**E-PRINT**

**A PROJECT REPORT**

***Submitted by***

**PATHANAZAMKHAN.M.**

**191260116045**

***In partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**in**

**INFORMATION TECHNOLOGY**

**SAL ENGINEERING AND TECHNICAL INSTITUTE, AHMEDABAD**

****

**Gujarat Technological University, Ahmedabad**

**MAY 2023**

****

**SAL EDUCATION**

**SAL ENGINEERING AND TECHNICAL INSTITUTE**

**Opp. Science City,**

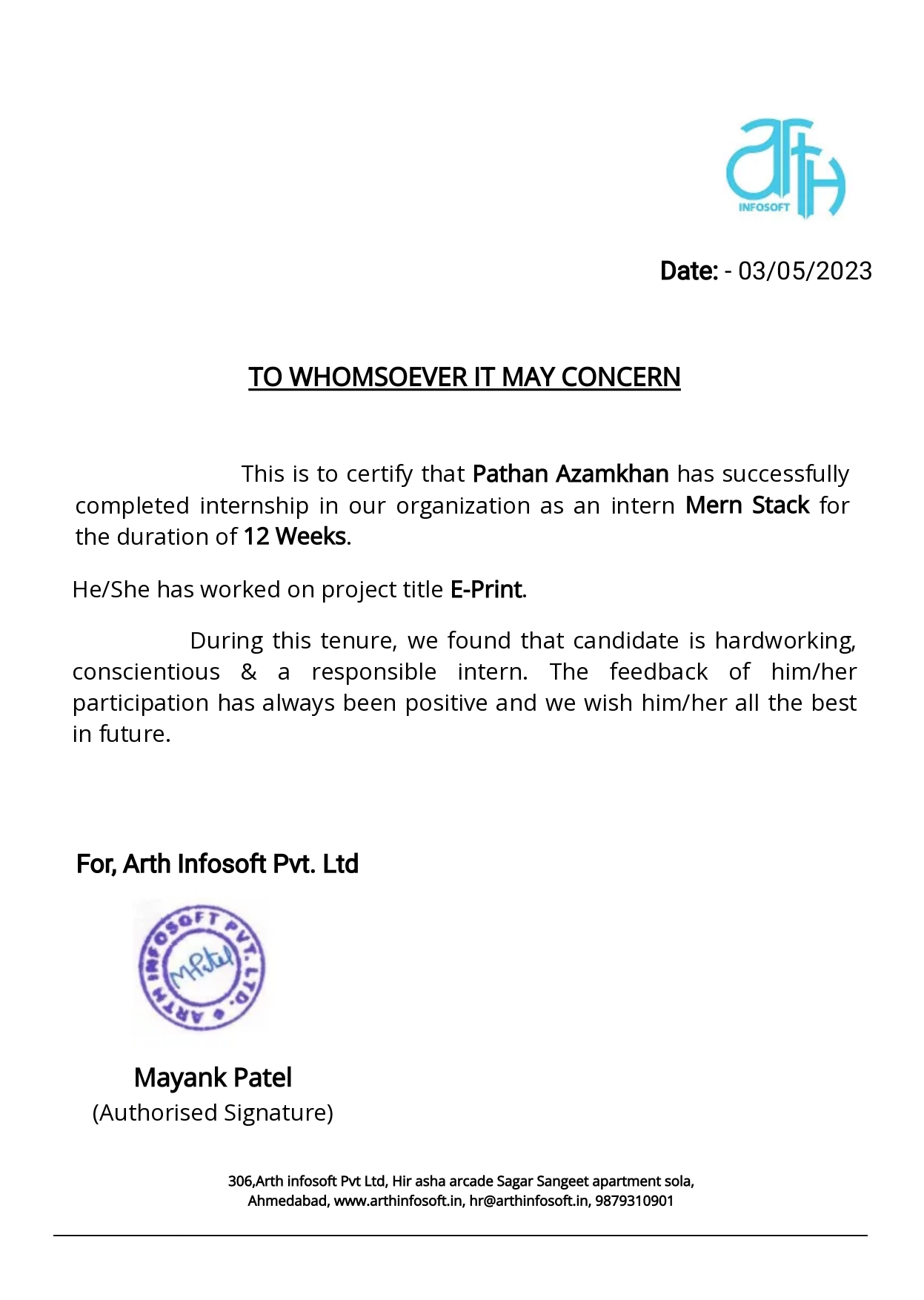
**Sola Bridge Road, Ahmedabad, Gujarat-380060**

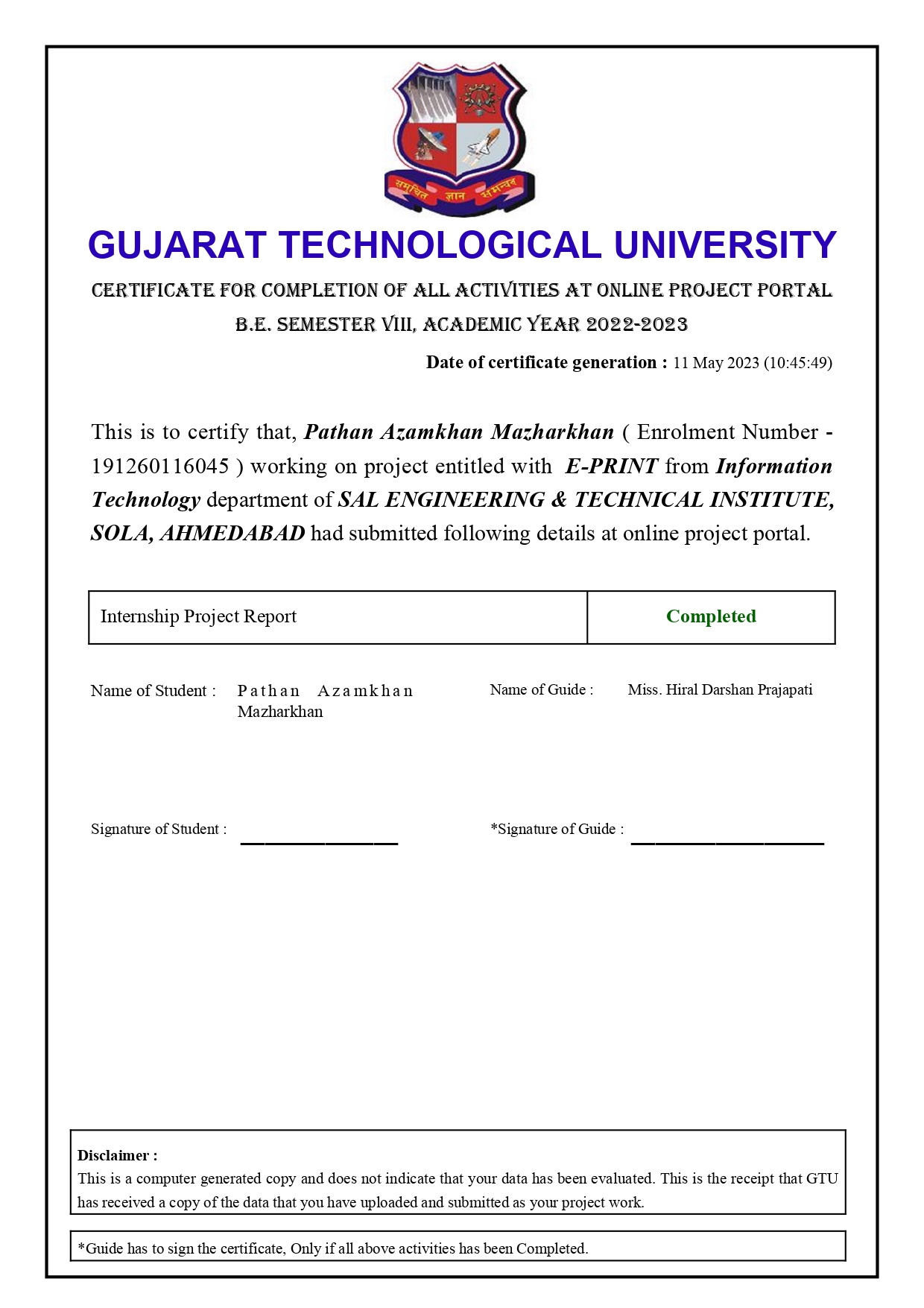
**CERTIFICATE**

This is to certify that the project report submitted along with the project entitled **E-PRINT** has been carried out by **PATHANAZAMKHAN.M.** **(191260116045)** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in **INFORMATION TECHNOLOGY**, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year **2022-23**.

Prof. Hiral Prajapati Dr. Ajay N. Upadhyaya

**Internal guide Head of the Department**







****

**SAL ENGINEERING AND TECHNICAL INSTITUTE**

**Opp. Science City,**

**Sola Bridge Road, Ahmedabad, Gujarat-380060**

**DECLARATION**

We hereby declare that the Internship report submitted along with the Internship entitled **MongoDB, Express JS, React JS, Node JS** submitted in partial fulfillment for the degree of Bachelor of Engineering In Information Technology to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me at Arth Infosoft Pvt. Ltd. under the supervision of Prof. Hiral Prajapati and that no part of this report has been directly copied from any student’s reports or taken from any other source, without providing due reference.

**Name of the Student Sign of Student**

**PATHANAZAMKHAN.M.**

# **Acknowledgement**

I would like to express my deepest gratitude and appreciation to the individuals who have supported and guided me throughout my journey. Their mentorship, encouragement, and expertise have played a crucial role in shaping my professional development and personal growth.

First and foremost, I would like to extend my sincere thanks to Prof. Hiral Prajapati, my esteemed mentor. Your unwavering support, insightful guidance, and wealth of knowledge have been instrumental in my success. Your dedication to excellence, passion for learning, and commitment to nurturing your mentees have been truly inspiring. I am grateful for the opportunities you have provided me and for believing in my potential.

I would also like to acknowledge the invaluable support and encouragement I have received from the team at Arth Infosoft Pvt Ltd, the company where I have had the privilege to work. I am thankful to the entire team for creating a conducive and collaborative work environment that fosters growth, creativity, and innovation. Special thanks to my colleagues for their camaraderie and cooperation throughout my time with the company.

Furthermore, I would like to express my gratitude to Prof. Ajay Upadhyaya, my Head of Department. Your leadership, expertise, and guidance have been invaluable in shaping my academic and professional journey. Your unwavering commitment to excellence and passion for imparting knowledge have had a profound impact on my development.

Lastly, I would like to extend my heartfelt appreciation to all the friends, family members, and well-wishers who have supported me throughout this endeavour. Your constant encouragement, love, and belief in my abilities have been a source of strength and motivation.

PATHANAZAMKHAN.M.

# **Abstract**

Sometimes you come across small problems where a person or a shop keeper need to print some files, visiting card, letter heads book covers, Marriage invitation card or to print logo on T-shirt etc.

This software allows user to login in into our system, select predefined template and order for a product. If user want customize design, he/she can give order for a design. Also, after approval of design by user, product will start getting printed and delivered to that user.

In this project user will able to upload custom design for office files, visiting card, and, letterhead etc. Once design approved by company, printing process will start on it. Once it’s finished user will get final product delivered.

# **List of Figures**

|  |  |  |
| --- | --- | --- |
| **SR NO** | **FIGURE NAME** | **PAGE NO.** |
| Fig 2.2.1 | Schematic Layout | 4 |
| Fig 3.6.1 | Prototype Model | 10 |
| Fig 3.7.1 | Project Scheduling | 13 |
| Fig 5.1.1 | ER Diagram | 23 |
| Fig 5.1.2 | Use Case Diagram | 26 |
| Fig 5.1.3 | User Activity Diagram | 28 |
| Fig 5.1.4 | Admin Activity Diagram | 29 |
| Fig 5.1.5 | Vendor Activity Diagram | 30 |
| Fig 5.1.6 | Class Diagram | 33 |
| Fig 5.1.7 | Vendor Sequence Diagram | 40 |
| Fig 5.1.8 | Admin Sequence Diagram | 40 |
| Fig 5.1.9 | User Sequence Diagram | 41 |
| Fig 6.2.1 | Home page Code | 46 |
| Fig 6.2.2 | Admin Dashboard Code | 47 |
| Fig 6.2.3 | Vendor Dashboard Code | 47 |
| Fig 6.2.4 | Search Product/Category Code | 48 |
| Fig 6.3.1 | Home Page | 49 |
| Fig 6.3.2 | Cart Page | 50 |
| Fig 6.3.3 | Search Products Page | 50 |
| Fig 6.3.4 | Search Category Page | 51 |
| Fig 6.3.5 | Profile Page | 51 |
| Fig 6.3.6 | Vendor Dashboard Page | 52 |
| Fig 6.3.7 | Admin Dashboard Page | 52 |

# **List of Tables**

|  |  |  |
| --- | --- | --- |
| **SR NO** | **TABLE NAME** | **PAGE NO.** |
| Table 5.1.1 | Symbols and Components of ER Diagram | 22 |
| Table 5.1.2 | Symbols and components of Use Case Diagram | 25 |
| Table 5.1.3 | Role | 35 |
| Table 5.1.4 | User | 35 |
| Table 5.1.5 | Category | 36 |
| Table 5.1.6 | State | 36 |
| Table 5.1.7 | City | 36 |
| Table 5.1.8 | vendor\_detail | 36 |
| Table 5.1.9 | Product | 37 |
| Table 5.1.10 | vendor\_product | 37 |
| Table 5.1.11 | Cart | 38 |
| Table 5.1.12 | vendor\_product\_images | 38 |
| Table 7.2.1 | Test Case | 54 |

# **TABLE OF CONTENTS**

[Acknowledgement vi](#_Toc134573281)

[Abstract vii](#_Toc134573282)

[List of Figures viii](#_Toc134573283)

[List of Tables ix](#_Toc134573284)

[TABLE OF CONTENTS x](#_Toc134573285)

[**Chapter - 1 Overview of the Company** 1](#_Toc134573286)

