**Chapter-1**

**Introduction to Project**

**1.1 Introduction:**

Welcome to Electro-Mart, your ultimate destination for all things related to Electronic and Communication Engineering (ECE)! In today's digital age, access to quality resources is crucial for enthusiasts, students, and professionals alike. With this understanding in mind, we've embarked on a mission to create a comprehensive platform that caters specifically to the needs of the ECE community.

Creating an ECE (Electronics and Communication Engineering) components website involves developing a full-stack web application that provides users with the ability to browse and purchase electronic components while receiving email notifications upon successful transactions. The frontend is built using HTML, CSS, and JavaScript to create a responsive and user-friendly interface. On the backend, Node.js and Express.js manage server-side operations, and MongoDB securely stores user and product data. The website includes user authentication features for secure access through sign-up and login functionalities. After completing a purchase, users receive a confirmation email via an SMTP service like NodeMailer, thanking them for their purchase. This project demonstrates the integration of various web development technologies and emphasizes secure and efficient handling of user data and transactions.

Harnessing the power of HTML, CSS, and JavaScript, we've crafted a user-friendly website that serves as a one-stop hub for ECE components and books. Our dedication to excellence extends beyond mere accessibility; we've prioritized responsiveness, ensuring seamless navigation across various devices, be it desktops, tablets, or smartphones. Moreover we've leveraged the capabilities of MongoDB to store user data securely. This not only ensures privacy but also facilitates seamless access to saved preferences and recommendations across multiple sessions.

To enhance your shopping experience, we have integrated a robust email notification system using SMTP. After every purchase, users will receive a confirmation email thanking them for their purchase and providing details about their order. This ensures that users are always kept informed and appreciated for their trust in Electro-Mart.

The homepage serves as the central hub of the ECE components website. It welcomes users with a title and navigation links to other parts of the site, such as the sign-up and login pages. The main content area dynamically displays a list of available electronic components, allowing users to browse and select products. A footer provides additional site information, such as copyright details.

The login page is designed for returning users to authenticate themselves before accessing their account and making purchases. It features a form where users can enter their email and password. If the user does not have an account, the page provides a link to the sign-up page, encouraging new users to register. The login form includes validation to ensure that users provide the necessary information. After submission, the form handles the authentication process by sending the user's credentials to the backend for verification.

These pages are integral to the overall functionality of the ECE components website, ensuring a smooth and secure user experience from browsing products to making purchases and receiving confirmation emails.

**Chapter-2**

**Literature Review**

**Rahul Semi** presented a paper in 2022[1] titled **Web Page design** using html, css and js. The author suggests delving into the intricacies of web page designing using HTML, CSS, and JavaScript. Semil elucidates the importance of these coding languages in modern web development, highlighting their role in achieving a robust separation of content and style. The paper meticulously explores the evolution of HTML5, CSS3, and JavaScript, emphasizing their modular and functional capabilities in creating visually appealing and interactive websites. Through a detailed discussion, Semil provides insights into the nuanced techniques and methodologies employed in crafting compelling web designs, thereby underscoring the significance of these coding languages in shaping the digital landscape.

**Manish Kumar and Neel Doshi** presented a paper in 2022[2] titled **Ecommerce Website** using HTML, CSS3, javascript, bootstrap and Django. The author suggests delving into the expansive realm of e-commerce, comprehensively examining its significance, implementation, and impact on contemporary business practices. Through meticulous analysis, the paper explores the advantages of e-commerce implementations, ranging from cost efficiencies to enhanced customer experiences. Additionally, it delves into various facets of e-commerce operations, including system design, customer management, logistics, and payment systems. Through detailed discussions, the author provides insights into the complex interplay between technology, business strategies, and consumer behaviours in the digital marketplace, highlighting the transformative potential of e-commerce in reshaping traditional business paradigms.

