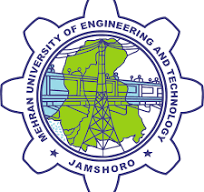
Mehran University of Engineering &Technology

Jamshoro Sindh, Pakistan



WE-CEP Report

Groups Members:

**Khet (22CS077)**

**Vias Kohli (22CS053)**

**Bilal Qazi (22CS029)**

Submitted to: Engr.Zaib ul Nisa

DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING

|  |
| --- |
| Department of Computer Systems Engineering  Mehran University of Engineering and Technology, Jamshoro |

|  |  |  |  |
| --- | --- | --- | --- |
| Course: Web Engineering (CS-373) | | | |
| Instructor | Engr. Zaib Ul Nisa | **Assignment Type** | Complex Engineering Problem |
| Semester | 5th | **Year** | 3rd |
| Submission Deadline | 10-04-2025 | **Assessment Score** |  |

The semester project is designed to enable students to solve complex engineering problems using Web Engineering Concepts. The following characteristics of complex engineering problems are targeted in this semester's project on WE.

|  |  |  |
| --- | --- | --- |
| Complex Engineering Problem – Characteristics | | |
| 1 | Depth of knowledge Required | 🗹 |
| 2 | Range of Conflicting Requirements | 🗹 |
| 3 | Depth of Analysis Required | 🗹 |
| 4 | Infrequently Encountered Issues Involved | 🞏 |
| 5 | Beyond codes/standards of practice | 🗹 |
| 6 | Diverse groups of stakeholders with widely varying needs involved | 🞏 |
| 7 | Interdependence (high-level problems including many component parts/sub-problems) | 🞏 |
| 8 | Have significant consequences in a range of contexts | 🞏 |
| 9 | Judgment (Require judgment in decision-making) | 🞏 |

**Project Objectives:**

**Front-End Development:** Design and develop a responsive, intuitive, and user-friendly front-end interface using HTML, CSS, and JavaScript. Ensure cross-browser compatibility and mobile-first design principles.

**Back-End Development:** Implement a robust server-side architecture using

Server-side language like PHP frameworks like Node.js, Django, or others. Ensure smooth handling of client requests, database connections, and application logic.

**Database Design:** Develop a scalable and efficient database schema using SQL Optimize data retrieval and storage with normalized structures and indexing.

**User Authentication and Authorization**: Implement secure user authentication mechanisms, such as OAuth or JWT, and establish role-based access control to protect user data and application resources.

**Form Handling and Validation**: Develop and implement efficient form handling systems with client-side and server-side validation to ensure data accuracy, security, and usability.

**Security and Data Protection**: Integrate security measures like HTTPS, data encryption, CSRF protection, and secure cookies to protect user data and ensure secure interactions between clients and servers.

**Responsive and Adaptive Design**: Ensure that the website adapts to various screen sizes and devices by implementing responsive design patterns and media queries.

**Performance Optimization**: Apply optimization techniques, such as lazy loading, minification, caching, and CDN integration, to reduce load times and improve the overall user experience.

**Error Handling and Logging**: Design a system for proper error handling and logging, capturing potential issues in both client-side and server-side components to ensure smooth debugging and maintenance.

**Performance Optimization:** Apply optimization techniques to enhance the Website’s performance, responsiveness, and energy efficiency.

**Project Deliverables:**

**Technical Report:** A comprehensive technical report detailing the Website design, architecture, and implementation, including documentation of key algorithms and methodologies.

**Source Code:** Complete source code of the Web Engineering Project, well-organized and thoroughly commented for ease of understanding.

**User Documentation:** User guides and manuals explaining the system's functionalities, installation, and usage instructions.

Students would try to apply in-depth **engineering knowledge (WP1)**to complete each project. During the initial study and formulation of the proposed solution, they would focus on the detailed **requirements (WP2), and realtime constraints (WP2), and** performed **in-depth analysis (WP3).**

Introduction:

The ElectroMart project is a secure and responsive eCommerce platform developed to solve real-world challenges in online shopping systems. Built as part of the Web Engineering course, this platform enables users to register, log in, browse electronic products, and place orders seamlessly. The system is secured using JWT-based authentication and developed using modern web development practices including PHP, MySQL, JavaScript, and HTML/CSS. This report outlines the planning, execution, and delivery of the ElectroMart backend system while showcasing its ability to fulfill the requirements of a complex engineering problem (CEP).

