

A REPORT OF SEMESTER INDUSTRIAL TRAINING
at
[AGBE TECHNOLOGIES]
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF DEGREE OF
BACHELOR OF TECHNOLOGY
(Computer Science and Engineering)



JANUARY 2025 – MARCH 2025

SUBMITTED BY:

NAME: SALONI

UNIVERSITY ROLL NO.: 2124397

DEPARTMENT OF COMPUTER SCIENCE and ENGINEERING
I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY

CONTENTS

Topic	Page No.
<i>Certificate by Company</i>	3
<i>Candidate's Declaration</i>	4
<i>Abstract</i>	5
<i>Acknowledgement</i>	6
<i>About the Company/ Industry / Institute</i>	7-10

<i>Chapter 1: INTRODUCTION</i> <i>1.1 Background of the Topic</i> <i>1.2 Theoretical Explanation</i> <i>1.3 Software/Hardware Tools Learned</i>	11-19
<i>Chapter 2: FIELD OF TRAINING</i> <i>2.1 Technologies, Tools and IDE used in the training</i>	20-25
<i>Chapter 3: TRAINING WORK UNDERTAKEN</i> <i>Task-1 Designing clone of Amazon website</i> <i>Task-2 Designing an E-Commerce website</i> <i>Task-3 API fetching using Redux</i> <i>Task-4 UI/UX Design Phase-Client Project</i>	26-38
<i>Chapter 4: RESULTS AND DISCUSSIONS</i> <i>4.1: Results</i> <i>4.2: Discussions</i>	38
<i>Chapter 5: CONCLUSION AND FUTURE SCOPE</i> <i>5.1: Conclusion</i> <i>5.2: Future Scope</i>	39
<i>Chapter 6: REFERENCES</i>	40

Certificate by AgBe Technologies



AgBe Technologies LLP

307, Palm Square, Emaar, Golf Course Extension Road, Gurugram

www.AgBeIndia.com

20-March-2025

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Ms. Saloni Narang, a student of I.K. Gujral Punjab Technical University, has completed her training at Agbe Technologies LLP in Frontend Development.

Her internship was conducted under the Engineering Department from **3rd February 2025 to 17th March 2025**, under the supervision of Mr. Ashutosh Verma (Training Coordinator). She actively participated in various real-time projects and tasks related to HTML, CSS, JavaScript, and frontend frameworks. She demonstrated a strong understanding of frontend principles, UI/UX design basics, and responsive web development.

During the internship, Saloni completed the modules assigned in the project with sincerity and punctuality and picked up technological aspects very well.

AgBe wishes the candidate a great career and a bright future ahead.

A handwritten signature in blue ink that reads "Manju Aggarwal".

Manju Aggarwal
Designated Partner & Managing Director

AgBe Technologies LLP
Gurugram

I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY, KAPURTHALA

CANDIDATE'S DECLARATION

I SALONI hereby declare that I have undertaken two months industrial training at “Agbe Technologies” during a period from January to March in partial fulfilment of requirements for the award of degree of B.Tech (Department of computer Science Engineering) at I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY, KAPURTHALA. The work which is being presented in the training report submitted to Department of Computer Science and Engineering at I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY, KAPURTHALA is an authentic record of training work.

Signature of the Student

The six months software training Viva–Voce Examination of _____ has been held on _____ and accepted.

Signature of Internal Examiner

Signature of External Examiner

ABSTRACT

At Agbe Technologies, I worked as a Software Developer Intern (SDE), primarily focusing on the frontend development of web applications. In this role, I collaborated with both the design and development teams to bring client requirements to life.

The process began with gathering client specifications, followed by creating initial designs using Canva. These designs were then presented to the clients for feedback, and any necessary changes were incorporated before moving forward with the development phase.

As part of the frontend team, my primary responsibility was to ensure that the websites were not only visually appealing but also fully responsive and optimized for a seamless user experience across devices.

I worked closely with both frontend and backend developers, ensuring smooth integration of the user interface with the backend systems. My focus was on creating high-quality, responsive layouts, and ensuring that the websites adhered to best design practices, performance standards, and accessibility guidelines.

This internship allowed me to gain hands-on experience in web development and sharpen my skills in UI/UX design, as well as enhance my teamwork and communication abilities in a fast-paced environment.

Throughout my internship at Agbe Technologies, I also developed a deeper understanding of the agile development process and the importance of client collaboration. Working in close coordination with the design and backend teams, I contributed to the iterative development cycle, ensuring timely delivery of features and bug fixes.

Additionally, I learned how to efficiently implement modern frontend frameworks and tools to enhance performance and maintain scalability.

This experience not only strengthened my technical skills but also improved my problem-solving capabilities, as I was required to troubleshoot issues related to design, responsiveness, and user interaction.

Overall, this internship significantly enhanced my proficiency in frontend development and solidified my ability to work effectively within a collaborative software development environment.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Dr. Manju Aggarwal** for providing me with the opportunity to work as an intern at AgBe Technologies. I am truly grateful for this valuable experience.

It is my pleasure to express with deep sense of gratitude to Mr. **Prem**, Senior Software Developer, AgBe Technologies, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert.

I would like to express my gratitude Mr. **Aushtosh Verma**, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Head of Department (Computer Science and Engineering), all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

ABOUT THE COMPANY

COMPANY BRIEF INTRODUCTION

AgBe Technologies is a leading technology company that specializes in providing innovative and tailored solutions for various industries, including agriculture, healthcare, and education. Founded in 2017, AgBe Technologies was established with a vision to leverage cutting-edge technologies to improve business processes, enhance efficiency, and drive sustainable growth. With a team of highly skilled professionals, the company focuses on delivering value through AI-powered solutions, cloud-based platforms, and data-driven insights. Over the years, AgBe Technologies has successfully developed a strong presence in both local and international markets, helping clients streamline operations and stay ahead of the competition.



Figure 1:Agbe Logo



Figure 2: AgBe Technologies

Mission of AgBe Technologies

The mission of AgBe Technologies is to empower businesses by providing innovative technology solutions that drive growth, efficiency, and sustainability. By leveraging the power of AI, machine learning, and big data, the company aims to create customized solutions that meet the specific needs of various sectors. AgBe Technologies is committed to fostering long-term relationships with clients by delivering exceptional value, superior customer service, and innovative products. The company's mission is also focused on creating a positive impact on society by contributing to the digital transformation of industries and supporting sustainable practices.

Vision of AgBe Technologies

AgBe Technologies envisions becoming a global leader in delivering cutting-edge technology solutions across a wide range of industries. The company strives to be at the forefront of innovation by continuously evolving its services and embracing emerging technologies. AgBe's vision is to revolutionize industries such as agriculture, healthcare, and education by providing tools that empower organizations to make data-driven decisions, optimize operations, and drive sustainable growth. Through its solutions, AgBe Technologies aims to build a more connected and technologically advanced world.

