six months and was completed within that period, further reinforcing the project's budget adherence. When considering the minimal development cost in contrast with the social value it offers—especially by empowering visually impaired users to shop independently—the return on investment is clearly favorable.

Major Cost Considerations:

- Planning and analysis expenses
- Use of already available hardware and software
- Leveraging existing development platforms
- Minimal to no maintenance cost during the initial deployment phase
- Overall, the project stands as economically feasible, offering a high-impact solution without demanding significant financial resources.

6.3 Operational Feasibility

Operational feasibility evaluates whether the proposed system can be practically implemented within the intended environment and whether it meets the expectations of its users. This includes factors such as ease of use, user satisfaction, system compatibility with existing infrastructure, and the level of effort required for ongoing maintenance.

In this project, operational feasibility was ensured by designing the system specifically around the real needs of visually impaired users. Features such as voice navigation, keyboard-friendly interfaces, and compatibility with screen readers were integrated, following the Web Content Accessibility Guidelines (WCAG 2.1). These additions make the system highly accessible and allow users to navigate the site independently, without needing visual cues.

The development process was handled by a dedicated team of four students, covering everything from design and development to testing—without requiring additional support or staff. Furthermore, the system was designed to work on standard hardware, meaning there is no need for any specialized equipment. Daily operational tasks like backing up data, updating the system, and modifying voice scripts have all been scheduled to ensure smooth, long-term use.

SUMMARY

Chapter 6 presents a feasibility analysis of the proposed e-commerce website tailored for visually impaired users. It evaluates the project based on three core aspects: technical, economic, and operational feasibility. Technically, the system is viable as it utilizes widely supported technologies like ASP.NET with C# and Microsoft SQL Server, all of which are compatible with accessible interfaces and voice-guided features. Economically, the project is cost-effective because it leverages existing academic resources and software, eliminating the need for additional investment. Since the development was completed by students as part of their curriculum, labor costs were negligible. Operationally, the system is designed to align with the needs of visually impaired users by incorporating voice navigation, keyboard inputs, and screen reader compatibility in accordance with WCAG 2.1 guidelines. These aspects ensure the system is functional, affordable, and user-friendly, making it a feasible and impactful solution.

CHAPTER 7

TESTING

Testing plays a critical role in the success of any large-scale project, including this one. To ensure that all components work as expected under various conditions and deliver the intended outcomes, thorough testing must be carried out. A well-tested system boosts confidence in its reliability and helps guarantee a smoother user experience.

7.1 System Testing

In this project, system testing was conducted to ensure that the application fulfilled all specified user requirements. The development was done using ASP.NET with C# for both the front-end interface and back-end logic. The system underwent extensive testing with feedback from actual users. Any issues or faults discovered during this process were resolved before final implementation. The flow of forms and user interactions within the application was consistent with the actual data movement, making the system more intuitive and dependable.

7.2 Levels of Testing

To detect potential errors at various stages of development, multiple levels of testing were performed before proceeding to user acceptance testing. These levels are:

Unit Testing

This involves testing individual software modules or components independently. It’s also referred to as module testing. Each unit was examined during the coding phase to ensure it delivered the correct output for a given input. This process helps identify and resolve issues at the earliest stage of development.

Integration Testing

This involves testing individual software modules or components independently. It’s also referred to as module testing. Each unit was examined during the coding phase to ensure it delivered the correct output for a given input. This process helps identify and resolve issues at the earliest stage of development.

System Testing

Once integration is complete, system testing evaluates the performance of the complete application in a real-time environment. It assesses whether all components function as expected when brought together, validating the software's readiness for actual deployment.

Validation Testing

After all internal testing is done, the system is validated to ensure it meets the initial business requirements. Validation involves checking if the software behaves according to customer expectations. If it meets the desired criteria, the system is accepted. If discrepancies are found, a list of issues is compiled and addressed.

OutputTesting

This testing step ensures that the system produces the correct output in the right format. User feedback was taken into account to design both on-screen and printed output formats. As the output generation matched user expectations, no changes were necessary at this stage.

UAI

The final stage in the testing process involved confirming that the system met user expectations. This was achieved by regularly engaging with potential users during development, incorporating their suggestions, and adjusting the system where needed. Positive user feedback during UAT confirmed the system's success.

7.3 Test Cases

One of the main functionalities tested was Registration, where users had to fill in all required fields accurately. Validation was applied to ensure that no characters were entered into numeric fields, especially in login IDs. Another critical area tested was the Login process, where both the login ID and password were mandatory. If the input was incorrect, an appropriate error message was shown to guide the user.

7.4 Validation Criteria

To ensure data accuracy and security, several validation rules were enforced:

- Required fields could not be left empty.
- Numeric fields were checked to ensure they didn’t contain letters, and text fields like names were verified to avoid numeric input.
- All primary keys were system-generated to avoid duplication.
- Error-handling mechanisms were implemented across core operations such as saving, editing, and deleting.
- Instant validation occurred when a user moved out of a field (using Tab or Enter), and if the input was invalid, the focus returned to that field with a suitable message.

Summary

Chapter 7 centers around the essential process of testing in the development of the e-commerce platform for visually impaired users. It outlines various testing methodologies employed to ensure the software meets user expectations and performs reliably. The chapter begins with system testing to verify that all user requirements have been fulfilled. Multiple layers of testing were conducted—unit testing focused on individual modules, integration testing checked how components interact, and system testing evaluated the complete application’s performance. Validation testing confirmed that the final product meets both functional and non-functional criteria. Test cases for critical functionalities like login and registration were executed, and necessary corrections were made. Emphasis was also placed on validation rules, such as mandatory field checks and proper error handling, ensuring that all data inputs are accurate and secure. Overall, the testing efforts contributed significantly to the project’s success, ensuring stability, reliability, and usability across all aspects of the system. [1][3][6][9].