**Avnish Kumar** presented a paper in 2021[3] titled **Big Buy (E-Commerce website)** using fronted web development. The author suggests delving into the expansive realm of e-commerce, comprehensively examining its significance, implementation, and impact on contemporary business practices. Through meticulous analysis, the paper explores the advantages of e-commerce implementations, ranging from cost efficiencies to enhanced customer experiences. Additionally, it delves into various facets of e-commerce operations, including system design, customer management, logistics, and payment systems. Through detailed discussions, the author provides insights into the complex interplay between technology, business strategies, and consumer behaviours in the digital marketplace, highlighting the transformative potential of e-commerce in reshaping traditional business paradigms.

**Ashis Kumar .al** presented a paper in 2018[4] **titled Html5 in Web Development: A New Approach** using Html and Html5. The authors presented p an insightful exploration of HTML5 in web development, offering a fresh perspective on this innovative technology. HTML5 emerges as a pivotal standard, revolutionizing the landscape of web development by providing enhanced functionality to both web users and developers. The paper contextualizes HTML5 within the broader evolution of web standards, highlighting its transformative impact on the field. With a focus on multimedia applications, the authors showcase HTML5's capabilities in graphics rendering, audio and video playback, offline functionality, and location-based services. Through a comprehensive examination of HTML5 features and elements, including canvas graphics, audio/video tags, and web storage, the paper elucidates the myriad possibilities unlocked by this cutting-edge technology. Furthermore, the authors underscore HTML5's role in fostering platform independence and web openness, paving the way for real-time collaborations and improved user experiences. In conclusion, the paper serves as a valuable resource for developers and enthusiasts alike, shedding light on the potential of HTML5 to shape the future of web development.

**Chapter-3**

**Project Description**

**3.1 Title of the Project:**

The title of the project is “**Electro-Mart**”. Electro-Mart is an E-Commerce platform dedicated to providing a seamless shopping experience for electronic components enthusiasts, hobbyists, and professionals alike. The platform offers a range of electronic components, including microcontrollers, sensors, modules, cables, and more. Built with HTML, CSS, JavaScript for the frontend, and Node.js for the backend, Electro-Mart aims to streamline the process of purchasing electronic components online.

**3.2 Objectives of Project:**

* The comprehensive electronic components website will be developed utilizing HTML, CSS, and JavaScript.
* A responsive design will be implemented to ensure seamless shopping experiences across various devices.
* A feature will be implemented where users can access relevant courses or books based on the product they intend to purchase when clicking on "buy new".
* Encryption gateways will be integrated to protect user data securely.
* Scalability will be ensured by considering future enhancements such as expanding e-commerce features and community engagement tools options to accommodate potential growth.

**3.3 Description:**

Electro-Mart is an innovative e-commerce platform designed to cater to the needs of electronic components enthusiasts, hobbyists, and professionals alike. Built with cutting-edge technologies such as HTML, CSS, and JavaScript for the frontend, and powered by Node.js for the backend, Electro-Mart offers a seamless shopping experience with its intuitive interface and user-friendly design. With Electro-Mart, users can browse through a vast catalog of electronic components, ranging from microcontrollers and sensors to modules and cables. The platform's responsive design ensures that users can access it effortlessly across various devices, including desktops, laptops, tablets, and smartphones, enhancing convenience and accessibility. One of Electro-Mart's standout features is its robust authentication system, which includes secure login and signup functionalities. Users can create accounts with ease, ensuring a personalized shopping experience and access to exclusive features. Additionally, Electro-Mart prioritizes user data security by implementing encryption gateways, safeguarding sensitive information such as personal details and payment credentials. This commitment to security instills trust and confidence among users, fostering long-term relationships and loyalty. Electro-Mart sets the standard for excellence in the electronic components marketplace, offering a blend of cutting-edge technology, user-friendly design, and robust security features. Whether you're a seasoned professional or a curious hobbyist, Electro-Mart provides everything you need to fuel your passion for electronics.