Objectives of AgBe Technologies

- **Short-term objectives (1-2 years):**
 - Launch new AI-driven solutions for the agriculture and healthcare sectors to improve decision-making processes.
 - Expand the company's presence in international markets, with a focus on Europe and North America.
 - Enhance customer support services by implementing 24/7 customer care & technical support.
 - Achieve a 20% increase in revenue through the introduction of new product offerings.
- **Medium-term objectives (3-5 years):**
 - Develop partnerships with key stakeholders in the technology and business sectors to enhance product offerings.
 - Launch a training and certification program for businesses to help them adopt digital technologies effectively.
 - Build a robust research and development (R&D) department to focus on next-generation tech such as blockchain and Internet of Things (IoT).
- **Long-term objectives (5-10 years):**
 - Establish AgBe Technologies as a global leader in digital transformation, with a focus on sustainability.
 - Create a comprehensive suite of integrated solutions that span multiple industries and business functions.
 - Develop eco-friendly technologies that reduce the environmental impact of business operations across industries.

Technologies Offered by AgBe Technologies

- **AI and Machine Learning Solutions:**

AgBe Technologies offers advanced AI-driven platforms that help businesses make informed - decisions. The company's machine learning models are designed to improve business operations through predictive analytics and data automation.
- **Cloud Computing Solutions:**

With its secure and scalable cloud infrastructure, AgBe Technologies helps organizations migrate to the cloud, ensuring a smooth transition, enhanced productivity, and cost savings.
- **Custom Software Development:**

The company provides bespoke software development services, from mobile applications to enterprise-grade solutions, designed to meet the unique needs of each client.
- **Consulting Services:**

AgBe Technologies offers expert consulting to help businesses optimize their digital strategies. This includes business process reengineering, technology adoption, and digital transformation.

- **Data Analytics and Insights:**

The company specializes in transforming raw data into actionable insights that businesses can use to make better strategic decisions.

- **IoT Solutions:**

AgBe Technologies also provides Internet of Things (IoT) solutions that enable organizations to monitor, control, and optimize processes in real time across various industries like manufacturing, agriculture, and logistics.

Services Offered by AgBe Technologies

- **Software Development**

We offer more than just web and mobile application development; we provide comprehensive guidance across the entire digital ecosystem. From fundamental aspects such as Website development to advanced solutions and any other custom mobile and web applications, we ensure that you receive the most suitable technology solutions for your specific needs.

- **Digital Products**

We believe that certain challenges are common across industries, making custom-built solutions unnecessary for every individual client. Instead, developing scalable products offers a more efficient approach, enabling businesses to immediately serve their customers without the delays. Our product portfolio includes a diverse range of solutions, such as an Automobile CRM platform, an e-commerce portal, and an Audit management system that has been successfully operating in multiple countries for several years.

- **Hosting Services**

We provide end-to-end infrastructure hosting solutions on AWS, Azure, and other leading cloud providers, covering everything from domain registration to hosting management. Our services include server and database administration, security optimization, and regular maintenance, ensuring a scalable, secure, and high-performing infrastructure for your business.

- **IT Consultancy**

As a technology-driven organization, we take pride in offering expert advisory services on all aspects of IT strategy, infrastructure, and start up technology solutions. Leveraging Generative AI (Gen AI) in the Software Development Lifecycle (SDLC), we enhance requirement analysis, code generation, testing automation, and deployment optimization, ensuring faster development cycles and higher software quality.

- **Cyber Security**

We offer comprehensive cybersecurity solutions designed to protect businesses from evolving threats and vulnerabilities. Our services include SIEM (Security Information and Event Management), SOAR (Security Orchestration, Automation, and Response), VAPT (Vulnerability Assessment and Penetration Testing), cloud security, and endpoint protection.

- **User Experience**

We provide holistic user experience (UX) design services that prioritize usability, accessibility, & engagement across digital platforms. Our approach integrates human-centred design principles with AI-driven insights to create intuitive, visually compelling, and seamless experiences for web and mobile applications.

Industries Served

✓ Agriculture

- **Smart Farming:** AgBe Technologies brings IoT and AI solutions to the agriculture sector, helping farmers make informed decisions regarding irrigation, crop health monitoring, and pest control.
- **Supply Chain Efficiency:** AgBe has developed AI-powered platforms that streamline the agricultural supply chain, ensuring the quick, efficient delivery of produce from farms to markets, reducing waste, and improving profitability.

✓ Healthcare

- **AI-Powered Diagnostics:** AgBe's AI solutions assist healthcare providers with diagnostic tools that analyze medical images, patient data, and symptoms to improve accuracy and speed.
- **Predictive Healthcare:** AgBe's predictive analytics help identify early warning signs of diseases and potential healthcare trends, allowing providers to offer proactive treatments.
- **Telemedicine and Remote Monitoring:** AgBe's solutions enable healthcare professionals to monitor patients remotely, improving access to healthcare services, particularly in underserved or rural areas.
- **Cost Reduction:** By automating administrative tasks such as billing and patient scheduling, AgBe's healthcare solutions improve efficiency and reduce operational costs.

✓ Education

- **Personalized Learning:** AgBe has developed AI-powered educational platforms that offer personalized learning experiences based on the needs and performance of individual students.
- **Data-Driven Insights for Educators:** The Company's data analytics tools provide valuable insights to educators, enabling them to make informed decisions about curriculum adjustments and student performance improvements.
- **Virtual Classrooms and Remote Learning:** AgBe Technologies offers cloud-based learning management systems that facilitate remote education, supporting students and educators in a hybrid learning environment.

Research and Development

- **Innovation Culture:** AgBe Technologies places a high priority on research and development, with a dedicated R&D team focused on developing innovative solutions. Discuss the company's approach to fostering a culture of innovation and its investments in new technologies.
- **Future Technologies:** AgBe is exploring future technologies such as block chain, 5G, or quantum computing to further enhance its solutions, positioning itself as a forward-thinking company in the tech industry.

CHAPTER-1: INTRODUCTION

THE OPPORTUNITY:

During my time at AgBe Technologies, I gained invaluable insight into the inner workings of a tech-driven environment and developed a deeper understanding of how the corporate world approaches problem-solving. AgBe Technologies focuses on providing innovative IT solutions to various industries, where I was able to work closely with cross-functional teams to address client needs. The company prioritizes streamlining processes and ensuring seamless integration between different systems, which allowed me to observe how technology can be leveraged to optimize workflow and enhance productivity. I learned how to adapt quickly to evolving challenges, all while working in a collaborative and dynamic atmosphere that emphasized both individual contributions and team success.