**Problem/Requirement Identification:**

In today’s digital landscape, eCommerce platforms are essential for both customers and businesses. The challenge was to design a scalable, user-friendly eCommerce website for a fictional brand named ElectroMart, which sells electronic products. The website must offer user authentication, secure ordering, product browsing, and responsive design. This project addresses multiple engineering challenges, including secure login systems, efficient data flow between frontend and backend, database normalization, and token-based authentication**.**

**Project Objectives**

* To design and develop a secure and scalable eCommerce backend system.
* To implement user authentication using JSON Web Tokens (JWT).
* To manage product data and user orders using MySQL and PHP.
* To create RESTful APIs for front-end-backend communication.
* To ensure code modularity, security, and performance optimization.
* To validate system f unctionality through structured testing using Postman.
* To adhere to real-world software engineering standards and web development practices.

**Implementation:**

**Frontend Technologies:**

* HTML5, CSS3 (responsive layouts, Flexbox/Grid, animations)
* JavaScript (form validation, dynamic product rendering)

**Backend Technologies:**

* PHP (server-side scripting and API creation)
* SQL (database design and queries)
* JWT (secure token-based authentication)
* Postman (API testing)
* SafePay (Payment API)

**Development Tools:**

* XAMPP (Local Server)
* VS Code (Development IDE)
* Composer (Dependency Management)
* Git (Version Control)

**File Structure Overview:**

Final project

├─index.html # Homepage for the storefront

├── login.html # Login interface for users

├── register.html # User registration form

├── products.html # Product listing UI

├── cart.html # Cart overview and actions

├── checkout.html # Checkout and payment interface

├── categories.html # Product category listing page

├── script.js # Main frontend logic (fetch APIs, DOM control)

├── styles.css # Global styling for layout and colors

├── stylesp.css # Product-specific styling (cards, hover, etc.)

├── generateToken.php # PHP script to generate JWT manually (testing only)