[1.1 History 1](#_Toc134573287)

[1.2 Different product / scope of work 1](#_Toc134573288)

[**Chapter - 2 Overview of process being carried out in company** 3](#_Toc134573289)

[2.1 It includes the details about the work being carried out in each department. 3](#_Toc134573290)

[2.2 Prepare schematic layout which shows the sequence of operation for manufacturing of end product. 4](#_Toc134573291)

[2.3 Explain in details about each stage of production. 4](#_Toc134573292)

[**Chapter - 3 Introduction to Project** 6](#_Toc134573293)

[3.1 Project / Internship Summary 6](#_Toc134573294)

[3.2 Purpose 6](#_Toc134573295)

[3.3 Objective 6](#_Toc134573296)

[3.4 Scope (what it can do and can’t do) 7](#_Toc134573297)

[3.5 Technology and Literature Review 8](#_Toc134573298)

[3.6 Project Planning 10](#_Toc134573299)

[3.6.1 Internship Development Approach and Justification 10](#_Toc134573300)

[3.7 Project Scheduling (Gantt Chart/PERT/Network Chart) 13](#_Toc134573301)

[**Chapter - 4 System Analysis** 14](#_Toc134573302)

[4.1 Study of Current System 14](#_Toc134573303)

[4.2 Problem and Weaknesses of Current System 14](#_Toc134573304)

[4.3 Requirements of New System 15](#_Toc134573305)

[4.4 System Feasibility 15](#_Toc134573306)

[4.4.1 Does the system contribute to the overall objectives of the organization? 15](#_Toc134573307)

[4.4.2 Can the system be implemented using the current technology and within the given cost and schedule constraints? 16](#_Toc134573308)

[4.4.3 Can the system be integrated with other systems which are already in place? 16](#_Toc134573309)

[4.5 Process in New System 16](#_Toc134573310)

[4.6 Features of New System 17](#_Toc134573311)

[4.7 List Main Modules 17](#_Toc134573312)

[4.8 Selection of Hardware / Software / Algorithms / Methodology / Techniques / Approaches and Justification 18](#_Toc134573313)

[**Chapter - 5 System Design** 21](#_Toc134573314)

[5.1 System Design & Methodology 21](#_Toc134573315)

[5.1.1 ER Diagram: 21](#_Toc134573316)

[5.1.2 Use Case Diagram: 24](#_Toc134573317)

[5.1.3 Activity Diagram: 27](#_Toc134573318)

[5.1.4 Class Diagram: 31](#_Toc134573319)

[5.1.5 Data Dictionary Diagram: 34](#_Toc134573320)

[5.1.6 Sequence Diagram: 39](#_Toc134573321)

[**Chapter - 6 Implementation** 42](#_Toc134573322)

[6.1 Environment 42](#_Toc134573323)

[6.2 Modules Specification(s) 46](#_Toc134573324)

[6.3 Project Screenshot and Sneak peek 49](#_Toc134573325)

[**Chapter - 7 Testing** 53](#_Toc134573326)

[7.1 Testing Plan / Strategy 53](#_Toc134573327)

[7.2 Test Results and Analysis 54](#_Toc134573328)

[7.2.1 Test Cases (test ID, test condition, expected output, actual output, remark) 54](#_Toc134573329)

[**Chapter - 8 Conclusion and Discussion** 57](#_Toc134573330)

[8.1 Project Viabilities 57](#_Toc134573331)

[8.2 Problem Encountered and Possible Solutions 57](#_Toc134573332)

[8.3 Summary of Internship 58](#_Toc134573333)

[8.4 Limitation and Future Enhancement 59](#_Toc134573334)

[**REFRENCES** 61](#_Toc134573335)

# **Chapter - 1 Overview of the Company**

# **1.1 History**

Arth Infosoft Pvt. Ltd. has Indian established IT growing company, aims to serve with affordable Graphic, Web and Software solutions. Arth specializes in particular, but not limited to graphic, website designing, web application and web portal development. Moreover, we are also ready to offer your business or services to a wider audience through our strong SEO (Search Engine Optimization) campaigns. Our online reputation is founded on the ability to deliver web services in Word press, Magento, PHP, Drupal, Joomla and much more with custom solutions based on client’s requirement. We make every effort the long run clientele and hence 24X7 supports are not only our strategy but our business value.

Arth comprises of a qualified and experienced team of creative web designers, developers and software programmers. The company abides by systematic and timely execution of all the small to the large range of web assignments.

# **1.2 Different product / scope of work**

We at Arth Infosoft with more than decades of experience in hundreds of projects in various industries as like: education, health, manufacturing, engineering, hospitality, consulting and utilities are well equipped to handle all these challenges for you. With expertise, teams possess vast experience and object-oriented development methods and we have handled clients globally with upgrading and customizing their present applications with the latest technologies and also create a conceptual software application for their industry.

Some of Our Remarkable Custom Application Development:

eduExam - Online Examination Software  
This software used to conduct online exam most to medical institution and private tuition organizations.

eduWare - School Management Software

This application developed to help educational institution to manage their daily activity and entire administration. This is complete school administration software.

EduSalary is the best Time Attendance system for any type of organisation. Our user-friendly and cost-effective time & attendance system with integrated software applications will give your company insight into employee labor patterns, save money on direct and indirect labor costs, and give you the competitive edge. As the name suggests, time attendance software can accurately track hours logged and paid time off.

# **Chapter - 2 Overview of process being carried out in company**

# **2.1 It includes the details about the work being carried out in each department.**

1. Web Designing: Web designing is the process of creating and designing the visual elements and aesthetics of a website, such as the layout, colours, graphics, fonts, and images. It involves using various design tools and software to create an attractive and user-friendly website that meets the client's needs and requirements.

2. WordPress Development: WordPress is a popular open-source content management system that allows users to easily create and manage websites and blogs. WordPress development involves creating and customizing WordPress themes, plugins, and other features to create a unique and functional website that meets the client's needs.

3. SEO Services: SEO (Search Engine Optimization) is the process of optimizing a website to improve its visibility and ranking on search engine results pages (SERPs). SEO services involve a range of techniques and strategies, such as keyword research, on-page optimization, link building, and content creation, to help a website rank higher on search engines and attract more organic traffic.

4. Software Development: Software development is the process of designing, creating, testing, and maintaining software applications and systems. It involves using various programming languages, tools, and frameworks to create custom software solutions that meet the client's needs and requirements.

5. Mobile Development: Mobile development is the process of creating and designing mobile applications for smartphones and other mobile devices. It involves using various mobile development platforms, such as iOS and Android, to create custom mobile applications that meet the client's needs and requirements.

# **2.2 Prepare schematic layout which shows the sequence of operation for manufacturing of end product.**

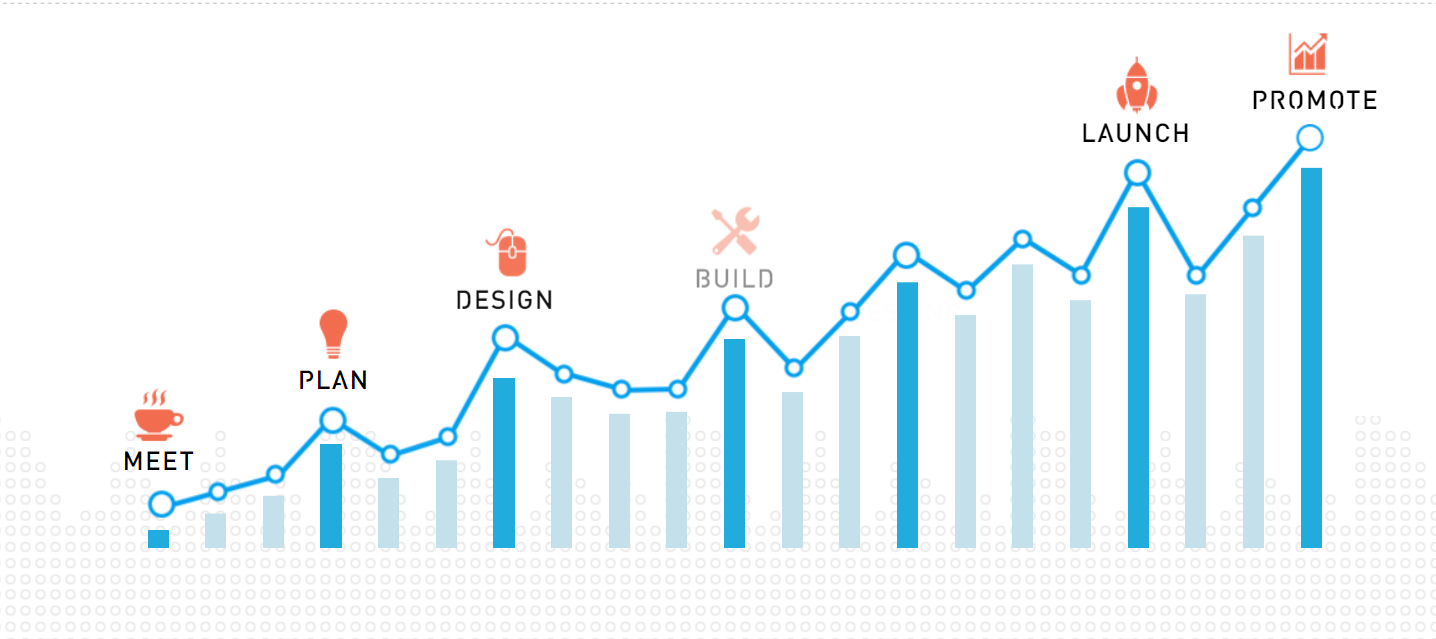
****

Fig 2.2. Schematic Layout

# **2.3 Explain in details about each stage of production.**

* Meet: In this phase, the project team conducts meetings with stakeholders to gather information about the project's requirements. This information is used to define the project's scope, objectives, and deliverables.
* Plan: After gathering requirements, the project team develops a plan for the project. This includes a timeline, budget, resource allocation, and risk management plan. The planning phase also involves defining the system architecture and designing the user interface.
* Design: In the design phase, the project team creates a detailed technical design based on the requirements gathered in the first phase. The design includes data structures, algorithms, and system interfaces.
* Build: In this phase, the project team develops the software code based on the technical design. The build phase involves writing and testing the code, as well as creating and testing any necessary database structures.
* Testing: Once the build phase is complete, the project team tests the software to ensure that it meets the requirements and is free of bugs. Testing can include functional testing, performance testing, security testing, and usability testing.
* Launch: After the testing phase is complete, the project team launches the software. This involves deploying it to the production environment and making it available to end-users.
* Promote: Finally, in the promotion phase, the project team works on promoting the software to end-users. This can include marketing, training, and support activities to ensure that users are aware of the software's features and are able to use it effectively.

# **Chapter - 3 Introduction to Project**

# **3.1 Project / Internship Summary**

Our project is a web-based platform that allows users to easily create and order high-quality personalized photo products such as photo books, prints, cards, and gifts. Once a user has created their desired product, they can preview and order it online.

We strive to provide our users with an easy-to-use and intuitive interface, as well as exceptional customer service and support. Our goal is to help users preserve and share their cherished memories in a creative and meaningful way.

# **3.2 Purpose**

A comprehensive online platform for users to store, organize, edit, and share their photos. The platform allows users to create personalized photo gifts such as photo books, cards, calendars, and other products using their photos.

The aim is to provide a user-friendly, easy-to-use platform that offers high-quality prints and products at affordable prices. The ultimate goal is to provide a one-stop-shop for all things photo-related, from storage and organization to printing and personalized gifting.

# **3.3 Objective**

An online platform that allows users to easily create and customize high-quality personalized photo products, such as photo books, cards, calendars, gifts, and wall art. The platform should provide a user-friendly interface with a variety of design options as well as the ability to upload photos from various sources.

The platform should also include features such as photo adding tools to help users create their perfect product. The objective should be to provide a seamless and enjoyable user experience, with fast and reliable shipping, high-quality products, and exceptional customer service.

Additionally, the objective should be to continually innovate and add new products and features to keep up with changing customer preferences and technology advancements. Overall, the objective of the project should be to become the go-to platform for personalized photo products, providing customers with a memorable and convenient way to preserve and share their memories.

# **3.4 Scope (what it can do and can’t do)**

* User Registration and Profile Management: Users can create accounts, edit their profile information, and manage their personal preferences.
* Photo Upload and Management: Users can upload and organize their photos into albums, edit their photos, and add captions and tags to make them searchable.
* Product Creation: Users can create a variety of products such as photo books, cards, calendars, prints, wall art, and gifts using their uploaded photos.
* Product Customization: Users can customize their products by selecting layouts, backgrounds, and design elements, as well as add text, stickers, and embellishments.

The project is not be able to provide physical printing services, such as running a printing press or creating physical copies of digital products. Additionally, it may not be able to provide design or photography services, such as photo retouching, graphic design, or professional photography. It also may not be able to offer custom framing or matting services for printed products.