**3.4 Software Requirement:**

* **Operating System:**
  + Windows 10 or later
  + macOS Catalina (10.15) or later macOS Catalina (10.15) or later
  + Linux distributions such as Ubuntu 20.04 LTS or later
* **Node.js and npm:**
  + Node.js: Version 12.x or later
  + npm: Version 6.x or later
* **Text Editor or IDE:**
  + Visual Studio Code, Sublime Text, Atom, WebStorm, or any other preferred text editor or IDE compatible with the operating system.
* **Web Browser:**
  + Latest versions of Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge compatible with the operating system.
* **Version Control System:**
  + Git: Version 2.x or later
  + Git GUI tools compatible with the operating system (optional)
* **Database Management System:**
  + MongoDB: Latest stable version
  + MongoDB Compass or other compatible GUI tools for managing databases (optional).
* **Web Server:**
  + Node.js with Express.js or another compatible web server framework.

**3.5 Hardware Requirements:**

* **Server:**
  + A server capable of running Node.js and MongoDB/Sql
  + Minimum CPU and RAM requirements will depend on the expected traffic and workload.
* **Client Devices:**
  + The application should be accessible from various client devices, including desktops, laptops, tablets, and smartphones.
  + Minimum hardware requirements for client devices will depend on the complexity of the application and the resources required to run modern web browsers.
* **Network:**
* Stable internet connectivity is required for accessing the application from client devices.
* The server should have sufficient bandwidth to handle incoming requests and data transfers.

**3.6 Hyper Text Markup Language (Html):**

HTML, or Hyper Text Markup Language, is the standard markup language used to create and structure content on the web. It forms the backbone of almost every web page and is essential for building and presenting content in browsers. HTML works in conjunction with Cascading Style Sheets (CSS) and JavaScript to create the interactive and visually appealing websites we see today.



Fig. 3.1; Html Logo

* **The key features is given below:**
* **Html Document:** An HTML document is a plain text file with a .html extension. It contains HTML tags that define the structure and content of a web page.
* **Html Tags:** HTML tags are used to define elements on a web page. Tags are enclosed in angle brackets (< >). They usually come in pairs, with an opening tag and a closing tag.
* **Document Structure:** HTML documents have a basic structure with <html>, <head>, and <body> sections. The <head> section contains metadata and links to external resources, while the <body> section contains the content visible to users.
* **Headings and Paragraphs:** HTML provides heading tags (<h1> to <h6>) for defining headings of various levels. Paragraphs are defined using the <p> tag.
* **Images:**
  + Images are embedded using the <img> tag. The src attribute specifies the path to the image file.
* **Forms:** HTML forms (<form>) are used to collect user input. Form elements include input fields, buttons, and more.
* **Comments:** Comments in HTML are written using <!-- to start the comment and --> to end it.
* **Form Elements:** HTML forms can include various input types, such as text fields, radio buttons, checkboxes, and more. The <label> element associates a label with a form element.
* **Semantic Elements:** HTML5 introduced semantic elements like <article>, <section>, <nav>, <header>, <footer>, and more. These elements provide meaning to the structure of a document.
* <**Audio and Video:** HTML supports embedding audio and video content using the <audio> and <video> elements.
* **IFrames:** The <iframe> element is used to embed another HTML document within the current document.
* **Meta Tags:** <meta> tags provide metadata about the HTML document. Common uses include setting the character set, viewport settings for responsive design, and specifying keywords for search engines.
* **Accessibility:** HTML supports accessibility features to make web content more usable for people with disabilities. Elements like <nav>, <main>, <header>, and <footer> can enhance screen reader navigation.
* **Tables:** HTML provides the <table> element for creating tables. Tables consist of rows (<tr>) and cells (<td> for data cells, <th> for header cells).

**3.7 Cascade Style Sheet (CSS):**

CSS, or Cascading Style Sheets, is a style sheet language used for describing the presentation and visual representation of HTML and XML documents on the web. In simple terms, CSS allows you to control the layout, styling, and appearance of web pages, making it an essential technology for web development.



Fig.3.2; CSS Logo

* **The key features is given below:**
* **Selectors:** CSS uses selectors to target HTML elements that you want to style. Selectors can be based on element types, classes, IDs, attributes, and more.
* **Properties and Values:** CSS properties define the style aspects you want to apply to selected elements. Each property has a corresponding value.
* **Box Model:** The box model is a fundamental concept in CSS that describes how elements are rendered on the web. It consists of content, padding, border, and margin.
* **CSS**:CSS provides various properties for controlling the layout of elements, such as display, position, float, flexbox, and grid.