├── /electromart-backend

│ ├── index.php # Backend health-check and API listing

│ ├── .htaccess # URL routing and security headers

│ ├── /api

│ │ ├── login.php # Login API: verifies user and issues JWT

│ │ ├── register.php # Register API: creates new user account

│ │ ├── products.php # Returns all products for frontend

│ │ ├── create\_checkout.php # Initiates mock checkout process

│ │ ├── payment.php # Processes mock payment

│ │ └── charge.php # Simulates payment gateway charge

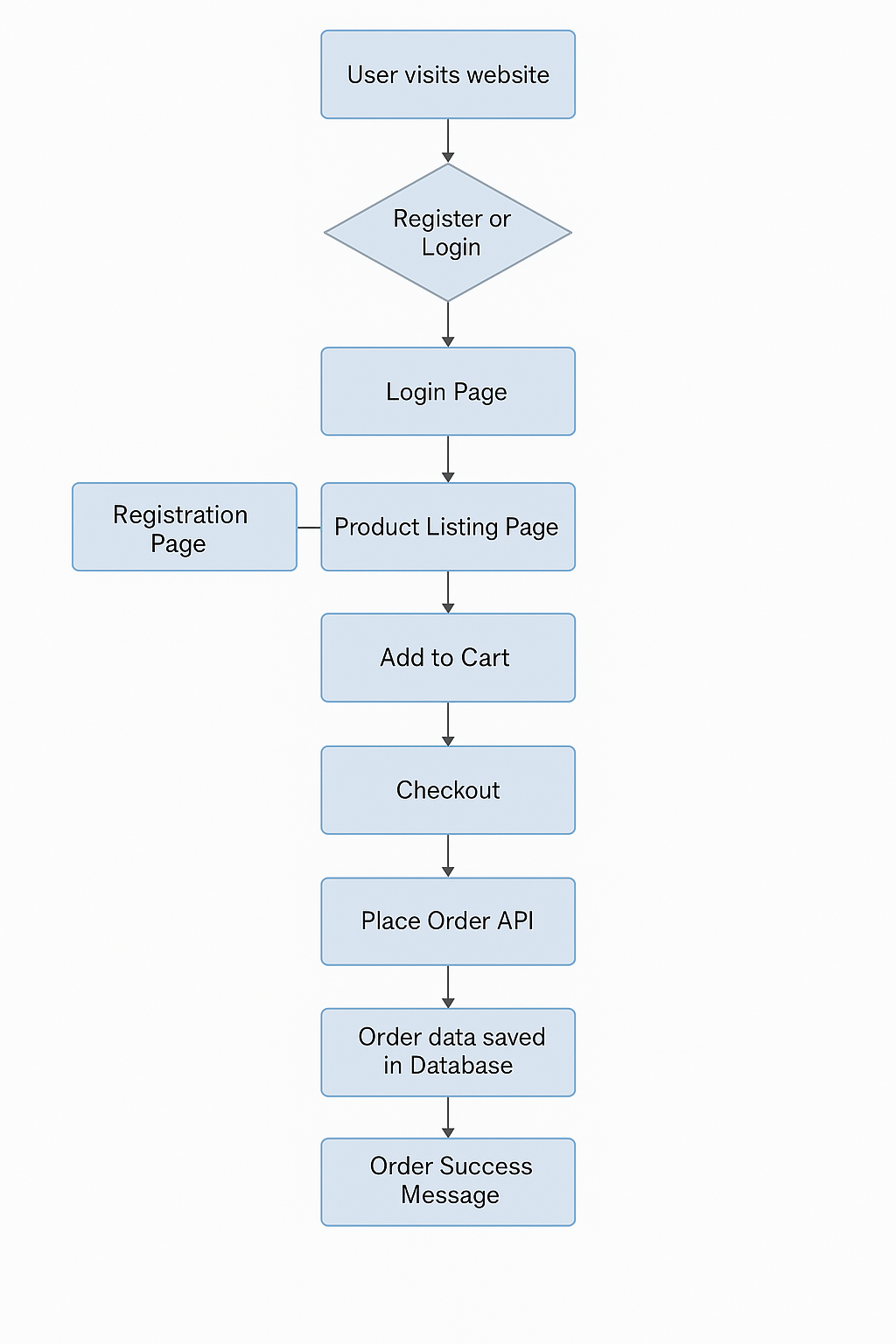
│ ├── /cart

│ │ ├── addtocart.php # Adds product to user cart

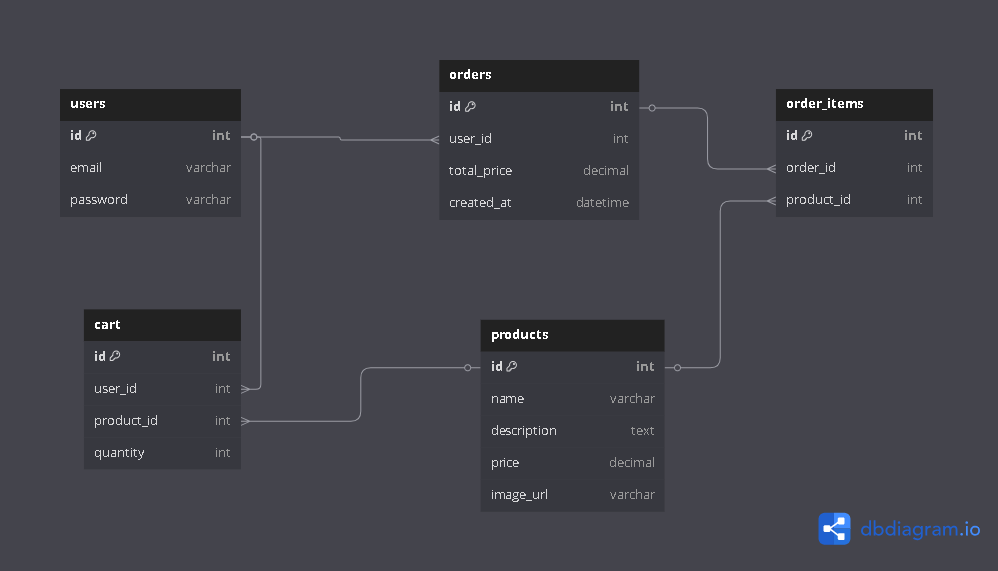
│ │ ├── getcart.php # Retrieves current user's cart items