# **3.5 Technology and Literature Review**

**Here is a technology review of the Advance Billing System with QR:**

**1. HTML:**

* Stands for Hyper Text Markup Language
* Used for creating the structure and content of web pages
* Consists of a series of tags and attributes that define the layout and functionality of a webpage

**2. CSS:**

* Stands for Cascading Style Sheets
* Used for styling the appearance of HTML elements on a web page
* Allows for separation of presentation and content
* Uses selectors to target specific elements and apply styling rules

**3. JAVASCRIPT:**

* A programming language used for adding interactivity to web pages
* Can manipulate HTML and CSS, and interact with backend servers and databases
* Supports event-driven programming, object-oriented programming, and functional programming
* Can be run in a browser or on a server with Node.js

**4. REACT:**

* A JavaScript library for building user interfaces
* Uses a component-based architecture for building reusable UI elements
* Enables dynamic and responsive web pages without the need for full page refreshes
* Often used in combination with other libraries and frameworks to create full-stack web applications

**5. NODE JS:**

* A JavaScript runtime environment that allows for running JavaScript on the server side
* Built on the Chrome V8 JavaScript engine
* Provides access to the file system, network, and other operating system functionalities
* Often used for building scalable, high-performance web applications

**6. REST API:**

* Stands for Representational State Transfer Application Programming Interface
* A set of guidelines and principles for building web services
* Uses HTTP requests to GET, POST, PUT, and DELETE data
* Supports interoperability between different systems, programming languages, and platforms

**7. EXPRESS JS:**

* A popular Node.js web framework
* Provides a set of features and tools for building web applications and APIs
* Enables the creation of routes, middleware, and templates
* Supports a wide range of plugins and extensions for additional functionality

**8. MONGODB:**

* A document-oriented NoSQL database
* Stores data in JSON-like documents with dynamic schemas
* Enables easy scalability and high availability through horizontal scaling
* Supports flexible querying and indexing for efficient data retrieval’

**9. TAILWIND:**

* Tailwind is a CSS framework that provides a set of pre-defined utility classes to quickly and easily style web applications
* It allows developers to easily apply styling to HTML elements without having to write custom CSS from scratch.
* Tailwind is highly customizable and flexible, allowing developers to configure the framework to fit their specific needs.
* It is particularly popular among React developers, as it integrates seamlessly with the React framework.

# **3.6 Project Planning**

# **3.6.1 Internship Development Approach and Justification**

**Prototype Model:**

The prototype model, also known as the prototyping model, is a software development approach that emphasizes the creation of a working prototype as an early representation of the final product. It involves building a scaled-down version of the software system to gather user feedback, demonstrate key functionalities, and validate design decisions.

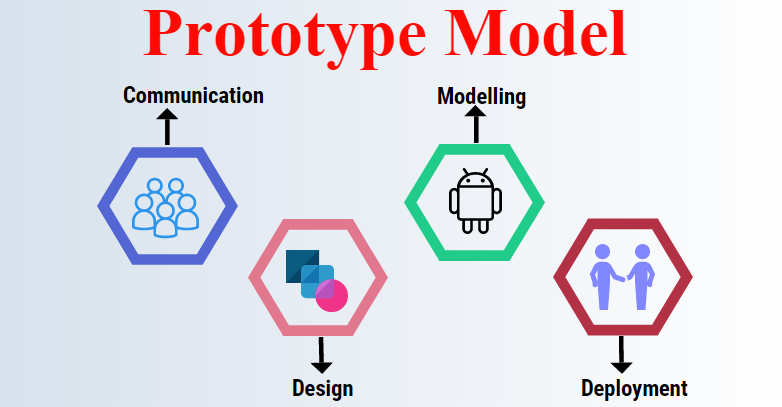


Fig 3.6. Prototype Model

**Here is a detailed explanation of the prototype model:**

Requirements Gathering: The process begins with gathering initial requirements from the stakeholders. These requirements may be in the form of high-level descriptions, user stories, or functional specifications.

Prototype Design: Based on the gathered requirements, a preliminary design of the software system is created. This design focuses on the core features and functionalities that need to be implemented in the prototype.

Prototype Development: The development team starts building the prototype using rapid development techniques. The emphasis is on creating a functional representation of the software system, often with limited or mocked data.

Prototype Evaluation: Once the prototype is ready, it is presented to the stakeholders, including end-users, for evaluation. The stakeholders provide feedback on the prototype's usability, features, and overall suitability for their needs.

Feedback Incorporation: The feedback received from stakeholders is analyzed, and necessary changes and improvements are identified. The prototype is then modified based on this feedback, and additional features or enhancements may be added.

Iterative Development: The development process continues in an iterative manner, with each iteration focusing on refining the prototype based on stakeholder feedback. Multiple iterations may be performed until the prototype meets the desired requirements and stakeholders' expectations.

Prototype Refinement: As the prototype evolves, it becomes more refined and closer to the final product. The iterative process helps in identifying and addressing any potential issues or challenges early on, reducing the risk of costly changes during later stages of development.

Final Product Development: Once the prototype is deemed satisfactory, the development team proceeds with the full-scale development of the software system, incorporating the lessons learned and enhancements made during the prototyping phase.

**Benefits of the Prototype Model:**

Early feedback: Users and stakeholders get a chance to interact with a working prototype and provide feedback, leading to better alignment with their requirements and expectations.

Reduced rework: The iterative nature of the model helps in identifying and resolving issues early on, reducing the need for major rework during later stages.

Improved communication: Prototypes serve as effective communication tools between developers, designers, and users, facilitating a better understanding of the system.

Risk mitigation: The prototype model allows for risk identification and mitigation at an early stage, minimizing the chances of costly errors or deviations from requirements.

**Limitations of the Prototype Model:**

Time and cost constraints: Developing prototypes and conducting iterative cycles can be time-consuming and costly, especially for complex software systems.

Incomplete requirements: If requirements are not properly gathered or analyzed before prototyping, there is a risk of developing a prototype that doesn't fully meet the stakeholders' needs.

Scope creep: Without proper control, there is a possibility of scope creep, where additional features and changes are continually requested during the prototyping phase, leading to delays and increased costs.

Overall, the prototype model is an effective approach when requirements are not fully defined or when user involvement and feedback are critical to the success of the software system. It allows for early validation, flexibility, and continuous improvement throughout the development process.

# **3.7 Project Scheduling (Gantt Chart/PERT/Network Chart)**

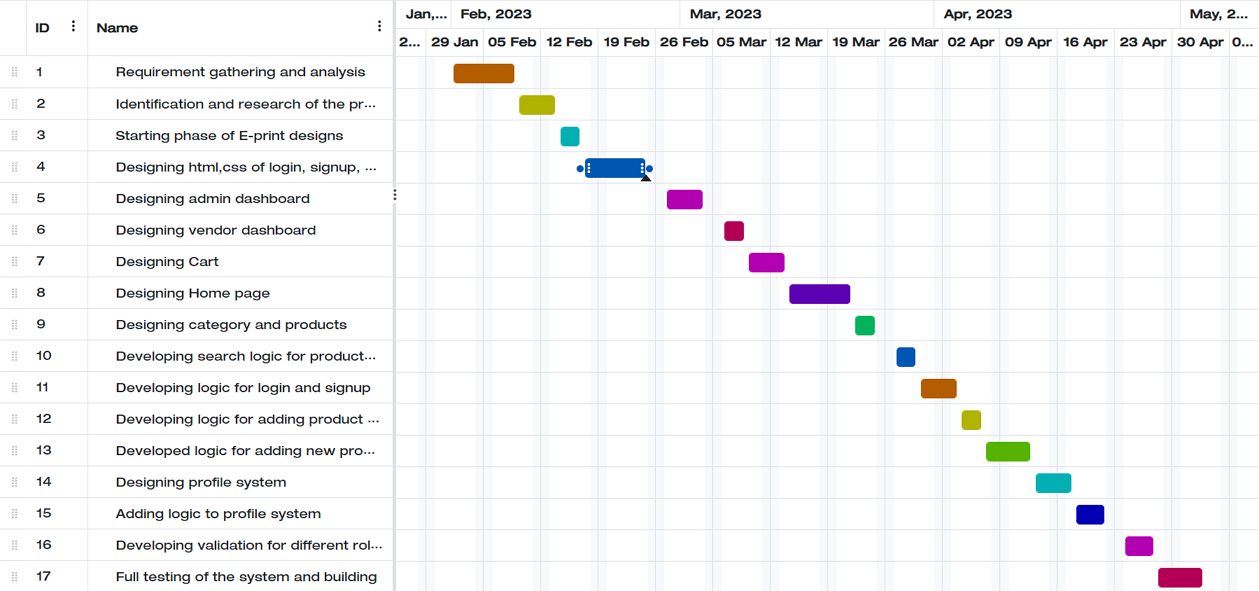


Fig 3.7. Project Scheduling

# **Chapter - 4 System Analysis**

# **4.1 Study of Current System**

Current system operates offline has a client application that the user installs on their device. This client application provides a user interface for the user to browse and select photos from their local device, as well as any cloud storage or network-attached storage devices that they have configured.

Once the user has selected the photos they want to print, the client application generate a print order file that contains all the necessary information for the printing process. This include details such as the quantity and size of the prints, any customizations or effects that the user has applied, and the shipping address and payment information.

The print order file is then transmitted to the offline system's print processing center, which is a physical location where the printing equipment is located. The print processing center then receives the print order file and use it to produce the physical prints. Once the prints are ready, they are shipped to the user's specified address.

Overall, the offline process of a system involve a combination of client-side software for selecting and ordering prints, as well as a centralized processing center for printing and fulfilment. This approach offers the benefits of offline access and control over the printing process, but also requires the user to install and manage client software on their device.

# **4.2 Problem and Weaknesses of Current System**

* Limited accessibility: Offline systems cannot be accessed from anywhere like online systems. They are limited to specific physical locations where the system is installed.
* Limited scalability: Offline systems may not be able to handle large volumes of data and transactions like online systems, and scaling them up may require hardware upgrades.
* Limited collaboration: Offline systems may not allow multiple users to collaborate on a project or share files and data in real-time.
* Limited updates: Offline systems require manual updates and maintenance, which can be time-consuming and costly.
* Limited data security: Offline systems may be more susceptible to data loss, theft, or damage, as they are not protected by online security measures such as firewalls, encryption, and backups.
* Limited customer engagement: Offline systems may not be able to offer personalized and interactive features like online systems, which can impact customer engagement and satisfaction.

# **4.3 Requirements of New System**

* User registration and login: Users should be able to create an account and log in to access their photo library, saved projects, and order history.
* Photo upload and storage: Users should be able to upload photos to their account, which will be stored securely on the server.
* Photo editing and customization: Users should be able to customize their photos using various tools such as cropping, resizing, adding filters, text, and borders.
* Project creation: Users should be able to create and save photo projects such as photo books, calendars, cards, and wall art.
* Security and privacy: The system should ensure the security and privacy of user data, including photos, personal information, and payment details.

# **4.4 System Feasibility**

# **4.4.1 Does the system contribute to the overall objectives of the organization?**

Yes, the system can potentially contribute to an organization's objectives in various ways. For example, it can help increase revenue and profitability by offering new products and services to customers. It can also help increase customer satisfaction and loyalty by providing a convenient and personalized experience. Additionally, it can help a company differentiate itself from competitors and build a strong brand image. Ultimately, whether a system contributes to the overall objectives of an organization would depend on how it is implemented and utilized within the context of the organization's broader strategy and goals.

# **4.4.2 Can the system be implemented using the current technology and within the given cost and schedule constraints?**

Yes, it is possible to implement the system using the MERN (MongoDB, Express.js, React, Node.js) technology stack. MERN is a popular technology stack for developing web applications, and it provides a robust and scalable foundation for building complex systems.

However, if the project is well-planned and executed efficiently, it is possible to complete it within the given cost and schedule constraints. It is important to consider factors such as project scope, requirements, risk assessment, and contingency planning to ensure a successful implementation.

# **4.4.3 Can the system be integrated with other systems which are already in place?**

No, the system cannot be integrated with other systems that are already in place. Integration can be achieved through the use of APIs and web services that enable communication between different systems. For example, E-print may need to integrate with a payment gateway or a third-party printing service. By using APIs, these services can be integrated seamlessly into the E-print platform. Integration can also help to streamline processes and improve efficiency by automating tasks and eliminating the need for manual data entry. However, integration must be included in future versions of E-print to ensure that it does not disrupt the existing systems or compromise data security.

# **4.5 Process in New System**

* Upload: Customers upload their digital photos to the E-print website or mobile app.
* Creation: Customers create their desired products by selecting their preferred size, design, layout, and other customization options.
* Review: Customers review their creations to ensure they are correct before submitting their order.
* Production: E-print produces the ordered products, which can include prints, photo books, cards, calendars, and other personalized items.

# **4.6 Features of New System**

* Photo Storage: E-print provides users with the ability to store their photos online, making it easy to access and share them from anywhere.
* Photo Printing: E-print offers a wide range of printing options, including photo books, cards, calendars, and other custom products..
* Security and Privacy: E-print takes the security and privacy of its users seriously, providing secure online storage and encryption for all data.

# **4.7 List Main Modules**

**Backend module:**

* User module: This module deals with user-related functionalities such as registration, login, profile management, and account settings.
* Cart module: This module manages the user's shopping cart, adding and removing products, and keeping track of the total cost.
* Category module: This module manages the different product categories available in the system, enabling users to browse and search for products by category.
* Product module: This module manages the products available in the system, including product details, pricing, and availability.
* Role module: This module defines the different user roles in the system, such as customer, vendor, and administrator, and assigns appropriate permissions and access levels.
* Vendor product module: This module manages the products listed by vendors, including product details, pricing, and availability.
* Vendor product image module: This module manages the images associated with vendor products, including upload, storage, and retrieval.
* Country module: This module manages the different countries available in the system, including country details and associated regions.
* City module: This module manages the different cities available in the system, including city details and associated regions.
* State module: This module manages the different states available in the system, including state details and associated regions.

**Frontend modules:**

**Individual modules:**

* Vendor module: This module manages vendor-related functionalities such as vendor registration, vendor login, vendor profile management, and vendor account settings.
* Admin module: This module deals with administrative tasks such as managing user accounts, managing vendor accounts, and managing product categories.
* Authentication module: This module manages the authentication and authorization process for users, including user login, password management, and access control.

**Common modules:**

* Search product module: This module enables users to search for products based on various criteria such as category, price range, and availability.
* Search categories module: This module enables users to search for product categories based on various criteria such as popularity, relevance, and availability.
* Payment module: This module manages the payment process for orders placed by users, including payment options, payment gateway integration, and transaction management.
* Update module: This module manages the process of updating user profiles, product details, and other information in the system.
* Delete module: This module manages the process of deleting user accounts, product listings, and other data from the system.