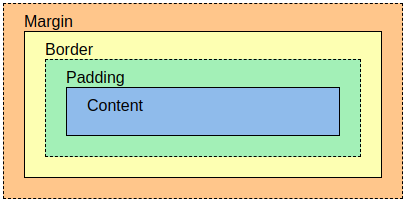


Fig. 3.3; Box Model

* **Colors and Backgrounds:** CSS allows you to set colors for text and backgrounds using properties like color, background-color, and background-image.
* **Responsive Design:** CSS supports responsive design, allowing you to create layouts that adapt to different screen sizes and devices using media queries.
* **Flexbox and Grid:** CSS Flexbox and Grid are layout models that provide powerful tools for creating complex and responsive layouts with ease.
* **Selectors and Combinators:** CSS selectors can be combined using various combinators to target specific elements or groups of elements.
* **CSS Preprocessors:** Preprocessors like Sass and Less extend the functionality of CSS by introducing variables, nesting, and functions.
* **Style Definition:** CSS is used to define the style and layout of HTML elements on a web page. It allows developers to control the appearance of elements, such as fonts, colors, spacing, and positioning, enhancing the visual presentation of the content.
* **Separation of Concerns:** One of the fundamental principles of web development is the separation of concerns. CSS facilitates this by providing a way to separate the styling (CSS) from the structure and content (HTML). This separation enhances code maintainability, allowing changes to styling without altering the underlying content.

**3.8 JavaScript (JS):**

JavaScript is a versatile and powerful programming language that is primarily used for adding interactivity and dynamic behavior to websites. It is a key technology for front-end web development and is supported by all major web browsers.



Fig. 3.4; JS Logo

* **The key features is given below:**
* **Client-Side Scripting:** JavaScript is a client-side scripting language, meaning it runs on the user's browser rather than on the web server. This enables developers to create dynamic and interactive user interfaces.
* **Variables and Data Types:** JavaScript allows the declaration of variables using var, let, or const. It supports various data types, including numbers, strings, booleans, arrays, objects, and more.
* **Functions:** Functions in JavaScript allow the encapsulation of code into reusable blocks. They can be defined using the function keyword.
* **DOM Manipulation:** The Document Object Model (DOM) represents the structure of an HTML document as a tree of objects. JavaScript can manipulate the DOM to dynamically update the content and structure of a webpage.
* **Events:** JavaScript allows you to respond to user actions, such as clicks, keypresses, and mouse movements, by attaching event listeners to HTML elements.
* **Asynchronous Programming:** JavaScript supports asynchronous programming using features like callbacks, Promises, and the async/await syntax.
* **Conditional Statements and Loops:** JavaScript includes traditional programming constructs like if statements for conditional logic and for and while loops for repetitive tasks.
* **Object-Oriented Programming (OOP):** JavaScript supports object-oriented programming principles. Objects and classes allow developers to create and organize code in a more modular and reusable way.

**3.9 Node.js:**

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.



Fig. 3.5 Node.js Logo

This is in contrast to today's more common concurrency model, in which OS threads are employed. Thread-based networking is relatively inefficient and very difficult to use. Furthermore, users of Node.js are free from worries of dead-locking the process, since there are no locks. Almost no function in Node.js directly performs I/O, so the process never blocks except when the I/O is performed using synchronous methods of Node.js standard library. Because nothing blocks, scalable systems are very reasonable to develop in Node.js.

Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop. Typically, behaviour is defined through callbacks at the beginning of a script, and at the end a server is started through a blocking call like Event Machine::run(). In Node.js, there is no such start-the-event-loop call. Node.js simply enters the event loop after executing the input script. Node.js exits the event loop when there are no more callbacks to perform. This behaviour is like browser JavaScript — the event loop is hidden from the user.

HTTP is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Node.js well suited for the foundation of a web library or framework.