METHODOLOGY:

During my internship at Agbe Technologies, I followed a structured approach to ensure a comprehensive learning experience. Initially, I was given a set of courses to complete, which helped me get familiar with the core technologies and tools used by the company. Following this, I was assigned a mentor who conducted knowledge transfer (KT) sessions, providing me with deeper insights into the company's projects, processes, and systems. Once I had completed my KT sessions, I was given hands-on tasks and assignments to apply my learning in real-world scenarios. This approach allowed me to gradually take on more responsibility and understand the company's operational model. My manager then entrusted me with a project, which focused on integrating solutions for client-specific needs. The overall methodology aimed at preparing me for a potential full-time role while ensuring I understood the day-to-day functioning of the company.

KEY FINDINGS:

- Gained hands-on experience in a fast-paced tech environment.
- Developed a deeper understanding of Agbe Technologies' project management methodologies and client-focused solutions.
- Contributed to ongoing projects by offering fresh perspectives on technical challenges.
- Gained insight into effective collaboration within cross-functional teams.
- Learned how to adapt and align with company protocols and expectations.
- Enhanced my problem-solving skills through direct involvement in real-world assignments.

BENEFITS TO THE COMPANY:

Actualizing an internship program gives organization direct access to ongoing turns of events. As much as assistants will gain from organization, organization can likewise gain from understudies. As current worker's coach and regulate assistants, they'll increase important administration abilities. This can be incredible preparing for a representative who will in the long run involve an administration position. Managing assistants is frequently less unpleasant than regulating previously settled representatives. Coaching and managing others frequently persuade representatives to consider themselves progressively responsible and become powerful pioneers. Along these lines, adding understudies to your organization can upgrade and increase initiative.

At AgBe Technologies, the internship program not only benefited the interns but also provided the organization with significant advantages. By integrating interns into the company, Agbe Technologies had direct access to fresh perspectives and the latest trends in technology. This constant infusion of new ideas and enthusiasm allowed the company to stay current and innovative in its approach. Additionally, current employees had the opportunity to mentor and guide the interns, which helped them develop essential leadership and management skills. Managing interns is often a less stressful experience compared to overseeing seasoned employees, as interns are more adaptable and open to learning.

1.1 BACKGROUND OF THE TOPIC

During my 2-month internship at Agbe Technologies, my primary objective was to gain practical experience in frontend web development, focusing on building scalable, reusable, and modular components for various web applications. This project involved working with modern web development technologies like React, Angular, and Vanilla JavaScript, which are essential for creating dynamic, responsive, and interactive web interfaces.

One of the key technologies I worked with during this internship was Web Components. Web Components are a set of standardized web platform APIs that enable developers to create custom, reusable HTML elements, which can be shared and used across multiple web applications. Web Components consist of three core technologies: Custom Elements, Shadow DOM, and HTML-Templates.

- **Custom Elements:** This technology allows developers to define their own HTML elements with custom behaviour and properties. Custom Elements enable the creation of unique UI components like buttons, sliders, or dropdown menus, which can have specific functionality that standard HTML elements do not provide.
- **Shadow DOM:** This feature enables developers to create a "shadow" DOM tree that is separate from the main DOM tree of the webpage. The Shadow DOM helps encapsulate the styles and functionality of a custom element, preventing any potential style conflicts with other elements on the page. This isolation is key for creating modular, reusable components without affecting the rest of the application.
- **HTML Templates:** HTML Templates allow developers to define reusable chunks of HTML code, which can be dynamically inserted into a web page or web application. This is especially useful for creating complex UI components such as forms or data tables that need to be reused in different parts of the application.

The use of Web Components in web development is particularly advantageous for creating modular & reusable UI components that can be seamlessly integrated into multiple pages or platforms. By encapsulating both styles and functionality within custom elements, developers can enhance the maintainability, scalability, and consistency of their web applications.

ABSTRACT

This report provides a comprehensive overview of my 2-month internship at Agbe, where I gained valuable experience working with frontend technologies. During this internship, I focused on developing scalable and efficient web components, leveraging modern tools to enhance the development of dynamic and responsive user interfaces. The opportunity to work on real-world projects

allowed me to explore key principles of web development, including component-based architecture, automation, and performance optimization. I actively contributed to building user-centric solutions, collaborating with the team to improve the overall functionality and design of the applications. The report highlights the methodologies used in our development processes, my involvement in enhancing the user experience, and the lessons learned throughout the internship.

Introduction:

- Overview of frontend technologies and their significance in modern web development
- Introduction to Agbe's role in creating dynamic and responsive user interfaces for web applications
- Brief history of Agbe and its adoption of modern frontend frameworks and tools to enhance user experience

Understanding Frontend Development:

- Definition and characteristics of efficient frontend design and development
- The role of HTML, CSS, JavaScript, and modern frameworks in creating interactive user interfaces
- Advantages of using scalable, reusable components for maintainable and performance-optimized web applications

Introducing Agbe:

- Overview of Agbe and its key services in the web development ecosystem
- Comparison with other tech companies in the frontend development space
- Agbe's approach to building responsive and user-centred applications

Agbe's Key Technologies:

- Use of React-Js for building reusable, encapsulated web components
- Implementation of state management systems for smoother user experiences
- Integration of APIs to enhance functionality and data handling in frontend applications
- Role of automation tools in improving development workflows and testing processes

Working with Agbe:

- Overview of the internship setup and onboarding process
- Gaining exposure to project-based workflows and development sprints
- Collaborating with teams to integrate ReactJS components into real-world projects
- Hands-on exposure to frontend best practices, design patterns, and optimization techniques

Agbe Tools and Ecosystem:

- Tools used for frontend development (e.g., React, TypeScript, etc.)
- Integration of modern version control systems and CI/CD pipelines for streamlined development

- Documentation practices and collaborative development within the Agbe team
- Resources for continuous learning and keeping up with evolving frontend technologies

Benefits of Agbe's Frontend Solutions:

- Enhanced user experience through performance-optimized and responsive web interfaces
- Scalability and reusability of web components for long-term maintainability
- Increased developer productivity through modern frameworks and automated processes
- Strong focus on collaboration and innovation in the frontend development ecosystem

Real-world Applications:

- Collaboration with clients to develop tailored, dynamic user interfaces
- Case studies of Agbe's success in delivering high-performing web applications for diverse industries
- Implementation of modern frontend frameworks to enhance functionality and user interaction

Challenges and Limitations:

- Challenges in ensuring cross-browser compatibility and responsiveness
- Managing legacy code and integrating new frameworks into existing systems
- Overcoming design and development bottlenecks in large-scale projects

Conclusion:

- Recap of key learnings and insights gained during the internship at Agbe
- Future prospects for Agbe's frontend development strategies and services
- The importance of Agbe's contributions to shaping modern web applications and enhancing user experience

KEY FEATURES & METHODOLOGIES AT AGBE:

Frontend Development at Agbe Technologies:

At Agbe Technologies, I worked as a Software Developer Intern (SDE), where I focused on frontend development for web applications. My role required close collaboration with design and backend teams to transform client requirements into functional and visually appealing web interfaces. Similar to how web components encapsulate functionality for modular reuse, I adopted structured development practices to create scalable and efficient UI components that enhanced the user experience across various platforms.