# **4.8 Selection of Hardware / Software / Algorithms / Methodology / Techniques / Approaches and Justification**

**Hardware Requirements:**

* Processor: Intel Core i5 or equivalent
* RAM: 8 GB or more
* Storage: At least 500 GB hard disk or equivalent SSD
* Display: Minimum 15.6-inch Full HD display

**Software Requirements:**

* Operating System: Windows 10 or macOS Big Sur or later versions
* Web Browser: Latest version of Google Chrome, Mozilla Firefox, or Safari
* Development Environment: Visual Studio Code or any other similar IDE

**Algorithms:**

* Image compression for efficient storage and faster upload/download times
* Search algorithms for efficient searching and filtering of products and categories

**Methodology:**

* Agile methodology for project management and development
* Continuous Integration/Continuous Deployment (CI/CD) for automated testing and deployment
* Test-driven development (TDD) for ensuring code quality and reducing bugs
* User-centred design methodology for designing a user-friendly interface

**Techniques:**

* Responsive web design for optimal user experience on different devices
* Scalable and modular code architecture for easier maintenance and future development
* API integration for seamless integration with third-party services

**Approaches:**

* Object-oriented programming (OOP) for modular and reusable code
* Model-View-Controller (MVC) architecture for separation of concerns and easier testing
* Microservices architecture for scalability and flexibility

**Justification:**

The above hardware, software, algorithms, methodology, techniques, and approaches are necessary for running the system smoothly and efficiently. The recommended hardware and software specifications ensure that the system is fast and responsive, and the algorithms ensure that the images are processed and displayed quickly and efficiently. The methodologies, techniques, and approaches ensure that the system is developed, tested, and deployed in a structured and efficient manner, resulting in a high-quality and user-friendly product.

# **Chapter - 5 System Design**

# **5.1 System Design & Methodology**

# **5.1.1 ER Diagram:**

An Entity-Relationship (ER) diagram is a graphical representation of entities and their relationships to each other. It is commonly used in software engineering to design and model databases, especially in the context of relational database management systems (RDBMS). ER diagrams are used to show the relationships between entities, which are represented as boxes, and the attributes of those entities, which are represented as ovals. The relationships between entities are represented by lines, which connect the boxes. The main components of an ER diagram are entities, attributes, relationships, and cardinality.

Entities represent the objects or concepts in the system being modelled. Attributes are the characteristics or properties of the entities. Relationships describe how entities are related to each other, such as "one-to-one," "one-to-many," or "many-to-many." Cardinality is the numerical relationship between entities, such as "one-to-one," "one-to-many," or "many-to-many."

ER diagrams are useful for modelling complex systems because they provide a visual representation of the relationships between entities, which can help to identify potential problems or areas of improvement. They are commonly used in the early stages of software development to design and plan a database schema, and they can be used throughout the software development lifecycle to communicate the design of the database to stakeholders.

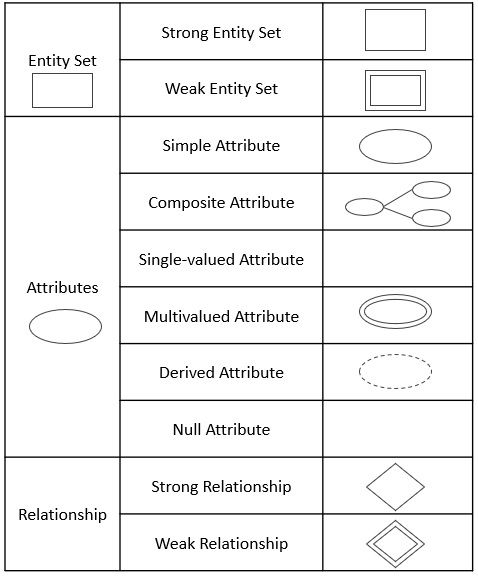


Table 5.1. Symbols and Components of ER Diagram

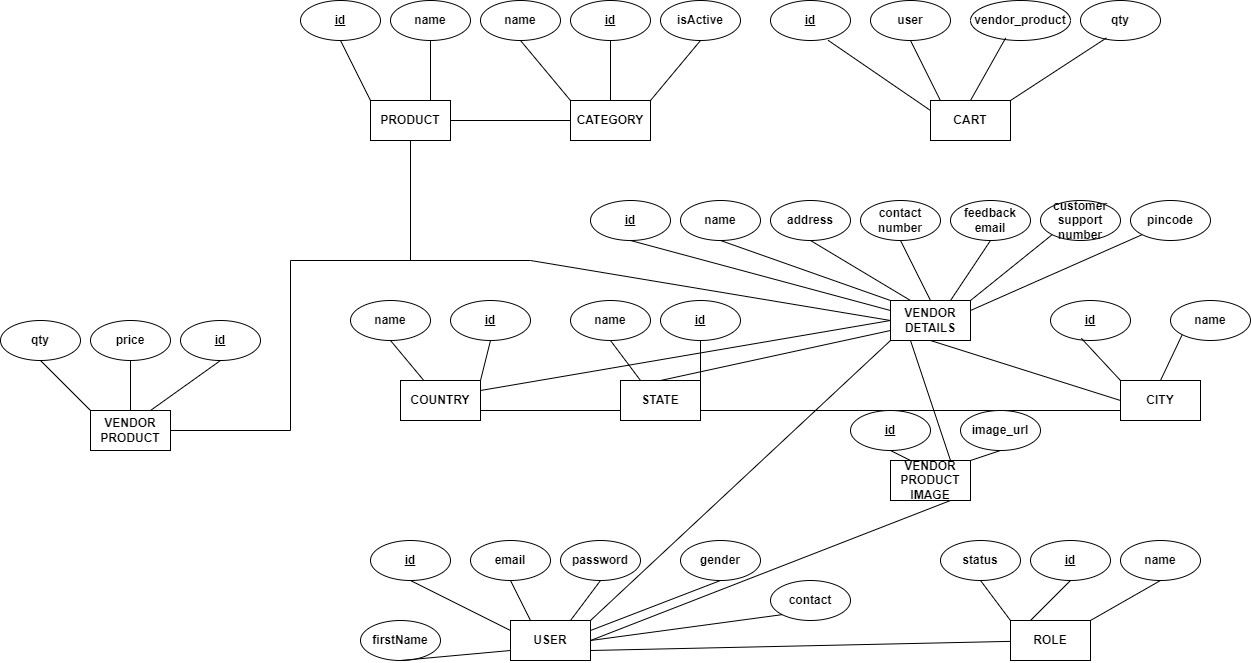


Fig 5.1. ER Diagram

# **5.1.2 Use Case Diagram:**

A use case diagram is a graphical representation of the interactions between users (actors) and a system in the context of a specific task or process. It helps to visualize the system’s functionality from a user’s perspective and identifies the possible interactions between the user and the system.

In a use case diagram, the actors represent the users or external systems that interact with the system. The use cases represent the tasks or processes that the system performs in response to the user or actor interactions. The relationships between the actors and the use cases are depicted using arrows.

Use case diagrams are often used in the early stages of software development to define the system requirements and to identify the different use cases that the system will need to support. They can also be used to communicate the system's functionality to stakeholders who may not be familiar with the technical details of the system.

Overall, a use case diagram is a powerful tool for modelling a system's functionality and interactions with its users, helping to ensure that the system meets the needs of its users and stakeholders.

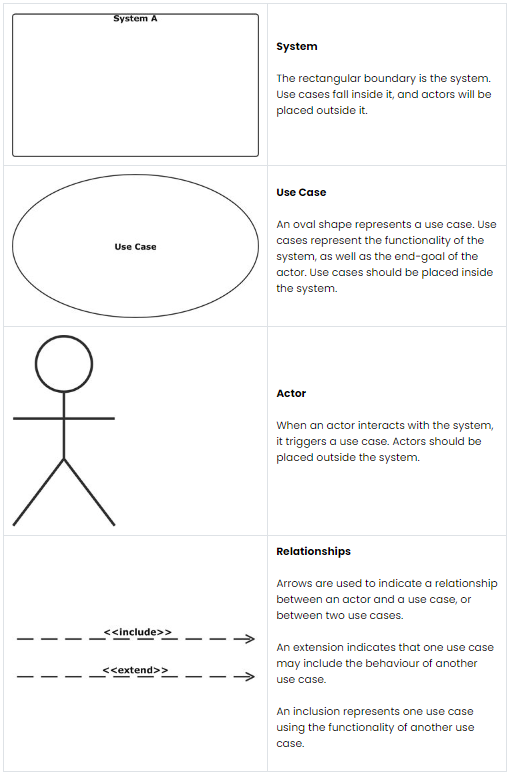
****

Table 5.1. Symbols and components of Use Case Diagram

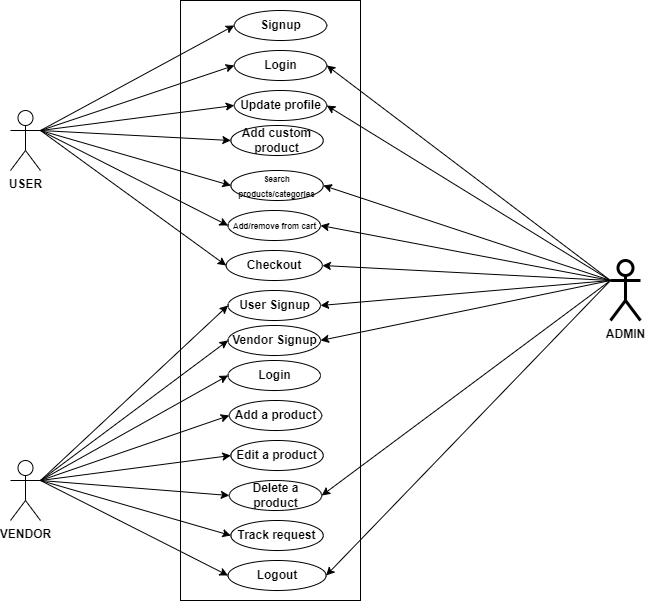


Fig 5.1. Use Case Diagram

# **5.1.3 Activity Diagram:**

An activity diagram is a type of behavioural diagram in UML (Unified Modelling Language) that depicts the flow of activities and actions within a system, process, or workflow. It is used to model the dynamic behaviour of a system and visualize the sequence of activities and the relationships between them.

Here are some key components and concepts used in activity diagrams:

* Activity: An activity represents a specific task or operation within a system. It can represent a single action or a collection of actions that need to be performed.
* Action: An action represents a specific behaviour or operation that occurs within an activity. It can be a simple action, such as printing a document, or a complex action involving multiple steps.
* Control Flow: Control flow arrows depict the flow of control from one activity or action to another. They show the sequence in which activities are executed.
* Decision Node: A decision node represents a point where the flow of control diverges based on a condition or decision. It typically has multiple outgoing control flows, each representing a different possible path based on the condition.
* Merge Node: A merge node represents a point where multiple control flows converge back into a single flow. It is used to combine different paths or branches back into a unified flow of control.
* Fork Node: A fork node represents a point where the flow of control splits into multiple concurrent flows. It is used to indicate parallel execution or concurrency of activities.
* Join Node: A join node represents a point where multiple concurrent control flows converge back into a single flow. It is used to synchronize parallel activities and wait for all concurrent flows to complete before continuing.
* Initial Node and Final Node: An initial node represents the starting point of the activity diagram, while a final node represents the end or termination point of the diagram.

Activity diagrams provide a visual representation of the workflow or process, making it easier to understand the sequence of activities, decision points, and parallel flows within a system. They are commonly used in software development, business process modelling, and system analysis to describe the behaviour and interactions of a system or process.

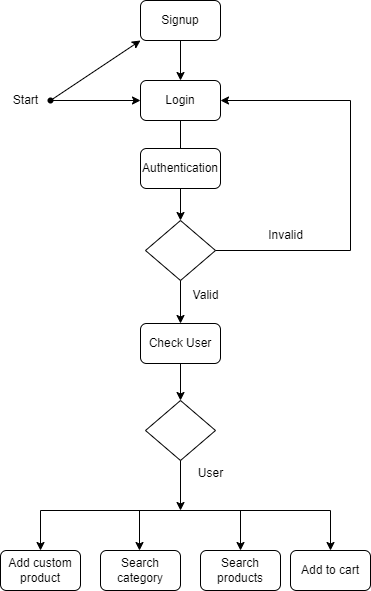


Fig 5.1. User Activity Diagram

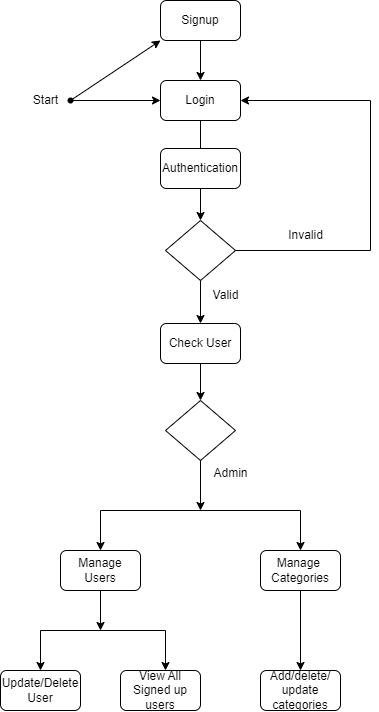


Fig 5.1. Admin Activity Diagram

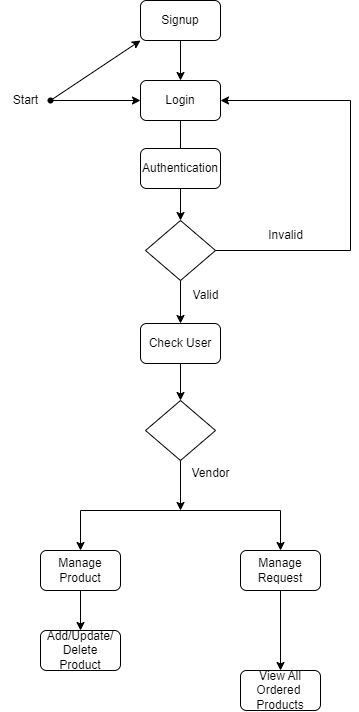


Fig 5.1. Vendor Activity Diagram

# **5.1.4 Class Diagram:**

A class diagram is a type of UML (Unified Modelling Language) diagram that represents the structure and relationships of classes in an object-oriented system. It provides a visual representation of the classes, their attributes, methods, and the associations between them.