Node.js being designed without threads doesn't mean you can't take advantage of multiple cores in your environment. Child processes can be spawned by using our child\_process.fork() API, and are designed to be easy to communicate with. Built upon that same interface is the cluster module, which allows you to share sockets between processes to enable load balancing over your cores.

**3.10 Mongo Db:**

MongoDB is a leading NoSQL database that has gained widespread popularity for its flexibility, scalability, and performance. Unlike traditional relational databases, MongoDB stores data in flexible, JSON-like documents, making it well-suited for handling unstructured or semi-structured data. One of MongoDB's key features is its ability to scale horizontally across multiple servers, allowing for seamless expansion as data volumes grow. Additionally, MongoDB supports dynamic schema design, enabling developers to iterate quickly and adapt to changing application requirements without sacrificing performance. With its rich querying capabilities and support for complex data structures, MongoDB has become the database of choice for a wide range of applications, from content management systems and e-commerce platforms to real-time analytics and Internet of Things (IoT) applications. Overall, MongoDB empowers developers to build modern, data-driven applications that can scale effortlessly to meet the demands of today's rapidly evolving digital landscape.



Fig. 3.6 Mongo Db Logo

MongoDB stands as a pioneering force in the realm of database technology, offering a versatile solution tailored to the demands of modern application development. Its document-oriented approach, where data is stored in JSON-like documents, fosters agility and adaptability, allowing developers to seamlessly iterate on their data models as project requirements evolve. MongoDB's distributed architecture ensures robustness and scalability, enabling applications to effortlessly scale out across multiple nodes to accommodate growing workloads and user bases. Moreover, MongoDB's comprehensive suite of features, including advanced querying capabilities, geospatial indexing, and full-text search, empowers developers to craft sophisticated applications that harness the full potential of their data. With its commitment to innovation and a vibrant community driving continuous improvement, MongoDB continues to shape the landscape of data management, powering the mission-critical applications of today and tomorrow.

**Chapter-4**

**Methodology**

**4.1 Methodology of Project:**

The development methodology for Electro-Mart, an online platform tailored for Electronic and Communication Engineering (ECE) enthusiasts, students, and professionals, follows a well-structured approach designed to effectively address user needs:

* **Requirements Analysis**: Conducted surveys, interviews, and competitor analysis.

Identified core features such as a product catalog, book listings, user accounts, and email notifications.

* **Design and Planning**: Created wireframes to visualize the user interface.

Planned the MongoDB database schema. Outlined the website structure using HTML, CSS, and JavaScript.

* **Development**: Developed the frontend and backend using technologies like Node.js and MongoDB. Integrated email notifications through SMTP setup.
* **Testing**: Ensured platform functionality across various devices.
* **Deployment**: Set up a production server environment. Deployed the code to make the platform live..

**4.2 Flow Chart of the Project:**

Fig.4.1; Flow Chart of Project

**Chapter-5**

**Result & Discussions**

**5.1 Overview of the Results:**

* Briefly summarize the main functionalities of the ECE shopping website, including user signup, login, product browsing, adding items to cart, and purchasing.
* Highlight the seamless flow from account creation to website navigation and shopping experience.

**5.2 User Signup Process:**

The signup process is important because it lets users create accounts on the ECE components website, giving them access to features like purchasing. The signup form is simple and asks for basic details like a username, email, and password. It is designed to be easy to use, with clear instructions and helpful error messages to guide users if they make mistakes. This makes the signup process smooth and user-friendly, encouraging more people to sign up. Using Node.js for the backend, the form data is securely sent to the server, where it is saved in a MongoDB database, making sure that user accounts are created quickly and safely.