Client Requirements and Design Iteration:

The development process began with gathering client specifications and understanding their needs. To facilitate the initial design phase, I utilized Canva to create UI mock-ups, which were then presented to clients for feedback. This iterative approach allowed for modifications before moving into the development phase, ensuring alignment with the client's vision and expectations. By integrating design

validation early in the workflow, the development cycle became more streamlined, reducing rework and improving efficiency.

Responsive UI Development:

As part of the frontend team, my primary responsibility was to ensure that the web applications were fully responsive and optimized for seamless user experiences across multiple devices. By leveraging modern frontend frameworks and CSS methodologies, I developed layouts that adhered to best design practices, performance standards, and accessibility guidelines. This approach ensured a consistent user experience, regardless of screen size or device type.

Cross-Functional Collaboration and Integration:

I worked closely with backend developers to integrate user interfaces with server-side functionalities. This required understanding API structures, handling data effectively, and ensuring smooth communication between frontend and backend systems. By maintaining modular and reusable code structures, I contributed to creating a scalable frontend architecture that could easily adapt to future enhancements.

Agile Development and Task Management:

Throughout my internship, I gained practical experience in agile development methodologies, emphasizing iterative progress and client collaboration. Working in sprints, I actively participated in feature development, bug fixing, and performance optimization tasks. My involvement in stand-up meetings and sprint reviews enhanced my ability to communicate progress effectively and adapt to evolving project requirements.

Performance Optimization and Troubleshooting:

In addition to development, I worked on optimizing frontend performance by implementing best practices such as lazy loading, efficient asset management, and minimizing render-blocking resources. Debugging UI inconsistencies and ensuring cross-browser compatibility were key aspects of my role, helping to deliver a polished and reliable product to end users.

By integrating structured development practices, collaboration, and performance-focused enhancements, my experience at Agbe Technologies significantly strengthened my frontend development skills. The exposure to real-world projects and agile workflows provided me with a deeper understanding of scalable web development, reinforcing my ability to contribute effectively in a professional software development environment.

BENEFITS AND IMPACT OF MY CONTRIBUTIONS:

By integrating structured development practices, collaboration, and performance-focused enhancements, my contributions led to noticeable improvements in the development workflow. Some of the key benefits included:

- **Increased Development Efficiency:** Standardizing UI components allowed for faster iteration

and reuse across projects.

- **Improved User Experience:** Optimization efforts resulted in smoother interactions and faster load times, enhancing customer satisfaction.
- **Reduced Manual Effort:** By automating certain repetitive UI tasks, I helped streamline development processes and minimize human error.
- **Enhanced Collaboration:** Effective communication with designers and backend developers ensured seamless integration and a cohesive final product.

WORKFLOW DETAILS AND FUTURE IMPROVEMENTS:

My workflow at Agbe Technologies followed a structured approach:

- **Requirement Gathering:** Understanding client needs and defining objectives.
- **Design Prototyping:** Creating UI mock ups and iterating based on feedback.
- **Development:** Implementing frontend components with best practices in performance and responsiveness.
- **Integration & Testing:** Connecting frontend with backend APIs and performing cross-browser testing.
- **Deployment & Optimization:** Ensuring smooth deployment while fine-tuning performance and resolving bugs.

REAL WORLD PROBLEMS I FACED:

- **Performance Optimization vs. UI Complexity** – Balancing feature-rich, visually appealing UI designs with efficient performance without affecting load times.
- **Cross-Browser Compatibility Issues** – Ensuring that the application looked and functioned consistently across different browsers like Chrome, Firefox, and Safari.
- **Slow API Responses & Asynchronous Handling** – Dealing with delayed or inconsistent API responses and implementing efficient state management to maintain a smooth user experience.
- **Mobile Responsiveness Challenges** – Adapting complex layouts to work seamlessly on various screen sizes, including mobile devices and tablets.
- **Client Requirement Changes** – Handling frequent last-minute design modifications and ensuring they were implemented efficiently without disrupting the development timeline.
- **State Management Issues** – Managing and synchronizing state across different components, especially when working with dynamic user interactions.
- **UI Design Inconsistencies** – Ensuring that designs provided by the UI/UX team were implemented pixel-perfect, considering variations in screen resolutions and rendering engines.
- **Code Maintainability & Scalability** – Writing clean, reusable code while ensuring future scalability as project requirements expanded.
- **Agile Workflow Adaptation** – Adjusting to fast-paced sprint cycles and ensuring that all development tasks were completed within tight deadlines.
- **Deployment & Version Control Challenges** – Managing version control using Git, handling merge conflicts, and ensuring smooth deployment without breaking existing functionality.

1.2 THEORETICAL EXPLANATION

During my internship at Agbe Technologies, I worked extensively on frontend development, focusing on 2124397

on modular UI design, performance optimization, API integration, and user experience enhancement. My contributions involved applying theoretical concepts related to web development best practices, agile methodologies, state management, and system efficiency to build scalable web applications. These concepts played a crucial role in ensuring that the applications met industry standards while being responsive, performant, and user-friendly.

1. COMPONENT-BASED DEVELOPMENT & MODULAR DESIGN

Component-based development is a core principle in modern frontend frameworks like React.js, which enables the creation of reusable, independent UI components. This theoretical approach ensures better code maintainability, modularity, and scalability.

- **Encapsulation of Functionality:** Each UI component is self-contained, with its logic, structure, and styling. This modular approach allows for easier debugging and reuse across projects.
- **Props & State Management:** Passing data between components using props and managing dynamic states effectively ensures seamless interactions between different UI elements.
- **Optimization with Virtual DOM:** Understanding how React's Virtual DOM works helps improve rendering efficiency, minimizing unnecessary re-renders for better performance.

2. RESPONSIVE WEB DESIGN (RWD) & CROSS-BROWSER COMPATIBILITY

Ensuring that web applications work across different devices and browsers is crucial for a seamless user experience.

- **CSS Media Queries & Flexbox/Grid:** Responsive layouts are created using media queries and CSS properties like Flexbox and Grid, ensuring adaptability to different screen sizes.
- **Cross-Browser Testing & Standardization:** Since different browsers interpret CSS and JavaScript differently, techniques such as CSS normalization and vendor-specific properties are applied to maintain consistency across platforms.
- **Performance Considerations:** Optimizing image loading, minimizing CSS/JS file sizes, and reducing unnecessary DOM elements contribute to faster load times and smoother user interactions.

3. STATE MANAGEMENT & ASYNCHRONOUS DATA HANDLING

Efficiently managing application state is a critical aspect of frontend development, particularly when dealing with real-time data and API-driven applications.