Here are the key components and concepts used in a class diagram:

* Class: A class is a blueprint or template for creating objects. It represents a concept, entity, or a group of similar objects in the system. Each class typically has a name, attributes (data variables), and methods (functions or operations).
* Attributes: Attributes are the properties or characteristics of a class. They represent the data associated with the class. For example, a "Person" class may have attributes like "name," "age," and "address."
* Methods: Methods represent the behaviour or operations that can be performed by the objects of a class. They define the actions that an object can perform or the computations it can carry out. For example, a "Person" class may have methods like "getAge()" or "changeAddress()".
* Relationships: Relationships illustrate how classes are connected or associated with each other. There are several types of relationships, including:
  1. Association: It represents a general relationship between two classes. For example, a "Teacher" class may be associated with a "Student" class.
  2. Aggregation: It represents a "part-of" relationship, where one class is composed of or contains other classes. It is denoted by a diamond-shaped arrow. For example, a "Car" class may have an aggregation relationship with a "Wheel" class.
  3. Composition: It is a stronger form of aggregation where one class is composed of other classes and has exclusive ownership over them. It is denoted by a filled diamond-shaped arrow. For example, a "Car" class may have a composition relationship with an "Engine" class.
  4. Inheritance: It represents an "is-a" relationship, where one class inherits the properties and behaviors of another class. It is denoted by an arrow with an open triangle. For example, a "Student" class may inherit from a "Person" class.
  5. Dependency: It represents a "uses" relationship, where one class depends on another class for some functionality or information. It is denoted by a dashed arrow. For example, a "Car" class may have a dependency on a "Fuel" class.

Class diagrams are widely used in software development to design and visualize the structure of a system's classes and their relationships. They help in understanding the overall architecture, organizing classes and their interactions, and communicating the design to developers and stakeholders.

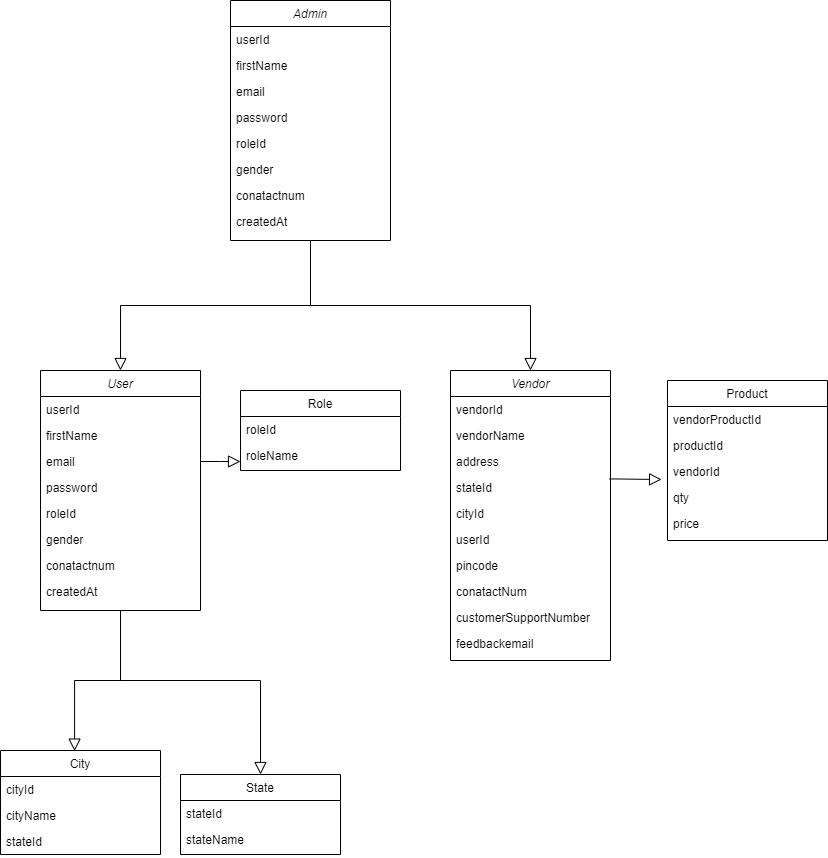


Fig 5.1. Class Diagram

# **5.1.5 Data Dictionary Diagram:**

A data dictionary diagram, also known as a data dictionary or data schema diagram, is a visual representation of the data elements, their relationships, and their attributes within a system or database. It provides a structured overview of the data entities and their characteristics, serving as a reference for data modelling, database design, and system documentation.

Here are the key components and concepts used in a data dictionary diagram:

* Data Entities: Data entities represent the main objects or concepts in the system. They can be entities such as customers, products, employees, or any other relevant objects. Each data entity is typically depicted as a rectangle or a table-like structure in the diagram.
* Attributes: Attributes are the characteristics or properties of a data entity. They describe the specific details or data elements associated with each entity. For example, a "customer" entity may have attributes such as "customer ID," "name," "address," and "phone number." Attributes are usually listed within the data entity rectangle or table.
* Relationships: Relationships illustrate how data entities are connected or associated with each other. They represent the logical connections or dependencies between entities. Relationships can be one-to-one, one-to-many, or many-to-many. They are often depicted as lines connecting the related entities, with labels indicating the nature of the relationship, such as "has," "belongs to," or "uses."
* Primary Keys: A primary key is a unique identifier for each instance of a data entity. It uniquely identifies a record or row within a database table. In a data dictionary diagram, primary keys are usually underlined or highlighted to indicate their significance and uniqueness.
* Data Types: Data types specify the nature or format of the data attributes. They define the kind of values that can be stored in each attribute. Common data types include text, numbers, dates, boolean values, and more. Data types are often mentioned alongside the attributes in the data dictionary diagram.

Data dictionary diagrams provide a concise and structured representation of the data elements, their relationships, and their attributes within a system or database. They serve as a valuable tool for data modelling, database design, and system analysis. By documenting the data entities and their characteristics, data dictionary diagrams aid in understanding the data structure, ensuring data integrity, and facilitating effective communication among stakeholders involved in the development and maintenance of the system.

Table 5.1. Role

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| roleId | integer | 9 | primary key |
| roleName | varchar | 20 | not null, unique |

Table 5.1. User

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| userId | integer | 9 | primary key |
| firstName | varchar | 30 | not null |
| email | varchar | 30 | not null |
| password | varchar | 30 | not null |
| roleId | integer | 9 | foreign key: role |
| gender | char | 1 | m or f |
| contactnum | varchar | 15 |  |
| createdAt | date |  |  |

Table 5.1. Category

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| categoryId | integer | 9 | primary key |
| categoryName | varcahar | 30 | not null |
| isActive | integer | 1 |  |

Table 5.1. State

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| stateId | integer | 9 | primary key |
| stateName | varchar | 30 | not null , unique |

Table 5.1. City

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| cityId | integer | 9 | primary key |
| cityName | varchar | 30 | not null |
| stateId | integer | 9 | foreign key |

Table 5.1. vendor\_detail

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| vendorId | integer | 9 | primary key |
| vendorName | varchar | 30 |  |
| address | varchar | 250 |  |
| stateId | integer | 9 | foreign key |
| cityId | integer | 9 | foreign key |
| userId | integer | 9 | foreign key |
| pincode | integer | 9 | not null |
| contactNum | varchar | 12 |  |
| customerSupportNumber | varchar | 12 |  |
| feedbackemail | varchar | 30 |  |

Table 5.1. Product

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| productId | integer | 9 | primary key |
| productName | varchar | 30 | not null |
| categoryId | integer | 9 | foreign key |

Table 5.1. vendor\_product

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| vendorProductId | integer | 9 | primary key |
| productId | integer | 9 | foreig key |
| vendorId | integer | 9 | foreig key |
| qty | integer | 9 | not null |
| price | float |  | not null |

Table 5.1. Cart

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **DataType** | **Size** | **Constraints** |
| cartId | integer | 9 | primary key |
| userId | integer | 9 | foregin key |
| vendorProductId | integer | 9 | foregin key |
| qty | integer | 9 |  |

Table 5.1. vendor\_product\_images

|  |  |  |  |
| --- | --- | --- | --- |
| **Table: vendor\_product\_images** |  |  |  |
| **field** | **DataType** | **Size** | **Constraints** |
| vendor\_product\_image\_id | integer | 9 | pk |
| vendro\_productId | integer | 9 | fk |
| vendor\_id | integer | 9 | fk |
| imageurl | varchar | 500 | not null |

# **5.1.6 Sequence Diagram:**

A sequence diagram is a type of interaction diagram in UML (Unified Modeling Language) that depicts the interactions between objects or components within a system over a specific time period. It illustrates the sequence of messages exchanged between the objects and the order in which they occur.

Here are the key components and concepts used in a sequence diagram:

* Objects or Lifelines: Objects represent the instances of classes or components participating in the interaction. Each object is depicted as a vertical line called a lifeline. Lifelines show the lifespan of the object during the sequence of interactions.
* Messages: Messages represent the communication or interaction between objects. They are depicted as horizontal arrows with a label indicating the type of message being sent. Messages can be synchronous, asynchronous, or self-referential. Synchronous messages are depicted as solid arrows and indicate a direct call that waits for a response. Asynchronous messages are depicted as dashed arrows and indicate a call that doesn't wait for an immediate response. Self-referential messages occur when an object sends a message to itself.
* Activation Boxes: Activation boxes, also known as activation bars or execution occurrences, represent the period of time during which an object is actively executing a method or processing a message. They are depicted as boxes or rectangles along the lifeline of an object and show the duration of the method execution.
* Lifeline Notations: Lifeline notations provide additional information about the objects or lifelines in the sequence diagram. They can include the name of the object, the class or component it represents, and other relevant details.
* Return Messages: Return messages indicate the response or return value from a called message. They are depicted as arrows that return from the target object to the sender, often with a label indicating the value or result being returned.

Sequence diagrams help visualize the dynamic behaviour of a system by showing how objects interact with each other through messages. They are particularly useful for understanding the order of interactions, the flow of control, and the collaboration between objects in a specific scenario or use case. Sequence diagrams are commonly used during the design and implementation phases of software development to ensure that the interactions among objects are properly defined and understood.

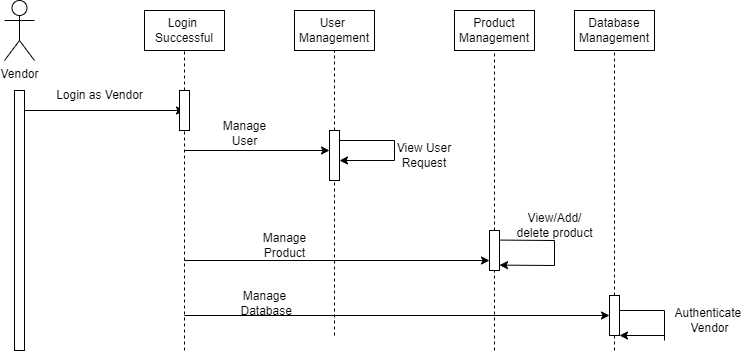


Fig 5.1. Vendor Sequence Diagram

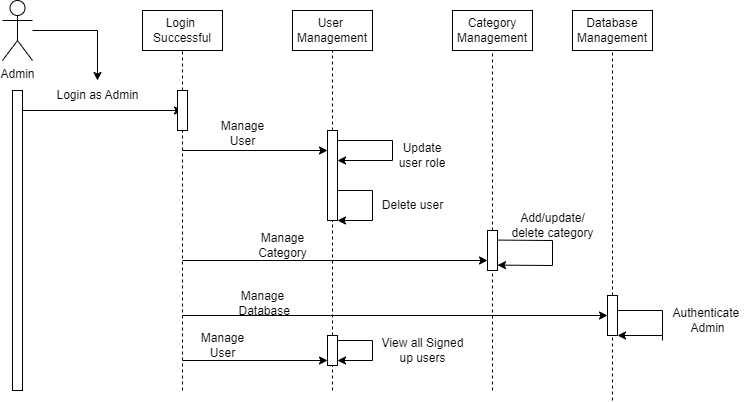


Fig 5.1. Admin Sequence Diagram

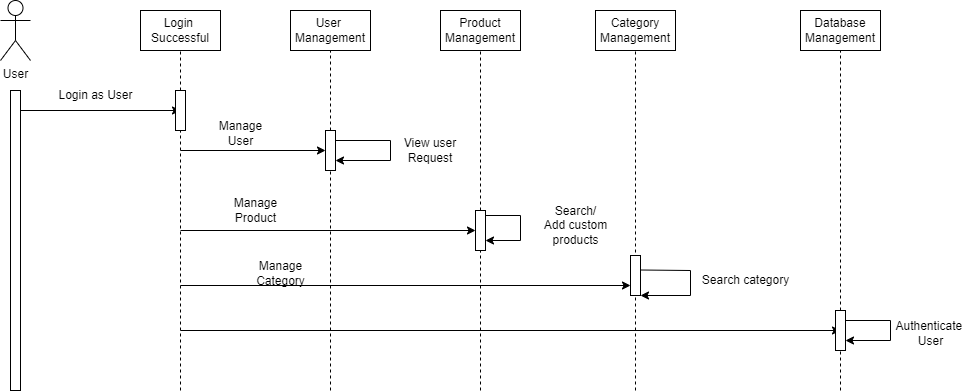


Fig 5.1. User Sequence Diagram

# **Chapter - 6 Implementation**

# **6.1 Environment**

**MongoDB:**

MongoDB is a popular open-source, document-oriented NoSQL database management system. Instead of storing data in tables and rows like a traditional relational database, MongoDB stores data as documents in flexible, JSON-like structures, called BSON (Binary JSON) documents. MongoDB was designed to provide high availability, scalability, and performance while also making it easy to develop modern applications.