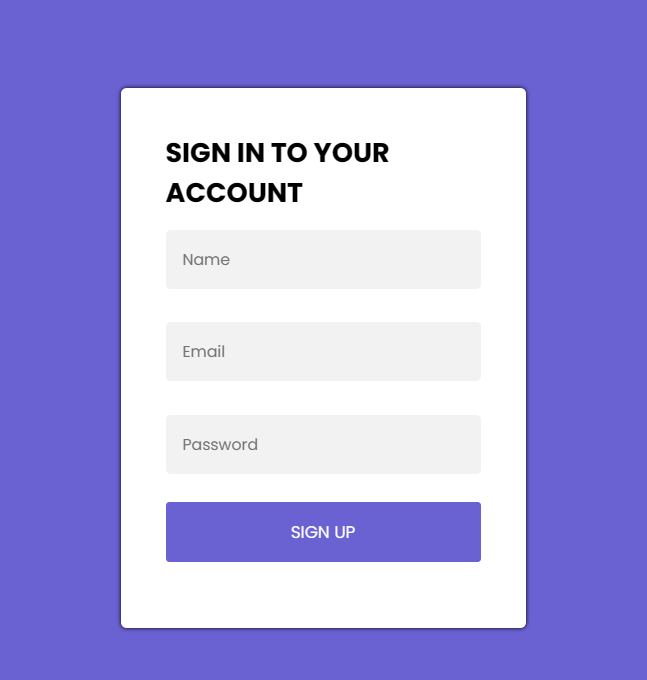


Fig. 5.1; Website Signup Process

**5.3 User Login Process:**

To log in to their accounts on Electro-Mart, users go to the login page. Here, they see boxes where they type their username or email and password. After they type their info and click "Login," the website checks if the info is correct. It makes sure the username or email and password match what's saved in the website's database. If everything matches up, they're allowed in. This keeps their account safe and makes sure only theright person can get in.

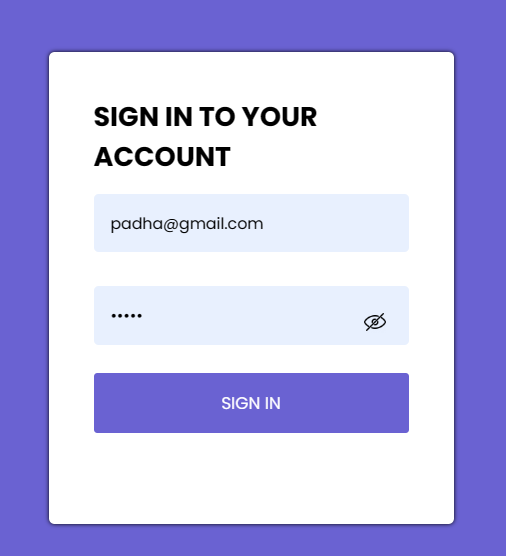


Fig 5.2; User Login Process

**5.4 Website Navigation and Shopping Experience:**

After logging in successfully to Electro-Mart, users immediately see the main page with all the products neatly sorted into different categories. They can easily browse through these categories to find what they're looking for. When they find something they want to buy, like a component or a book, they just click a button to add it to their shopping cart. The shopping cart shows all the items they've selected, and they can review and change their choices if they want. When they're ready, they go to the checkout, where they enter their payment and shipping details to complete their purchase. This process makes shopping on Electro-Mart easy and convenient for users.



Fig. 5.3 Website Home Page

**5.5 Implementation of the Project:**

The signup, login, and website navigation features were implemented using a combination of HTML, CSS, JavaScript, and Node.js. For signup and login functionalities, HTML forms were utilized to capture user input, while CSS was employed to style these forms and create an appealing visual interface. JavaScript was leveraged to add interactivity to the forms, such as client-side validation to ensure that users provide valid input before submission. On the server-side, Node.js was used to handle user authentication, manage user sessions, and interact with the MongoDB database to store and retrieve user account information securely. Additionally, third-party libraries such as Express.js were employed to streamline server-side development, providing a robust framework for building web applications with Node.js. These technologies were integrated cohesively to create a seamless user experience, allowing users to sign up, log in, and navigate the website effortlessly.