│ │ └── removeFromCart.php # Removes specific product from cart

**Flow Chart:**



**ER Diagram:**

****

**Testing and Validation**

* Postman used to test all API endpoints
* Manual and automated form validation
* Token expiration testing and invalid token detection
* Tested on multiple screen sizes and browsers

**Registration Page:**

A screenshot of a login form

AI-generated content may be incorrect.

**Login Page:**

A screenshot of a login form

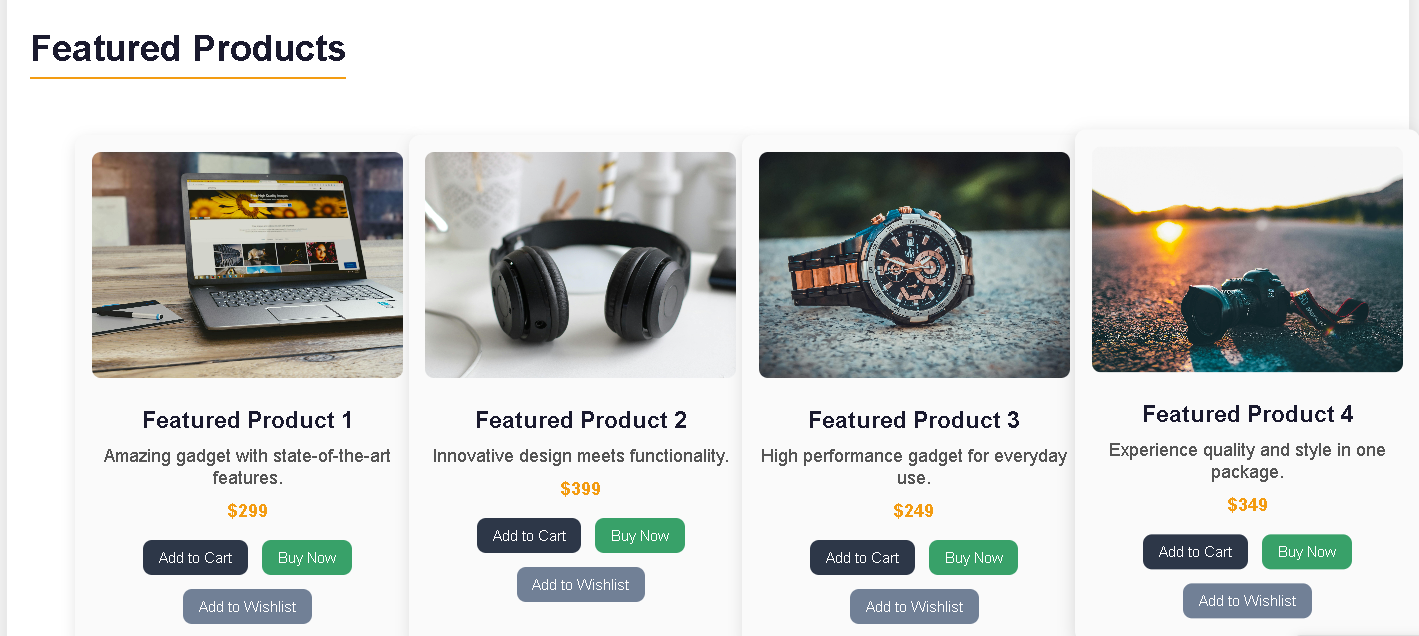
AI-generated content may be incorrect.