**Here are some key features and concepts of MongoDB:**

* Document-Oriented: MongoDB stores data in documents, which are self-contained data structures consisting of fields in key-value pairs. These documents are similar to JSON objects, making it easier for developers to work with data.
* NoSQL: MongoDB is a NoSQL database, which means it doesn't use a traditional relational database model. Instead, it uses dynamic schemas, which allows for more flexible and agile development.
* Indexing: MongoDB provides multiple types of indexing, including single field, compound, multi-key, text, and geospatial indexes. This makes it easy to perform complex queries and improve query performance.
* Replication and Sharding: MongoDB provides automatic replication and sharding, which helps ensure high availability, scalability, and fault tolerance.
* Aggregation: MongoDB provides powerful aggregation and data processing capabilities, including grouping, filtering, sorting, and more.
* GridFS: MongoDB provides a file storage system called GridFS, which allows users to store and retrieve large files, such as images, audio, and video.

To run MongoDB, you will need to have compatible hardware and software. MongoDB supports a wide range of operating systems, including Windows, Linux, macOS, and more. It also requires a minimum amount of RAM and storage space, which will vary depending on your usage requirements.

In terms of methodology and techniques, MongoDB is designed to be developer-friendly and easy to use. It provides a rich set of APIs and drivers for many programming languages, including Java, Python, Node.js, and more. MongoDB also provides extensive documentation and community support to help developers get started quickly and easily.

Overall, MongoDB is a powerful, flexible, and easy-to-use NoSQL database that provides high availability, scalability, and performance for modern applications.

**Express JS:**

Express.js is a popular Node.js web application framework that provides a robust set of features for building web and mobile applications. It provides a fast and minimalist web framework that allows developers to build scalable and secure web applications easily.

Some of the key features of Express.js include:

* Routing: Express provides a flexible routing system that allows developers to define routes and associated HTTP methods. This enables easy handling of requests and responses.
* Middleware: Express has a middleware architecture that allows developers to plug in additional functionality to their applications, such as authentication, logging, error handling, and more.
* Template engines: Express supports a variety of template engines like EJS, Pug, and Handlebars, which help in rendering dynamic content on the server side.
* Static file serving: Express makes it easy to serve static files like images, videos, and documents from a directory on the server.
* Error handling: Express has built-in error handling middleware that provides a centralized way to handle errors and exceptions that occur during the request/response cycle.
* Integration with other libraries: Express can be easily integrated with other popular Node.js libraries and frameworks like Passport.js for authentication, Socket.io for real-time communication, and more.

Express.js is widely used in production environments for building scalable and performant web applications. Its lightweight nature and flexibility make it a popular choice among developers.

**NodeJS:**

Node.js is an open-source, cross-platform runtime environment that allows developers to build server-side applications using JavaScript. It uses an event-driven, non-blocking I/O model that makes it lightweight, efficient, and ideal for building scalable network applications.

Node.js provides an extensive library of modules that can be easily incorporated into applications. These modules provide support for a wide range of functionality, including file system I/O, networking, databases, cryptography, and much more.

One of the most significant features of Node.js is its ability to handle asynchronous programming, which is essential for building scalable, high-performance applications. Node.js uses a single-threaded event loop to handle I/O operations, making it possible to handle large numbers of simultaneous connections without blocking.

Another key benefit of Node.js is its ability to easily connect to and interact with databases, such as MongoDB, MySQL, and PostgreSQL. Node.js has built-in support for many popular database systems, and there are numerous third-party libraries available for connecting to other databases.

Node.js also has a thriving ecosystem of tools, libraries, and frameworks that make it easier for developers to build applications. Some of the most popular frameworks include Express.js, Koa.js, and Hapi.js.

In summary, Node.js provides a powerful and efficient platform for building server-side applications using JavaScript. Its event-driven, non-blocking I/O model, extensive library of modules, and ability to easily connect to databases make it an ideal choice for building high-performance, scalable applications

.

**ReactJS:**

ReactJS is an open-source, front-end JavaScript library for building user interfaces or UI components. It was created by Facebook and is now maintained by Facebook and a community of developers. React allows developers to create reusable UI components and build complex UIs using a declarative programming style.

One of the key features of React is its use of a virtual DOM, which is a lightweight representation of the actual DOM. When a component's state changes, React updates the virtual DOM and then efficiently updates the actual DOM with only the necessary changes. This helps to reduce the amount of work required to update the DOM and makes React applications performant and responsive.

React also has a powerful component model, which allows developers to create complex UIs by composing smaller, reusable components. Each component can have its own state, which can be modified in response to user input or other events. React also supports a unidirectional data flow model, where data flows down from parent components to child components via props, and events flow up from child components to parent components via call-backs.

Another key feature of React is its use of JSX, a syntax extension that allows developers to write HTML-like code in JavaScript. JSX makes it easy to create complex UIs and helps to reduce the amount of boilerplate code required.

React has a large and active community of developers, which has led to the creation of many third-party libraries and tools for building React applications. These include popular libraries like Redux, React Router, and React Native, which allow developers to build complex web and mobile applications using React.

In summary, ReactJS is a popular front-end JavaScript library for building reusable UI components and complex user interfaces using a declarative programming style and a powerful component model. Its use of a virtual DOM, JSX syntax, and unidirectional data flow make it a performant and efficient library for building modern web and mobile applications.

# **6.2 Modules Specification(s)**

* Photo printing: Users can upload their digital photos and order prints in various sizes and formats, such as standard prints, canvas prints, photo books, calendars, and more.
* Personalized gifts: Users can create personalized gifts using their own photos, such as mugs, keychains, phone cases, and more.
* Stationery and cards: Users can create customized stationery and greeting cards using their own photos.
* Photo storage and organization: Users can store their digital photos online and organize them into albums.
* Special occasions: Users can create customized products for special occasions such as weddings, birthdays, and holidays.

**6.2.1 Process:**

There are 4 different pages in our website and also one hidden additional page that is activated according to user login.

**1. Home Page Code:**

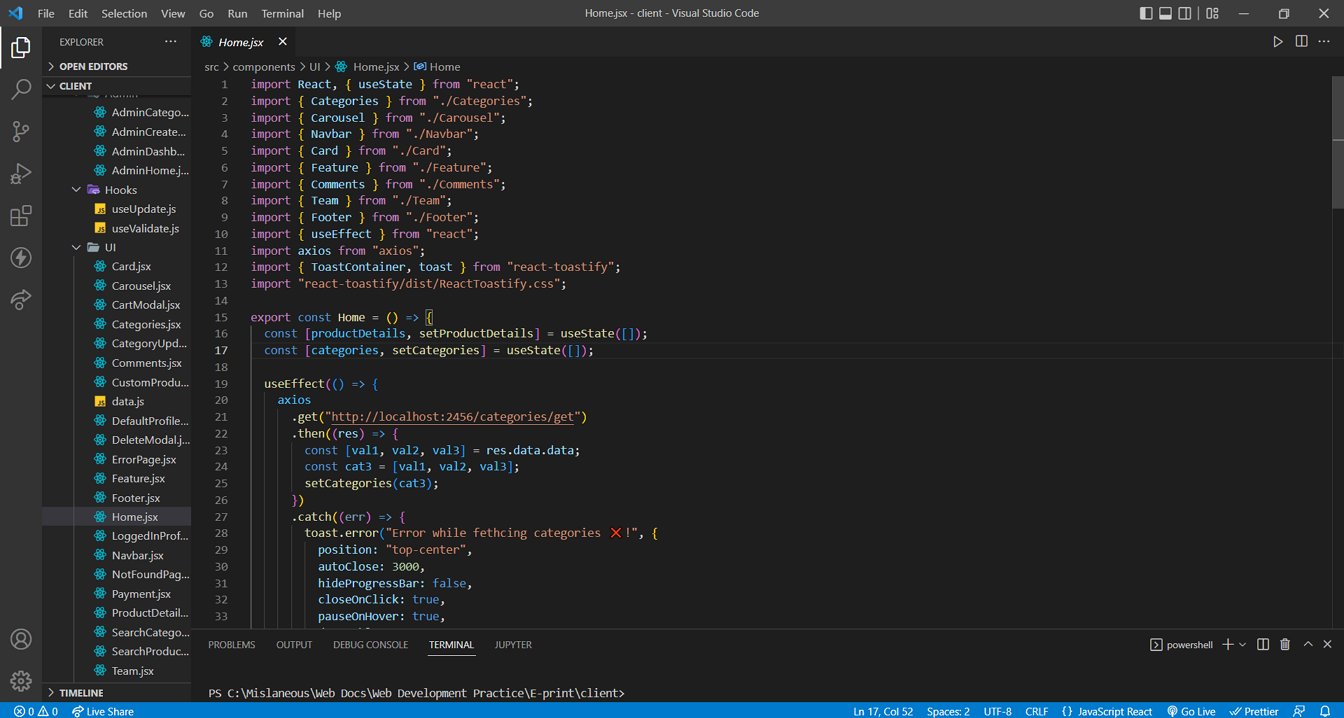


Fig 6.2. Home Page Code

**2. Admin Dashboard Code:**



Fig 6.2. Admin Dashboard Code

**3. Vendor Dashboard Code:**

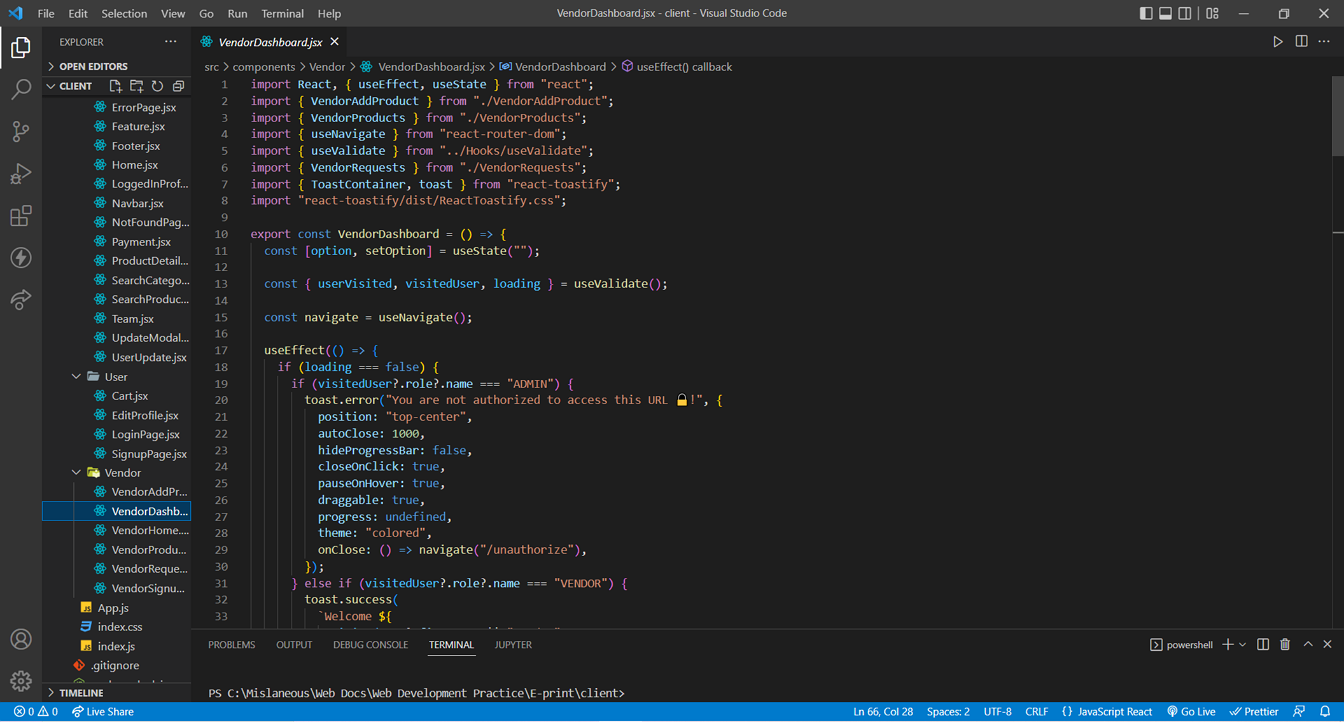


Fig 6.2. Vendor Dashboard Code

**4. Search Products/Category Code:**

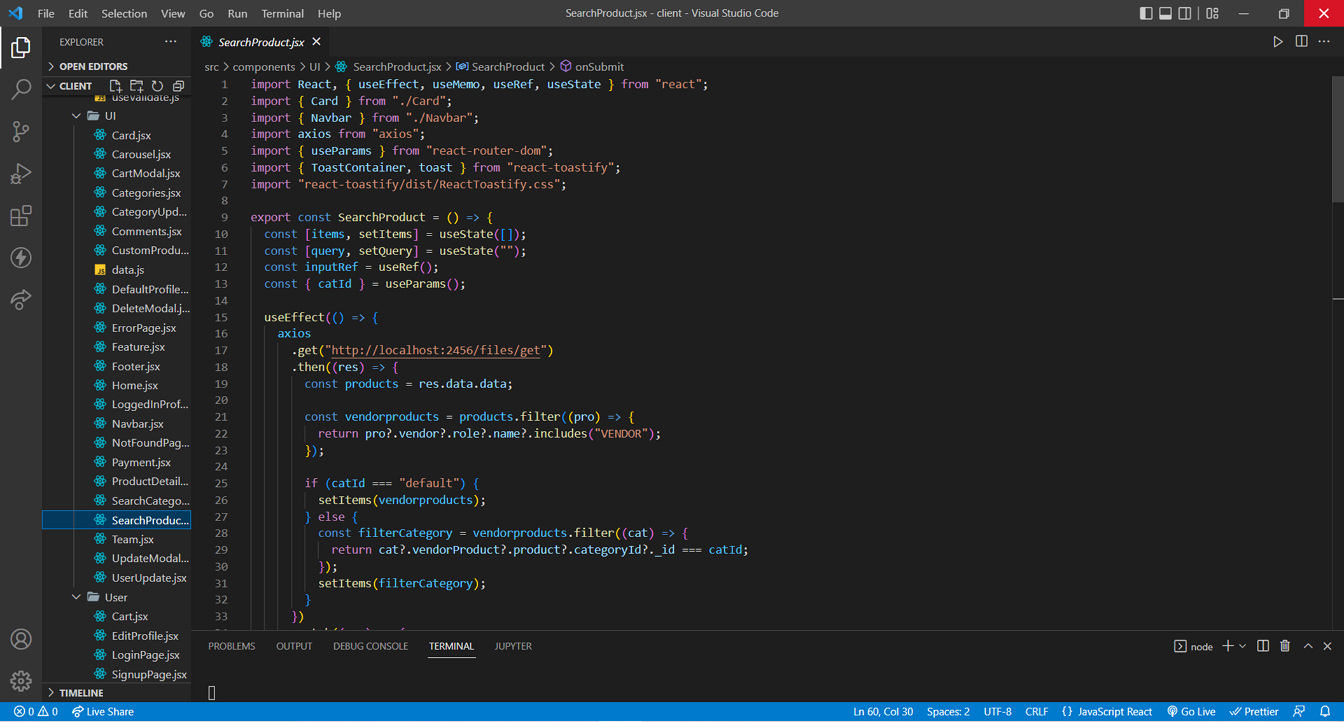


Fig 6.2. Search Product/Category Code

# **6.3 Project Screenshot and Sneak peek**

The project is laid out according to the modules that are discussed in above chapters, Although the modules are re-explained according to the working features.