**Chapter-6**

**Applications & Advantages**

**6.1 Applications:**

* **Schools and Colleges:** Helps students and teachers book electronic parts and related books for their studies and projects.
* **Researchers:** Assists researchers in finding and booking components and books needed for their experiments.
* **Hobbyists:** Makes it easy for electronics hobbyists to get the parts and guides they need for their projects.
* **Workshops:** Helps workshops and labs manage and book their electronic components and reference materials.
* **Online Courses:** Supports online learning by providing an easy way for students to book the parts and books they need.
* **Libraries:** Assists libraries in managing and lending out electronic components and related books.
* **Professionals:** Helps electronics professionals quickly book the parts and reference materials they need for their work.

**6.2 Advantages:**

* **Convenience:** Users can easily book components and related books online without needing to visit a physical store or library.
* **Time-Saving:** Streamlines the process of finding and reserving necessary items, saving users time.
* **Organization:** Provides a structured system for managing bookings and inventory, making it easier to handle multiple requests.
* **Resource Availability:** Ensures that the needed components and books are available when users need them, enhancing productivity and efficiency.
* **User-Friendly:** Simplifies the booking process with an intuitive interface, making it easy for users of all skill levels to navigate.
* **Centralized Information:** Combines information about components and related books in one place, making it easier to find and book what is needed.
* **Improved Planning:** Users can plan their projects better knowing they can reliably book the components and books they need.
* **Educational Support:** Enhances the learning experience by providing quick access to essential resources for students and educators.

**Chapter-7**

**Conclusion & Future Scope**

**7.1 Conclusion:**

At Electro-mart, we're dedicated to providing a comprehensive solution for electronics enthusiasts and professionals alike. In addition to offering a vast array of ECE components, we also provide access to a curated selection of books related to electronics and communication engineering. From textbooks covering fundamental concepts to advanced reference materials, our book collection is designed to support your learning journey and enhance your understanding of ECE principles. Whether you're a hobbyist working on a DIY project or an engineer seeking components for your latest innovation, Electro-mart is your one-stop destination for all things ECE. With our intuitive platform, seamless shopping experience, and commitment to quality, Electro-mart is your ultimate destination for all things ECE. Whether you're stocking up on components for your latest project or expanding your knowledge with insightful books, we're here to meet your needs.

**7.2 Future Scope:**

In the future, our vision for Electro-mart is to grow and improve in several ways. We're excited about the possibility of introducing new features that will make learning about electronics easier and more enjoyable for everyone. This could include things like interactive tutorials and online courses designed specifically for those interested in electronics and communication engineering. We also see potential in expanding our library of resources by collaborating with publishers to offer a wider selection of books covering various aspects of electronics. By continuously evolving and innovating, we aim to make Electro-mart the ultimate destination for individuals passionate about electronics. Our goal is to cultivate a welcoming and supportive community where people can come together to learn, explore, and create exciting projects. With these future developments, we're committed to ensuring that Electro-mart remains a valuable resource for enthusiasts and professionals alike, empowering them to pursue their interests and achieve their goals in the world of electronics.

**References**

[1] Rahul Semil , “Web Page Designing using Html,Css and JavaScript” International Research Journal of Modernization in Engineering Technology and Science Volume 4, Issue 5, 2022.ISSN 2582-5208

[2] Avnish Kumar, “Big Buy (E-Commerce Website) by using Frontend Web Development” International Journal for Modern Trends in Science and Technology, Volume 7, Issue 11, 2021.ISSN: 2455-3778

[3] Ashis Kumar Ratha and Shibani Sahu, “Html5 in Web Development” International Research Journal of Engineering and Technology, Volume 5, Issue 3, 2018.ISSN 2395-0056

[4] Mishra, S. V., & Kotkar, D. S. (2015, February). A Study on Current Status of E-Commerce in India: A Comparative Analysis of Flipkart and Amazon. International Journal of Advance Research in Computer Science and Management Studies, 3(2), (pp.133-137.)

[5] Gupta, A. (2014, January). E-Commerce: Role Of E-Commerce In Today\'s Business. International Journal of Computing and Corporate Research.

[6] Raghunath, A., & Panga, M. D. (2013). Problem and Prospects of E-Commerce. International Journal of Research and Development - A Management Review, 2(1), (pp.59-68).