- **State Management Libraries (Redux, Context API):** These libraries help manage global state across the application, reducing prop drilling and improving maintainability.
- **Asynchronous API Calls & Data Fetching:** Handling API responses using fetch or Axios while managing loading states, error handling, and caching mechanisms optimizes performance and user experience.
- **Lazy Loading & Code Splitting:** Implementing these techniques helps in reducing the initial load time of the application by dynamically loading components only when needed.

4. AGILE DEVELOPMENT & COLLABORATIVE WORKFLOWS

Agile methodologies ensure efficient development cycles by emphasizing iterative progress, collaboration, and continuous feedback.

- **Sprint-Based Development:** Breaking down tasks into sprints helps manage workloads efficiently and track progress effectively.
- **Task Management Tools:** These tools assist in organizing tasks, assigning priorities, and

ensuring deadlines are met.

- **Code Reviews & Version Control (Git, GitHub):** Maintaining code quality through peer reviews and version control practices enhances collaboration and reduces errors.

5. WEB PERFORMANCE OPTIMIZATION

Optimizing web applications for better performance is essential for enhancing user experience & search engine rankings.

- **Minimization of Render Blocking Resources:** Reducing the number of render-blocking scripts and optimizing CSS delivery help in improving page load speed.
- **Lazy Loading Images & Assets:** Ensuring that images and non-essential scripts load only when required reduces the initial page load time.
- **Reducing Unnecessary Re-renders:** Memorization techniques (use Memo, use Callback) and efficient state updates minimize the number of re-renders in React applications.

6. API INTEGRATION & DATA SECURITY

Integrating frontend applications with backend APIs involves handling data securely and efficiently.

- **RESTful API Principles:** Ensuring proper HTTP methods (GET, POST, PUT, DELETE) are used based on the type of operation being performed.
- **Authentication & Authorization:** Implementing secure authentication mechanisms, such as JWT tokens, OAuth, or session-based authentication, to protect user data.
- **Data Validation & Error Handling:** Proper validation of API responses and structured error handling prevent application crashes and improve robustness.

7. UI/UX PRINCIPLES & DESIGN THINKING

A well-designed UI not only improves usability but also ensures better engagement and accessibility.

- **User-Centred Design:** Understanding user behaviour and designing interfaces that prioritize ease of use and efficiency.
- **Accessibility Standards (WCAG Compliance):** Ensuring that the application is accessible to users with disabilities by incorporating ARIA attributes, proper colour contrasts, and keyboard navigation.
- **Micro-interactions & Feedback Mechanisms:** Adding animations, hover effects, and real-time validation feedback enhances the overall user experience.

8. REAL-TIME DATA HANDLING & DASHBOARDS

Some projects require handling real-time data updates for improved user interactions.

- **Web-Sockets & Polling:** Implementing Web-Sockets or periodic polling to fetch real-time data updates.
- **Charting Libraries for Data Visualization:** Using libraries like Chart.js, Recharts, or D3.js to represent large datasets visually for better analysis.
- **Data Caching Strategies:** Storing frequently accessed data in local storage or caching layers to reduce redundant API calls.

9. FUTURE IMPROVEMENTS & RESEARCH

Frontend technologies are constantly evolving, and staying updated is essential for building cutting edge applications.

- **Exploring Server-Side Rendering (SSR) & Static Site Generation (SSG):** Frameworks like Next.js offer benefits like improved SEO and faster initial load times.
- **Progressive Web Apps (PWAs):** Implementing offline capabilities and service workers to enhance user experience.

- **AI-Driven Personalization:** Leveraging machine learning models to provide dynamic, user-specific content and recommendations.

1.3 SOFTWARE/HARDWARE TOOLS LEARNT

HARDWARE REQUIREMENTS:

1. Laptop/PC Specifications:

- **Processor:** Intel Core i5/i7 (11th Gen or later) / AMD Ryzen 5/7
- **RAM:** Minimum 8GB (Recommended 16GB for better performance)
- **Storage:** SSD (256GB minimum, recommended 512GB+)
- **Graphics:** Integrated GPU (Dedicated GPU optional for design tools)
- **Display:** 1080p resolution or higher (for better UI/UX design clarity)

2. Peripherals:

- **External Monitor (Optional):** 24" Full HD for multi-screen productivity
- **Mouse & Keyboard:** Ergonomic setup for efficient development
- **Headset/Microphone:** For team communication & client discussions

SOFTWARE REQUIREMENTS:

1. Development Tools

- **Visual Studio Code (VS Code):** Primary code editor for frontend development
- **Node.js & npm:** For managing JavaScript dependencies and running the development server
- **Git & GitHub/GitLab/Bitbucket:** Version control system for managing code changes
- **Postman:** For API testing and integration

2. Frontend Technologies

- **React.js:** Modern JavaScript framework used for frontend development
- **Next.js:** For server-side rendering and performance optimization
- **Tailwind CSS:** For styling and responsive UI design
- **Canva:** For UI/UX design and client mock-ups

3. Performance & Debugging Tools

- **Redux Dev-Tools:** For debugging state management in React apps

4. Communication & Collaboration Tools

- **Microsoft Teams:** For team collaboration
- **Google Meet / Zoom:** For client meetings and feedback sessions

5. Deployment & Hosting Tools

- **Vercel / Netlify:** For deploying frontend applications
- **Firebase:** For authentication, hosting, and real-time database
- **Docker:** For containerizing frontend applications

CHAPTER 2: FIELD OF TRAINING

PROJECT SCOPE

The project scope involved developing a set of custom web components that could be used cross-platform, including React, Angular, and Vanilla JS. The web components needed to be easy to use and integrate into existing projects without requiring any additional coding.

METHODOLOGY

Software Development Life Cycle:

SDLC stands for Software Development Life Cycle. It is a structured approach or framework that describes the stages and activities involved in the development of software- applications. The SDLC provides a systematic and standardized way to plan, design, develop, test, deploy, and maintain software systems. It serves as a guide to ensure that software projects are completed efficiently, effectively, and with high quality.

The SDLC typically consists of several phases, which may vary in naming and order depending on the specific methodology or approach being followed. However, the common phases of the SDLC include:

- ✓ **Requirements Gathering:** In this phase, the project team works with stakeholders to understand and document the functional and non-functional requirements of the software. This involves gathering user needs, analyzing business processes, and defining system requirements.
- ✓ **System Design:** During this phase, the high-level architecture and system design are created based on the gathered requirements. The design may include the overall system structure, user interface design, database schema, and integration of external components.
- ✓ **Development:** The development phase involves writing the code according to the design specifications. Programmers or developers implement the software based on the requirements and design, following coding standards and best practices.
- ✓ **Testing:** The testing phase is dedicated to ensuring the quality and reliability of the software. Various testing activities, such as unit testing, integration-testing, system testing, and user acceptance testing, are performed to identify and fix any defects or issues.
- ✓ **Deployment:** In this phase, the software is deployed to the production environment or made available to end-users. It involves activities such as installation, configuration, data migration, and user training.
- ✓ **Maintenance:** Once the software is deployed, it enters the maintenance phase. This involves ongoing support, bug fixes, updates, and enhancements based on user feedback and changing requirements. Maintenance can include corrective, adaptive, perfective, and preventive maintenance activities.