**1. HOME:**

The home page contains navbar, slider images, top products, top categories, comments, team members and footer.

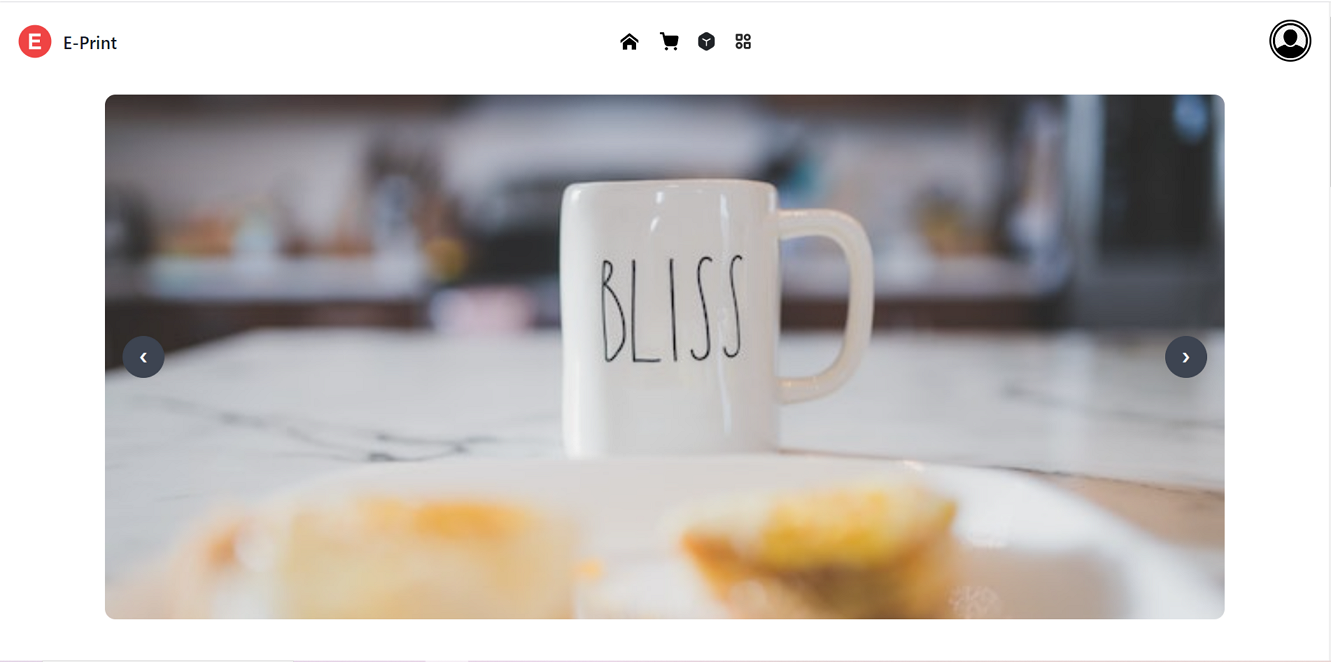


Fig 6.3. Home Page

**2. CART:**

Cart displays the user products that are added to the cart by the user.

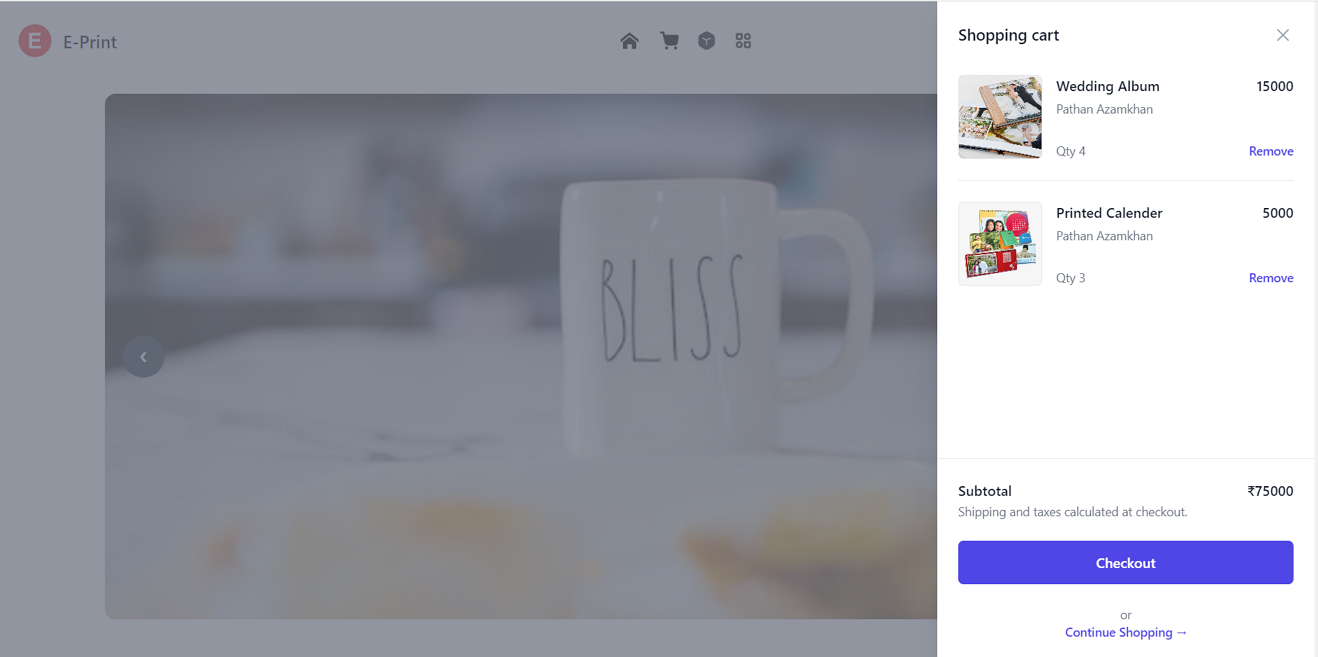


Fig 6.3. Cart Page

**3. Search Products:**

Entering the query fires the search algorithm to find the entered product.

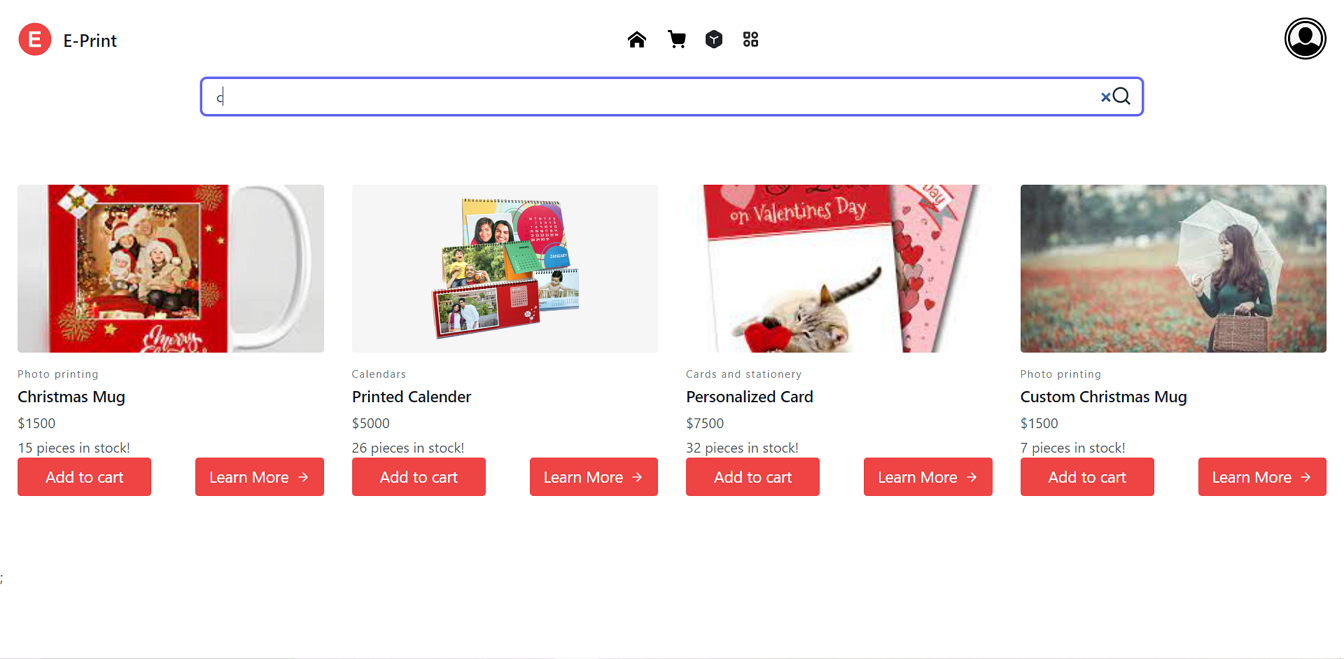


Fig 6.3. Search Products Page

**4. Search Category:**

Entering the query fires the search algorithm to find the entered category.

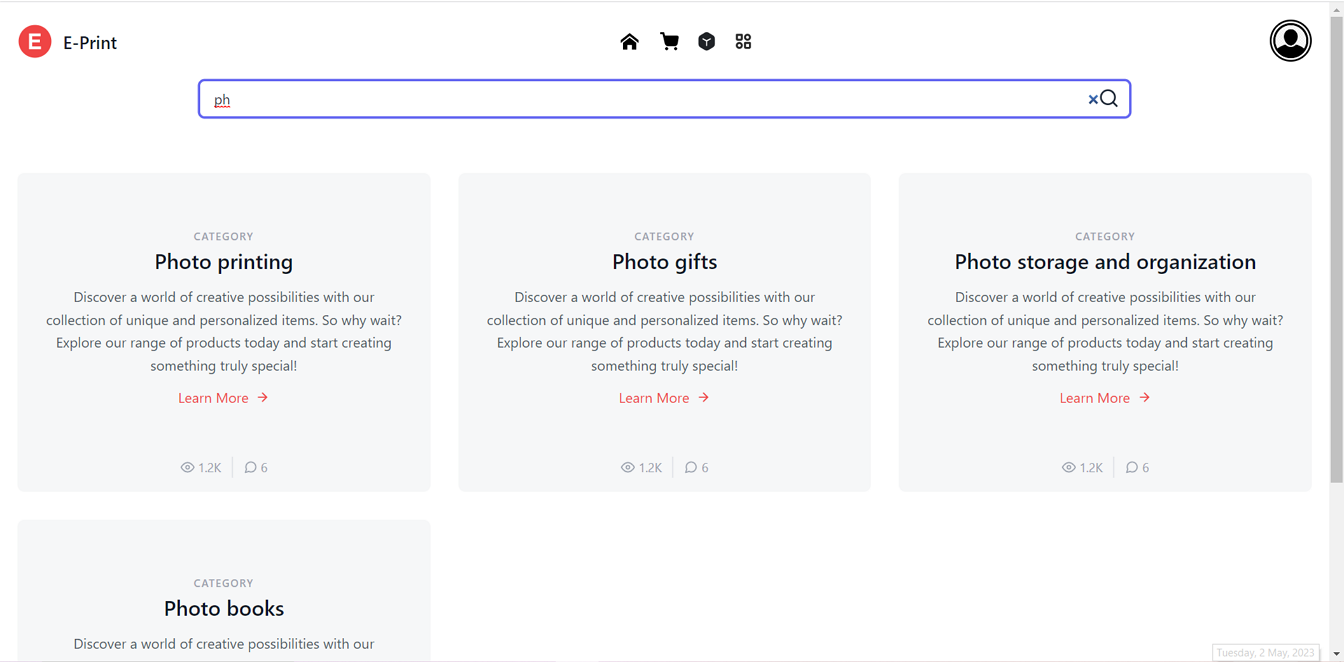


Fig 6.3. Search Category Page

**5. Profile:**

User profile on the top right corner for personal details forgery and logout options.