**Home page:**

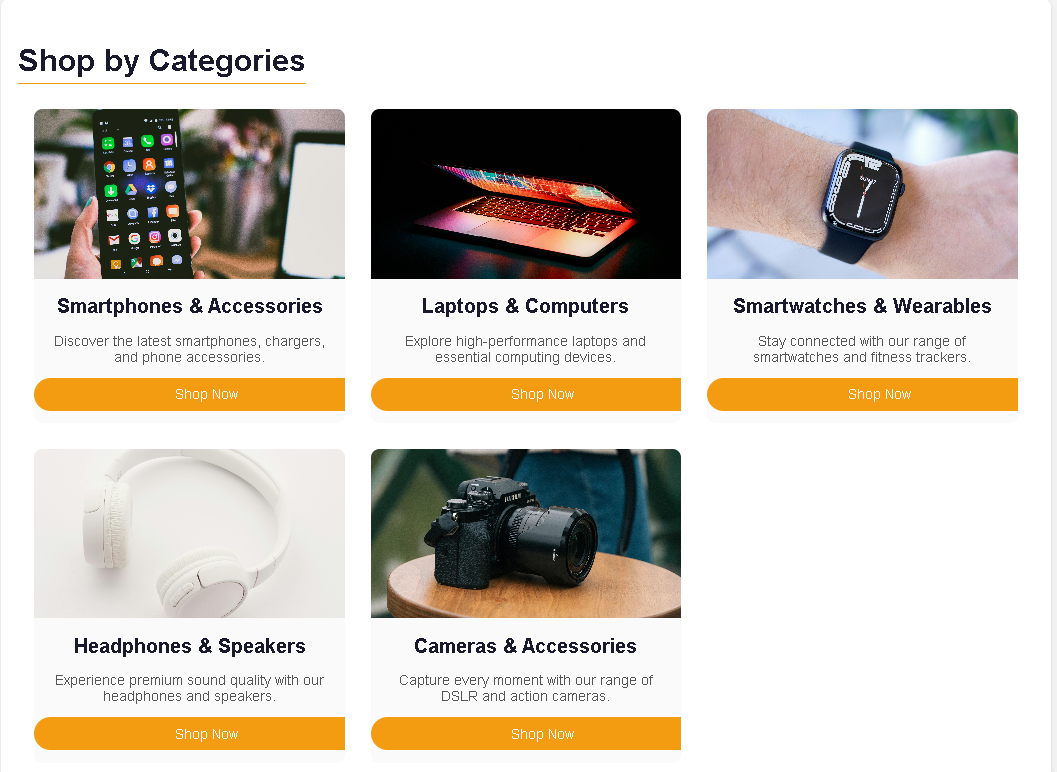
A screenshot of a computer

AI-generated content may be incorrect.