Throughout the SDLC, documentation, version control, and project management practices, are maintained to ensure proper tracking and control of the software development process. Additionally, collaboration, communication, and coordination among team members and stakeholders are vital for successful software development.

Different methodologies and approaches, such as Waterfall, Agile, and DevOps, have their variations and adaptations of the SDLC. Each methodology has its strengths and weaknesses and is suited for different types of projects and organizational contexts. The SDLC provides a structured approach to software development, helping teams deliver high-quality software that meets user requirements and is delivered

on time and within budget.

Agile methodology is an iterative and collaborative approach to project management and software development. It emphasizes flexibility, adaptability, and continuous improvement- throughout the project lifecycle. Agile methodologies, such as Scrum and Kanban, promote collaboration, transparency, and frequent feedback to deliver high-quality products efficiently. Here's a brief overview of Agile methodology:

Principles of Agile: Agile methodologies are guided by the Agile Manifesto, which outlines four key values:

- ✓ Individuals and interactions over processes and tools
- ✓ Working software over comprehensive documentation
- ✓ Customer collaboration over contract negotiation
- ✓ Responding to change over following a plan

These principles prioritize human interactions, product functionality, customer involvement, and adaptability, fostering a flexible and customer-centric approach.

Iterative and Incremental Development: Agile projects are divided into small iterations or time-bound cycles, often called sprints. Each sprint focuses on delivering a functional increment of the product. Iterative development allows for regular feedback, reduces risk, and enables continuous improvement throughout the project.

Roles and Responsibilities: Agile projects typically have specific roles, including:

- ✓ *Product Owner:* Represents the customer or stakeholder and defines the product vision, priorities, and requirements.
- ✓ *Scrum Master:* Facilitates the Scrum process, removes obstacles, and ensures the team adheres to Agile principles.
- ✓ *Development Team:* Cross-functional group responsible for developing and delivering increments of the product.

Scrum Framework: Scrum is a popular Agile framework that emphasizes teamwork, collaboration, and iterative development. It consists of specific events, artifacts, and roles:

- ✓ *Sprint Planning:* The team plans the work to be done in the upcoming sprint, breaking it down into actionable tasks.
- ✓ *Daily Stand-up:* A short daily meeting where team members discuss progress, challenges, and plans for the day.
- ✓ *Sprint Review:* The team demonstrates the completed work to stakeholders and gathers feedback for future iterations.
- ✓ *Sprint Retrospective:* The team reflects on the sprint, identifies improvement areas, and plans changes for the next iteration.

Kanban Methodology: Kanban is another Agile approach that focuses on visualizing the workflow and optimizing the flow of work. It uses a Kanban board to visualize tasks and their status, limiting work in progress, and ensuring a smooth and continuous flow of tasks.

Continuous Integration and Testing: Agile methodologies encourage continuous integration, where code changes are frequently integrated into a shared repository. Continuous integration and testing help

identify and address issues early, reducing risks and improving the overall product quality.

Adaptability and Collaboration: Agile promotes collaboration, transparency, and flexibility. The team frequently engages with stakeholders to gather feedback, incorporate changes, and ensure the project aligns with customer needs. Agile methodologies embrace change, enabling teams to adapt to evolving requirements and market conditions.

Agile Artifacts: Agile projects often utilize artifacts such as user stories, product backlogs, sprint backlogs, and burndown charts to track progress, prioritize work, and ensure visibility and transparency. Agile methodologies offer numerous benefits, including faster time-to-market, improved customer satisfaction, increased team collaboration, and the ability to respond to changing requirements effectively.

However, successful Agile implementation requires active participation, effective communication, and a commitment to continuous improvement from all project stakeholders.

WHY WEB COMPONENTS

Web Components are required for several reasons:

- ✓ **Reusability:** With Web Components, developers can create custom, reusable HTML elements that can be used across multiple web pages and web applications.
- ✓ **Modularity:** Web Components allow developers to encapsulate functionality and styles within custom elements, making it easier to manage and maintain complex web applications.
- ✓ **Standardization:** Web Components are based on standardized web technologies, such as Custom Elements, Shadow DOM, and HTML Templates.
- ✓ **Scalability:** Web Components can help improve the scalability of web applications by allowing developers to create independent, reusable UI components.
- ✓ **Interoperability:** Web Components can be used with other web technologies, such as JavaScript frameworks and libraries, to create powerful and flexible web applications.

Overall, Web Components can help improve the efficiency, maintainability, and scalability of web applications, making them an important technology for modern web development.

The Project Requirements:

Develop a set of web components that can be used across various platforms. Ensure that the components can be easily integrated into existing projects.

The components should be published using NPM, allowing anyone to use them by running a single command in the terminal.

The components should be customizable, allowing users to modify them based on their requirements.

SYSTEM REQUIREMENTS:

Hardware Requirements:

- ✓ Processor: Intel core or above
- ✓ RAM: Minimum 2GB
- ✓ Hard Disk: Minimum 160GB

Software requirements:

- ✓ Operating System: Windows 7 or above
- ✓ Browser: Google Chrome and Mozilla Firefox

- ✓ Frontend Framework: HTML/CSS/Tailwind CSS
- ✓ Language: Java script
- ✓ Text Editor (IDE): Visual Studio Code

Technologies Used:

To develop the web components, we used a range of web development technologies, including:

- ✓ **React:** React is a popular front-end framework for building web applications. We used React to develop some of the components that required dynamic user interactions and data manipulation.
- ✓ **Angular:** Angular is another popular front-end framework that we used to develop some of the more complex components. We found that Angular provided a more structured approach to web development and was particularly useful for building large-scale applications.
- ✓ **Vanilla JS:** We used Vanilla JS to develop some of the simpler components that did not require any external libraries or frameworks.

We also used a range of other tools and technologies, including Node.js, NPM, Git, and GitHub, to manage the project and collaborate with the team.

PROJECT IMPLEMENTATION

The project was implemented in several stages, including scoping the requirements, designing the components, developing the components, testing and debugging, and publishing the components.

SCOPING THE REQUIREMENTS

The first stage of the project involved scoping the requirements and identifying the key components that needed to be developed. We worked closely with the project manager to ensure that the requirements were clearly defined and aligned with the project's overall goals.

DESIGNING THE COMPONENTS

Once we had identified the key components, we started designing the components, including defining the user interfaces, data models, and component architecture. We used tools such as Sketch and Figma to create wireframes and mockups of the components to get a better understanding of how they would look and function.