[7] Chen Li-Li, Liu Zheng-Long, Design of Rich Client Web Architecture Based o HTML5, ICCIS, 2012

[8] Wenling Hu, Hao Yuan, Jiangong Wang, Liang Wang, The Research and Application of Power System Visualization Based on HTML, IEEE 2011.

[9] Ray, S. (2011). Emerging Trend of E-Commerce in India: Some Crucial Issues, Prospects and Challenges. Computer Engineering and Intelligent Systems, (pp.18-36.)

[10] Gunasekaran, A., Marri, H., McGaughey, R., & Nebhwani, M. (2002). E-commerce and its impact on operations management. International Journal of Production Economics, (pp.185-197.)

[11] HTML: [**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

[12] CSS: [**https://www.w3schools.com/css/default.asp**](https://www.w3schools.com/css/default.asp)

[13] JS: [**https://www.w3schools.com/js/default.asp**](https://www.w3schools.com/js/default.asp)

**Appendix**

**Project Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Booking\_Page</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="style.css">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">

</head>

<body>

<nav class="navbar bg-primary" data-bs-theme="dark">

<a href="./index.html">Home</a>

<button id="go\_to\_Bok\_section">Books</button>

<a href="./combos.html" >Combos</a>

<a href="./cart.html">

<i>Cart</i>

</a>

</nav>

<img src="./Ard\_BOOK.jpg" alt="" class="centerR\_BOOK\_images">

<p id="book\_1\_text">Empower beginners with a concise introduction to

Arduino, demystifying concepts and facilitating easy grasp of fundamentals.</p>

</div>

<div class="buttons" id="ARd\_Book">

<p id="pr1">Price: ₹1599.00</p>

<button onclick="addToCart('Arduino Book for Beginners',1599.00)">Add to Cart</button><br>

<button id="buyButton">Buy Now</button>

</div>

</div>

</div>

<div class="book\_text"> Discover the world of Arduino—from

basics to advanced techniques. With step-by-step tutorials

and practical examples, master Arduino programming and hardware

interfacing. Unleash your creativity with

sensor integration, wireless communication, and IoT applications. Order now and elevate your DIY .</div>

</div>

<div class="content">

<div class="book\_1 \_2">

<div class="image-container">

<img src="./Nano.jpeg" alt="" class="centerR\_BOOK\_images">

<p id="book\_1\_text">

Demystify Arduino-Nano for beginners with clarity, easing them into the fundamentals effortlessly..</p>

</div>

<div class="buttons" id="ARd\_Book">

<p id="pr1">Price: ₹1299.00</p>

<button onclick="addToCart('Arduino Nano for Beginners',1299.00)">Add to Cart</button><br>

<button id="buyButton">Buy Now</button>

</div>

</div>

</div?>

<div class="book\_text"> Unleash creativity with Arduino Nano—explore tutorials,

examples, and tips for mastering programming and hardware. Featuring expert contributions, it's an

essential resource for makers of all levels. Elevate your skills with the Arduino Nano book today!

</div>

let cart = JSON.parse(localStorage.getItem('cart')) || [];

let totalAmount = 0;

// Function to display cart items

function displayCart() {

let cartItemsContainer = document.getElementById('cartItems');

let totalAmountContainer = document.getElementById('totalAmount');

cartItemsContainer.innerHTML = ''; // Clear previous content

totalAmountContainer.innerHTML = ''; // Clear previous total amount

let totalAmount = 0;

cart.forEach((item, index) => {

let itemElement = document.createElement('div');

itemElement.textContent = `${item.productName} - ₹${item.price}`;

// Add a delete button for each item

let deleteButton = document.createElement('button');

deleteButton.textContent = 'Delete';

deleteButton.addEventListener('click', () => deleteItem(index));

itemElement.appendChild(deleteButton);

cartItemsContainer.appendChild(itemElement);

totalAmount += item.price;

});

totalAmountContainer.textContent = `Total Amount: ₹${totalAmount.toFixed(2)}`;

}

// Function to delete an item from the cart

function deleteItem(index) {

cart.splice(index, 1);

localStorage.setItem('cart', JSON.stringify(cart));

displayCart();

}