**Home page \_ Featured Product:**



**Product\_categories:**

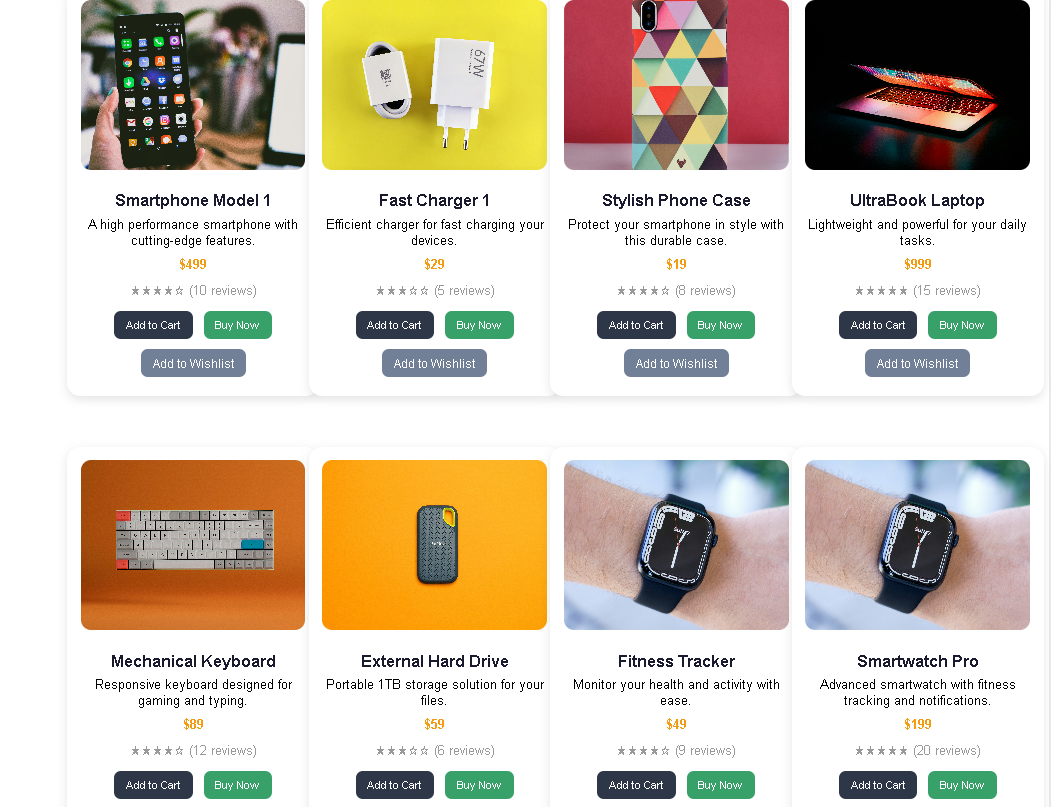
****

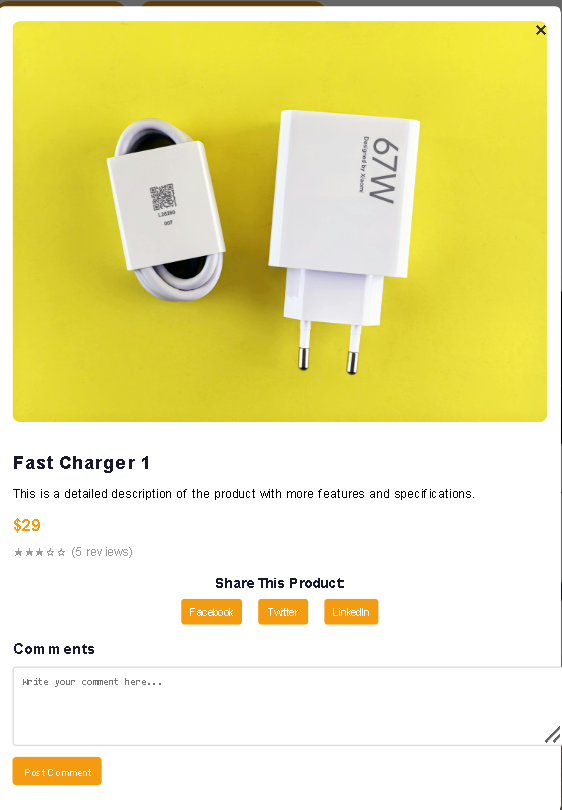
**Our Product:**

**A screenshot of a website

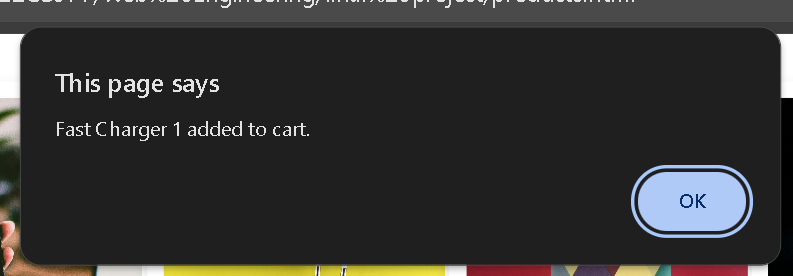
AI-generated content may be incorrect.**