There are many tools available for building and working with Web Components. Here, are a few popular tools:

- ✓ **Lit-Element:** Lit-Element is a lightweight library for building Web Components that is based on the Lit HTML templating engine. It provides a simple way to create custom elements, manage component state, and handle events.
- ✓ **Polymer:** Polymer is a JavaScript library that provides a set of tools for building Web Components. It includes a number of pre-built UI components, as well as a set of tools for managing data, handling events, and creating custom elements.
- ✓ **Stencil:** Stencil is a tool for building Web Components that is designed to work with popular front-end frameworks like React and Angular. It provides a set of tools for creating and managing custom elements, as well as a set of performance optimizations to improve the speed and efficiency of your Web Components.
- ✓ **Web Components Dev-Tools:** The Web Components Dev-Tools is a browser extension for Google Chrome that provides a set of tools for debugging and profiling Web Components. It

includes a component tree view, a property inspector, and a JavaScript console, among other features.

These are just a few examples of the many tools available for building and working with Web Components. As Web Components continue to gain popularity, it is likely that more tools will be developed to support this technology.

DEVELOPING THE COMPONENTS

After designing the components, we started developing the components using React, Angular, and Vanilla JS, depending on the complexity of the component. We followed best practices for coding and documentation to ensure that the components were easy to read, modify, and maintain.

Web Components are made using three main technologies:

Custom Elements, Shadow DOM, and HTML Templates. Here's an overview of how each of these technologies works:

- ✓ **Custom Elements:** Custom Elements allow developers to define their own HTML elements with custom behavior and properties. To create a Custom Element, developers define a new class that extends the built-in HTML Element class, and then register the new element using the customElements.define() method. For example, a developer could define a new custom element called "my-button" that extends the HTML Button Element class, and then use this element in their HTML code like any other button.
- ✓ **Shadow DOM:** Shadow DOM allows developers to create a "shadow" DOM tree that is separate from the main DOM tree, which can be used to encapsulate the styles and functionality of a custom element. To create a Shadow DOM, developers first create a new Shadow Root object for their custom element, and then attach the Shadow Root to the element using the element.attachShadow() method. Developers can then add HTML elements and styles to the Shadow DOM, which are hidden from the main DOM tree.
- ✓ **HTML Templates:** HTML Templates allow developers to define a chunk of HTML code that can be reused in multiple places within a web page or web application. To create an HTML Template, developers define a new <template> element in their HTML code, and then add the desired HTML content inside the <template> tags. Developers can then use the <template> element in their code to create new instances of the template, which can be customized as needed.

When used together, Custom Elements, Shadow DOM, and HTML Templates allow developers to create modular, reusable HTML components that can be used across multiple web pages and web applications. By encapsulating functionality and styles within custom elements and Shadow DOM, developers can create powerful and flexible UI components that are easy to manage and maintain.

TESTING AND DEBUGGING

Once we had developed the components, we tested them extensively to ensure that they were working as expected and that they could be easily integrated into existing projects. We used a range of testing frameworks and tools to automate the testing process and identify any bugs or issues.

PUBLISHING THE COMPONENTS

Finally, we published the components using NPM, making them available to anyone who wanted to use them. We provided detailed documentation and examples to help users get started with the components and customize them based on their requirements.

CHALLENGES FACED

During the project, I faced several challenges, including:

- ✓ **Cross-platform compatibility:** Ensuring that the components could be used across different platforms, including React, Angular, and Vanilla JS, was a significant challenge. I had to ensure that the components were designed and developed to work seamlessly across all platforms, which required a significant amount of testing and debugging.
- ✓ **Component customization:** Another challenge I faced was developing components that could be easily customized by users based on their requirements. I had to ensure that the components were modular and flexible, allowing users to modify them without affecting their core functionality.
- ✓ **Project management:** Managing a project of this scope and complexity required careful planning and coordination. I had to ensure that the project was completed within the given timeline, while also ensuring that the components met the project requirements and quality standards.

Despite these challenges, we were able to successfully develop a set of web components that met the project requirements and could be easily used across different platforms.

FUTURE WORK

While we were able to develop a set of functional and customizable web components, till here is still room for improvement and future development. Some of the future work that could be done includes:

- ✓ Developing more complex and specialized components that cater to specific industries or use cases.
- ✓ Integrating the components with popular web development frameworks and libraries to expand their usability.
- ✓ Developing more comprehensive documentation and examples to make it easier for users to get started with the components.
- ✓ Continuously updating and improving the components based on user feedback and changing technologies.

CONCLUSION

In conclusion, the 2-month internship at AgBe Technologies provided me with practical experience in web development, particularly in developing cross-platform web components that can be easily integrated into existing projects. The project involved using a range of web development technologies, including React, Angular etc. required careful planning, coordination, and testing.

While there were some challenges faced during the project, we were able to successfully develop a set of functional and customizable web components that could be used across different platforms. There is still room for improvement and future development, and I believe that this project has provided a strong foundation for further work in this field.

CHAPTER 3: TRAINING WORK UNDERTAKEN

Task-1: Designing clone of Amazon website

In the first week of my internship at Agbe Technologies, I was introduced to the team and given a detailed overview of the company's mission, ongoing projects, and development workflow. I had the opportunity to interact with senior developers and understand how the organization approaches client-based software solutions. As part of my internship at Agbe Technologies, I was assigned an individual project during the first week — to develop an Amazon Clone as a frontend exercise. I built this project using HTML, Tailwind CSS, and JavaScript, closely mimicking the UI and user experience of Amazon's e-commerce platform.

This project helped me understand key frontend concepts such as component design, responsive layout structuring, DOM manipulation, and user-focused interface development. It also served as a foundation for collaborating effectively on larger-scale web applications during the rest of my internship.

Project Overview

As part of my frontend development work during my internship at Agbe Technologies, I developed a responsive Amazon Clone using HTML, Tailwind CSS, and JavaScript, which closely replicates the UI and user experience of Amazon's e-commerce platform. This project was my initial task at Agbe and served as an excellent opportunity to apply my frontend skills in a real-world context.

Technologies Used

- HTML5 for structuring web pages
- Tailwind CSS for styling and making the design sleek and responsive
- JavaScript to handle interactive elements and logic
- React.js (based on project structure) for component-based architecture and maintainability

🔍 Key Features Implemented

- Modern UI: I used Tailwind CSS to design a clean, consistent, and responsive user interface.
- Responsive Layout: The entire site adjusts seamlessly across different screen sizes, offering a smooth user experience on desktops, tablets, and mobile devices.
- Navigation Bar: I implemented a fixed navbar at the top that includes sections like "Home," "Deals,".
- Product Listings: Multiple product cards are displayed in rows with pricing and images.
- Search Functionality: A functional search bar is implemented to simulate product searches dynamically.