Fig 6.3. Profile Page

**6. Vendor Dashboard:**

A Dashboard made for vendor to create products and list them to the products page

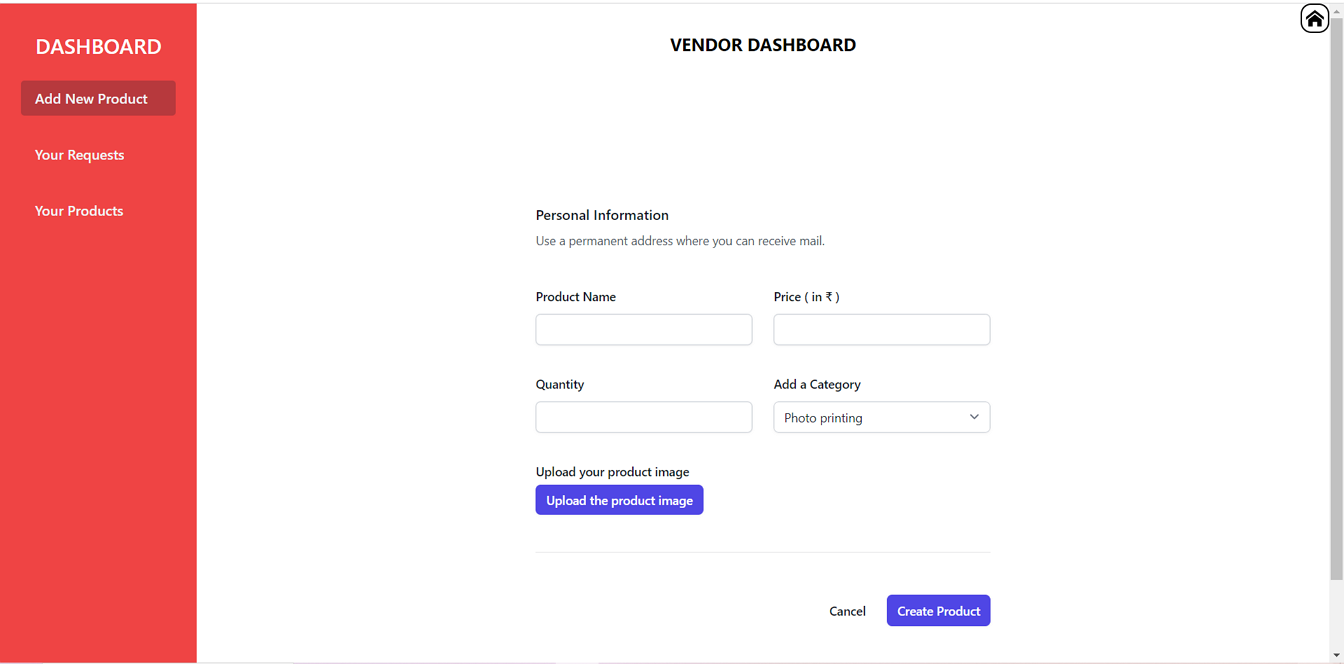


Fig 6.3. Vendor Dashboard Page

**7. Admin Dashboard:**

A dashboard created for admins to monitor user logins and category creation.

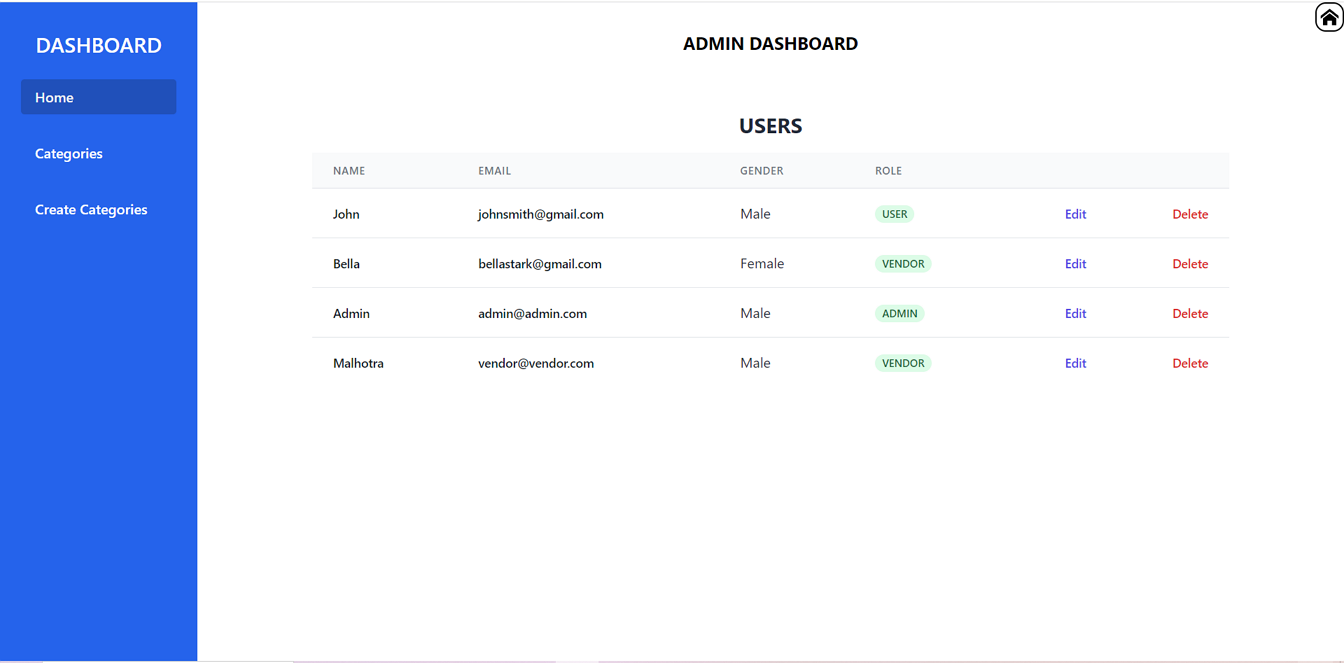


Fig 6.3. Admin Dashboard Page

# **Chapter - 7 Testing**

# **7.1 Testing Plan / Strategy**

* Functionality Testing: Tested all the features and functionalities of the website such as login, sign up, search, shopping cart, product listing and product details.
* Compatibility Testing: Tested the website's compatibility with different browsers, operating systems, and devices to ensure it works smoothly across all platforms.
* Performance Testing: Tested the website's performance by checking its load time, speed, and response time to ensure that it's fast and responsive.
* Usability Testing: Tested the website's usability by evaluating its user interface, navigation, and overall user experience to ensure that it's easy to use and intuitive.
* Security Testing: Tested the website's security by checking for vulnerabilities such as NoSQL injections, cross-site scripting, and other security threats to ensure that the website is secure and data is protected.
* Cross-functional Testing: Tested the integration between different modules of the website, such as the user module, cart module, vendor module, and admin module, to ensure that they work together smoothly.
* Regression Testing: Tested the website after any changes or updates have been made to ensure that no new issues or bugs have been introduced.
* User Acceptance Testing: Conducted user acceptance testing to gather feedback from users and ensure that the website meets their expectations.

# **7.2 Test Results and Analysis**

# **7.2.1 Test Cases (test ID, test condition, expected output, actual output, remark)**

Table 7.2. Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case ID | Test case Description | Input Data | Expected output | Actual output | Status |
| 1 | User Signup | First Name, Email, Password, Gender | Should save the data in the database and redirect user to login page | Saved the data in database and redirected the user to login page | Pass |
| 2 | User Login | Email, Password | Should validate whether the user logging in does exist and if it does what’s his/her roles. | Validates whether the user logging in does exist and his/her roles. | Pass |
| 3 | Vendor Signup | FirstName, Contact Number, Customer Support Number, Country, City, State, Postal code, Feedback Email Address, Street Address | Should save the data in the database and redirect the vendor, to vendor dashboard | Saved the data in the database and redirected the vendor, to vendor dashboard | Pass |
| 4 | Add product to user’s cart | Button click | Should add the product to user’s cart | Adds the product to the user’s cart | Pass |
| 5 | Remove product from user’s cart | Button click | Should remove the product from user’s cart | Removes the product from user’s cart | Pass |
| 6 | Search products | Search bar | Should list the product according to the product name | Lists the products according to the input | Pass |
| 7 | Search categories | Query | Should list the categories according to category name | Lists the categories according to the input | Pass |
| 8 | Vendor create product | Product name, price, quantity, category and image | Should create a new product that should be added to database and then displayed to the user | Creates the product and displays it to the user in search products | Pass |
| 9 | Admin create category | Category name, category state | Should create the category and lists it in the search category page | Creates a category and lists it to the search category page | Pass |

# **Chapter - 8 Conclusion and Discussion**

# **8.1 Project Viabilities**

E-print can solve several problems associated with online photo printing and storage. The system provides a platform for users to upload, store, and print their photos in various formats and sizes. With advanced features like customization, the system allows users to create personalized photo books, calendars, and gifts for special occasions. The integration of various modules like the user, cart, vendor, and admin modules makes the system efficient, easy to use, and provides a smooth user experience.

The system is built on robust technologies like the MERN stack, which ensures scalability, reliability, and security. The use of MongoDB as the database and Express.js and Node.js for server-side programming, provides a fast and responsive system. The client-side is developed using React.js, which ensures the system is highly interactive and user-friendly.

Overall, the system provides a comprehensive solution for users looking to store, share, and print their photos. The system's advanced features, scalability, reliability, and security make it capable of handling large volumes of users and data. The development of such a system is an excellent investment for organizations looking to venture into the online photo storage and printing business.

# **8.2 Problem Encountered and Possible Solutions**

Performance issues: The system may be slow or unresponsive, especially when dealing with large amounts of data. This addressed by optimizing the code, using caching mechanisms, or scaling up the hardware resources.

Security vulnerabilities: The system may be vulnerable to hacking, data breaches, or other security threats. This mitigated by implementing strong authentication and authorization mechanisms, encrypting sensitive data, and performing regular security audits.

Integration issues: The system may have difficulty integrating with other existing systems or APIs. This resolved by carefully designing the integration points, using standardized protocols, and conducting thorough testing before deployment.

User adoption: The system may be difficult for users to learn and use, leading to low adoption rates. This addressed by conducting user research, designing a user-friendly interface, and providing adequate training and support.

# **8.3 Summary of Internship**

E-print offers a range of photo products, including prints, photo books, cards, and personalized gifts. Users can upload their photos to the website, choose the product they want, and customize it with their own designs. The finished products are then printed and shipped to the user's address.

It also offers free, unlimited photo storage for its users, making it easy to store and organize all of your digital photos in one place. Users can also add their photos of friends and family by adding the images via add custom product feature.

My learnings from the Internship:

I learnt many new and existing languages like HTML, CSS, JAVASCRIPT, NODEJS, EXPRESSJS, REACTJS, MONGODB.

It made me capable of building any full stack project from scratch. Also made me aware of different API and React libraries out their which can make development easy.

It also made me aware of the best practices which are used in industry and the staff of the company was so helpful and also excited to teach new comers the work they do it was a enjoyable and full of learning experience.

**Conclusion:**

In conclusion, E-print provides a convenient and user-friendly platform for customers to store and print their photos. The system offers a range of features such as personalized gifts and a variety of printing options. The development of such a system requires a comprehensive approach, including the selection of appropriate technologies, designing and implementing different modules, testing, and integrating with other systems. While E-print can significantly improve the photo management experience for users, it also has its limitations, such as the need for an internet connection and potential security risks. Future enhancements for the system could include the integration of artificial intelligence and machine learning algorithms to improve photo organization and searching, as well as expanding the product offerings to include other types of customizable items.

# **8.4 Limitation and Future Enhancement**

**Limitation of E-print:**

* Internet connectivity: As E-print is an online platform, it requires a stable internet connection to operate. Users who have poor or no internet connectivity will not be able to use the service.
* Limited physical products: Although E-print offers a wide range of products, there are still some limitations to the physical products that can be offered. Some users may prefer to have a physical product that is not offered by E-print.
* Security risks: As with any online service, there is always the risk of security breaches or unauthorized access to user data. E-print takes steps to protect user data, but there is always a risk of a breach occurring.
* Cost: While E-print is free to use, users will need to pay for the products they order. Depending on the quantity and type of products ordered, the cost can add up quickly.
* Reliance on third-party vendors: E-print relies on third-party vendors for printing and shipping services. This can sometimes result in delays or issues with the quality of the final product.

**Future Enhancement for E-print:**

* Integration with social media platforms: The ability to connect with popular social media platforms, such as Facebook and Instagram, to import photos and create personalized products directly from the user's social media account.
* Augmented Reality (AR) technology: Incorporating AR technology to allow users to visualize their personalized products in real-time and in a realistic setting, giving them a better idea of what the final product will look like.
* Enhanced customization options: Providing users with more advanced customization options, such as the ability to change fonts, colours, and layouts, as well as offering more options for personalization beyond photos, such as personalized text and graphics.
* Improved mobile application: Developing a more user-friendly and feature-rich mobile application, allowing users to create and order products from their mobile devices more easily.
* Expanded product offerings: Expanding the range of products offered beyond traditional photo prints and personalized gifts, such as personalized apparel, home decor, and more.
* Payment gateway: Integrating a payment gateway for users to make the system flow smooth with online transactions.
* Order tracking system: Building an order tracing system and integrating it with E-print to give user live status of his/her ordered product.

# **REFRENCES**

* Shutterfly: Photo Books, Cards, Prints, Wall Art, Gifts, Wedding. <https://www.shutterfly.com/>
* Tailwind CSS Components - Tailwind UI. (n.d.). Tailwind CSS Components - Tailwind UI. <https://www.tailwindui.com/>
* SVG Repo - Free SVG Vectors and Icons. (n.d.). SVG Repo. <https://www.svgrepo.com/>
* Tailblocks — Ready-to-use Tailwind CSS blocks. (n.d.). Tailblocks — Ready-to-use Tailwind CSS Blocks. <https://tailblocks.cc/>
* daisyUI — Tailwind CSS Components. (n.d.). daisyUI — Tailwind CSS Components. <https://daisyui.com/>