CHAPTER 8

ADVANTAGES AND LIMITATIONS

8.1 Advantages

- Proposed e commerce web application help or guide blind people to order product [3][4].
- Interactive Voice response from the website to guide people [7][5].
- Convenience of navigating through pages and ease of access to every product[9].

8.2 Limitations

- This application requires an active internet connection for proper functionality [6].
- The users must enter accurate and complete information failure to do so may result in system errors or irregular behaviour .

8.3 Features

Load Balancing:

The system enforces load control by allowing access only during the administrator’s active session. This minimizes server load and enhances performance during designated time frames.

Easy Accessibility:

The platform allows for convenient storage and retrieval of records. Various types of information can be easily managed through the system's interface.

User-Friendly Interface:

Designed with the end-user in mind, the e-commerce website ensures smooth navigation and intuitive interaction for all customers.

Reliable and Efficient:

By utilizing a secure server-based database, the system eliminates the need for manual data entry or spreadsheet tracking. This approach enhances data reliability, security, and operational efficiency, all while reducing maintenance costs.

Summary

The e-commerce web application for the visually impaired offers several advantages, including interactive voice responses, convenient navigation, and ease of access to products [3][7]. However, it has limitations, such as requiring an active internet connection and correct user input [6][9]. The system's features include load balancing, easy accessibility, user-friendliness, and efficient and reliable data management, making it a robust and accessible platform for users [4][8][9].

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

9.1 Conclusion

This project, titled "E-Commerce Website for the Visually Challenged," was developed as part of our System Design course using the ASP.NET framework. The creation of this web application demanded significant dedication and teamwork. Throughout the development process, we gained valuable hands-on experience and deepened our understanding of web development and accessibility-focused design.This platform is built to provide a convenient and inclusive online shopping experience for individuals with visual impairments. By integrating assistive technologies such as screen readers, braille displays, and voice-controlled navigation, the website empowers users to independently explore, select, and purchase products. Features like audio descriptions, high-contrast visual modes, and keyboard-friendly navigation contribute to the platform’s usability and inclusivity.

9.2 Future Scope

1. **Enhanced Accessibility Through Emerging Technologies**
   Future improvements could include integration of artificial intelligence tools such as image recognition, object detection, and augmented reality to enrich the browsing experience for users with visual impairments.
2. **Personalized Shopping Experience**
   Implementation of machine learning algorithms to offer personalized product suggestions based on user preferences, behavior, and past purchase history.
3. **Voice-Driven Commerce**
   Expanding voice command capabilities will allow users to search for products, navigate pages, and complete purchases using voice interactions, making the process even more accessible.
4. **Integration with Smart Devices**
   Connecting the platform with smart home devices (like Amazon Alexa or Google Assistant) would enable users to interact with the website using voice assistants, offering a hands-free shopping experience.
5. **Continuous Accessibility Improvement via Feedback**
   Introducing a user feedback system to collect suggestions and experiences from visually impaired users, which can be used to refine and improve accessibility features over time.

6. **Collaborations with Advocacy Organizations**

   Partnering with institutions and nonprofits focused on visual disabilities can help promote the platform, enhance its functionality, and ensure it meets the specific needs of its intended users.

7. **Mobile Application Development**

   Developing a fully accessible mobile app will provide users with the convenience of shopping on-the-go while maintaining all accessibility features of the web version.

BIBILOGRAPHY

[1] Al-Wabil, A., Zaphiris, S., & Wilson, S. (2007). *Design principles and navigational requirements for users with visual impairments on the web*. *International Journal of Human-Computer Studies, 65*(11), 887–902.

[2] Kane, S. K., Bigham, J. P., & Wobbrock, J. O. (2015). *Slide Rule: Multi-touch techniques for enabling blind users to interact with mobile touchscreens*. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 73–80).

[3] Alattas, W. M., & Albakri, M. A. (2023). *Creating usable and accessible e-commerce platforms for visually impaired users: A practical usability analysis*. *International Journal of Advanced Computer Science and Applications, 11*(6), 385–392.

[4] Leung, L. A. (2021). *Overview of assistive technology in e-commerce for users with visual impairments*. *Journal of Accessibility and Design for All, 7*(2), 127–145.

[5] Huang, T. S., Klemmer, S. R., & Hearst, M. A. (2020). *Improving web navigation for blind users using audio-enhanced browsing techniques*. *ACM Transactions on Accessible Computing, 1*(2), 1–24.

[6] Ladner, R., & Christian, A. (2018). *Best practices in designing inclusive e-commerce systes for users with accessibility needs*. *Communications of the ACM, 61*(6), 62–69.

[7] Borodin, J., Dausch, A. S., & Ramakrishnan, I. V. (2015). *Voice-controlled interfaces in online retail: Making e-commerce more accessible for blind users*. *IEEE Transactions on Human-Machine Systems, 45*(6), 694–703.

[8] Balasubramanian, N., Teevan, J., & Cutrell, E. (2020). *Enhancing e-commerce search accessibility for blind shoppers*. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 317–326).

[9] Kher, S. R., & Mulay, M. S. (2022). *Inclusive design model for developing accessible e-commerce solutions for the visually challenged*. *International Journal of Computer Applications, 182*(10), 25–29.