**Product Listing:**

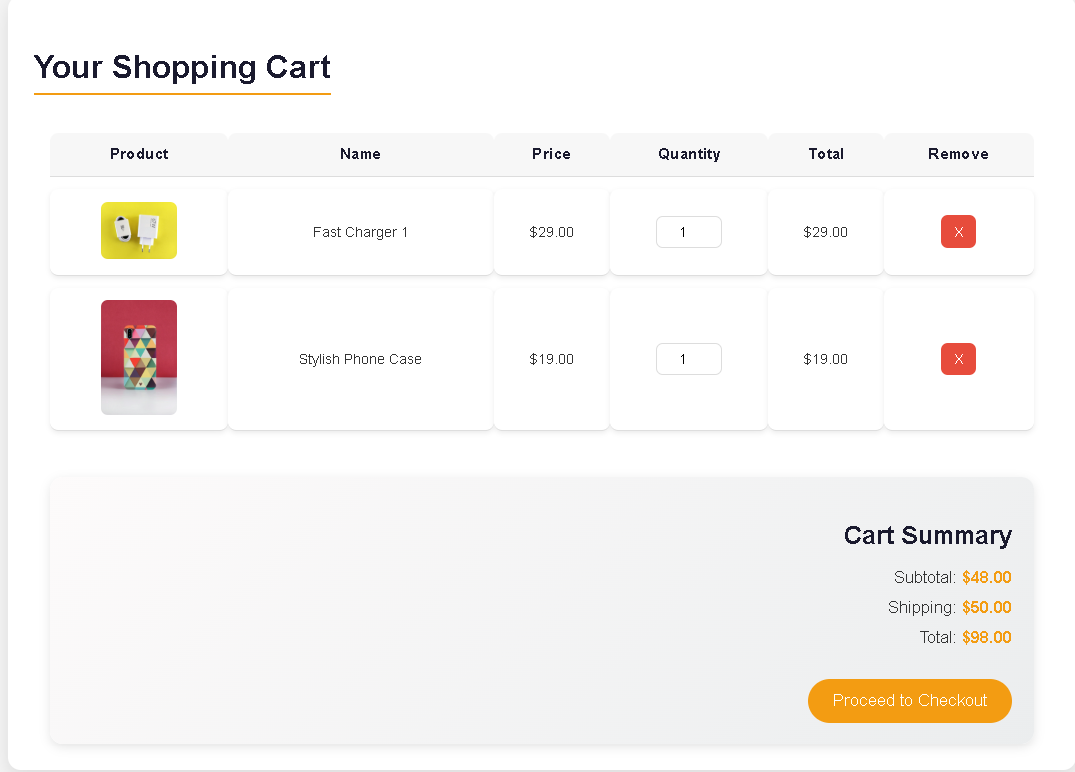
****

****

**Add to Cart:**



**Cart:**



**Checkout\_Customer Details:**

A screenshot of a computer

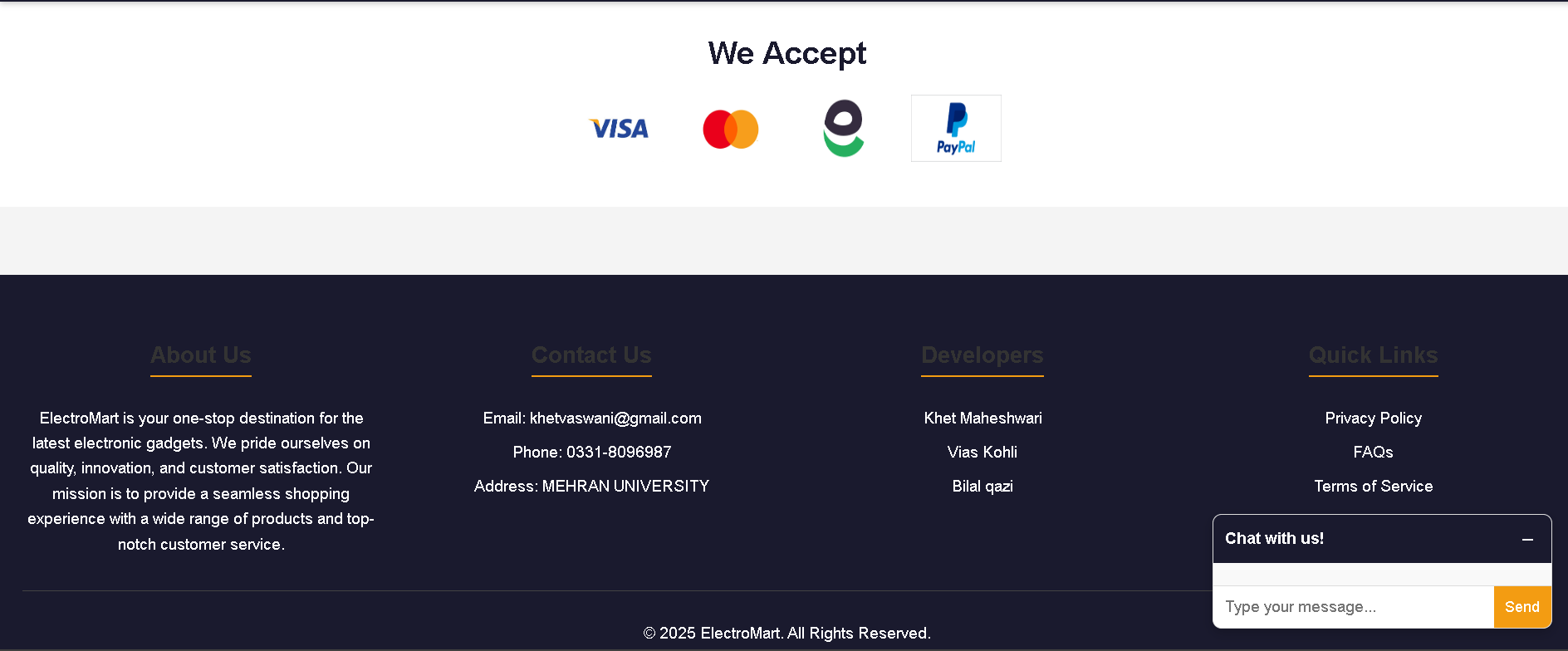
AI-generated content may be incorrect.

**Order\_Confrimed**

A screenshot of a black box

AI-generated content may be incorrect.

Footer:



**Projects would be evaluated on the following criteria: -**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rubrics** | | | | CEP characteristics | Marks distribution |
|  | **Unacceptable 2** | **Acceptable 7** | **Proficient 10** |
| **R1: Idea/Initial Study** | 🞏 | 🞏 | 🞏 | WP2 | 10% |
| **R2: Project Proposal** | 🞏 | 🞏 | 🞏 | WP1, WP3 | 10% |
| **R3: Project Progress** | 🞏 | 🞏 | 🞏 | WP3, WP2,WP5 | 30% |
| **R4: Final Demo** | 🞏 | 🞏 | 🞏 | WP3, WP2,WP5 | 40% |
| **R5: Final Report** | 🞏 | 🞏 | 🞏 | WP3, WP1 | 10% |

**RUBRICS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Unacceptable | Poor | Acceptable | Adequate | Proficient | Score |
| R1 Problem/Requirement Identification | Problem not identified | Problem poorly identified | Problem is identified | Problem is defined adequately. | Problem is identified and analyzed in a well-defined manner. |  |
| R2 Engineering knowledge (standards) | Can not apply engineering knowledge to the solution. | Has difficulty  applying  mathematics to  the solution  of complex  engineering  problems | Correctly applies basic  sciences to the solution  of complex engineering  problems | Correctly applies engineering  fundamentals to the solution of complex  engineering problems | Correctly applies  engineering  specialization to the  solution of complex  engineering problems. |  |
| R3 Efficiency of the solution | Solution does not meet requirements. | A difficult and inefficient solution. | A logical solution that is easy to follow but it is not the most efficient. | Solution is adequately efficient. | Solution is efficient, easy to understand, and maintain. |  |
| R4 Technical Writing | The report is submitted but lacks solutions to major requirements. | The report submitted but not according to the requirements. | The requirements of report writing are not properly addressed. | Reports meets all prescribed requirements. | Reports meets all requirements, and it is prepared in original and corrective way to engage readers. |  |
| Total | | | | | |  |
| Normalized Score = (Total /40)\*05 | | | | | |  |