My Contributions

- I was responsible for building the entire frontend from scratch, ensuring pixel-perfect design alignment with Amazon's layout.
- I paid close attention to responsiveness, making sure every section—from the navigation bar to the image sliders and product cards—adjusted correctly across screen sizes.
- I optimized the layout using Flexbox and Grid utilities provided by Tailwind, while also maintaining semantic HTML for better accessibility.
- I structured the code in a scalable way to allow future feature additions like authentication or product filters.

Project Links

- **Live Website:** Amazon Clone - Hosted on Netlify

- **GitHub Repository:** github.com/saloninarang27/amazon_clone

What I Learned

This project taught me how to:

- Translate designs into fully functional web pages.
- Use Tailwind CSS effectively for responsiveness and modular design.
- Create intuitive, dynamic UI components like scrollable sections and search bars.
- Work independently on a large-scale frontend module while maintaining clean, readable code.

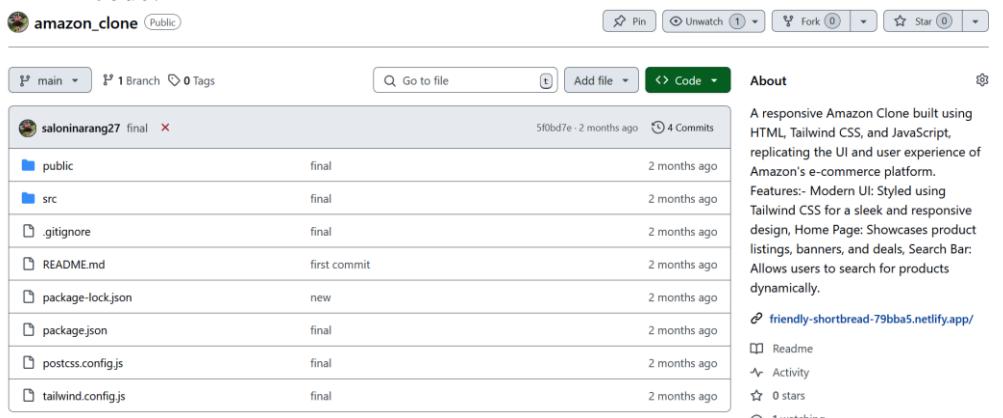


Figure 3.1: Amazon clone Github

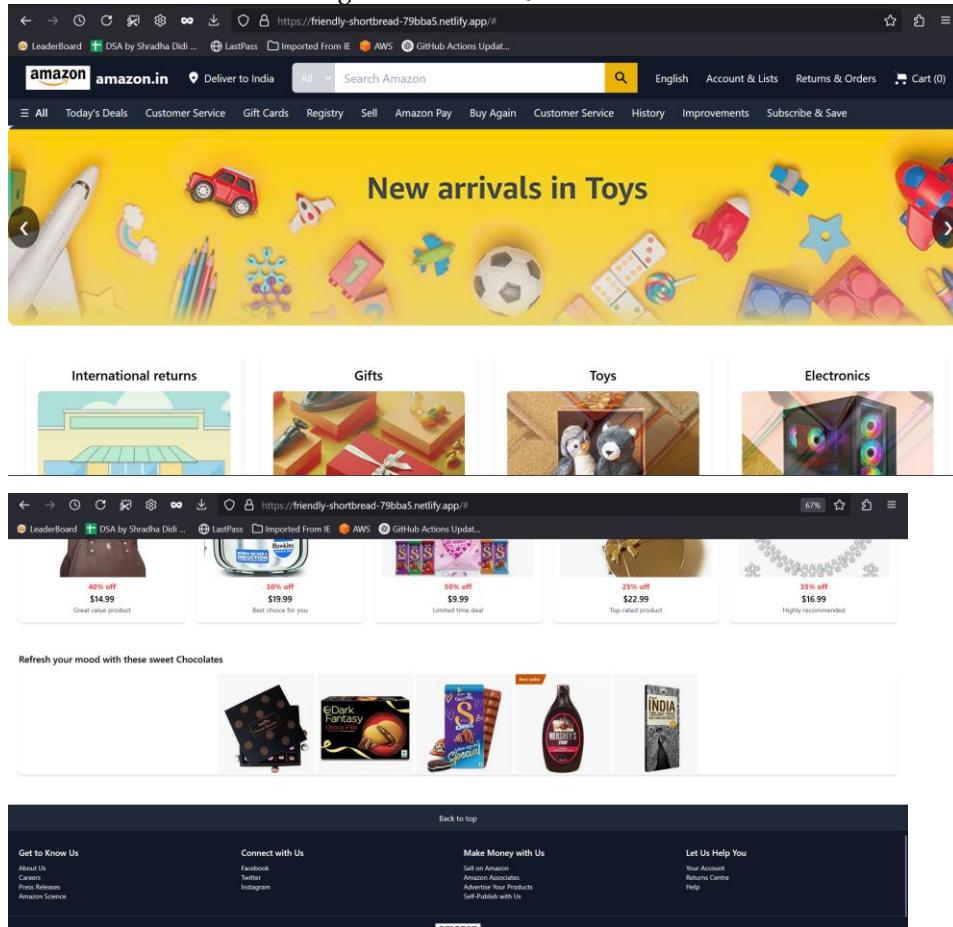


Figure 3.2: Amazon clone view

Task-2: Designing an E-Commerce website

In the second week of my internship at Agbe Technologies, my first task was to build a fully responsive e-commerce website called **Shop-Scape**. The objective was to design and develop a modern online shopping platform using HTML, Tailwind CSS, and JavaScript, focusing on clean UI, product listing, and user interaction.

I began by analysing existing e-commerce platforms and identifying core features such as product grids, cart functionality, and navigation patterns. Based on that, I structured ShopScape to provide an intuitive and visually appealing shopping experience. The project was deployed live and can be accessed [here](#) & the codebase is available on [GitHub](#).

Project Overview

The goal was to build a visually engaging, responsive e-commerce site using HTML, Tailwind CSS, JavaScript, and integrate **Redux** to handle state management. I focused on replicating a smooth UI/UX with dynamic product interactions and a working cart system.

Key Features Implemented

- **Responsive Design:** Optimized for mobile, tablet, and desktop views using Tailwind's utility-first approach.
- **Navigation Bar:** A sticky navbar with links to Home, About, and Contact Us pages.
- **Product Sections:**
 - **Dynamic Product Grid:** Showcased over 20+ products in categorized sections such as "Up to 70% Off" and "Inspired by Your Shopping Trends".
 - **Image Integration:** Downloaded and added real product images to make the UI look professional and realistic.
- **Add to Cart Functionality:**
 - Products can be added to the cart from both the main listing and the individual product detail page.
 - A dedicated **Cart Page** to view selected items, adjust quantities, and view total billing amount.
- **Order Confirmation Flow:** After checking out from the cart, users are redirected to an order confirmation screen.
- **Routing:** Simulated navigation flow using window.Location and dynamic content rendering via JavaScript.

Technical Highlights

- Built entirely with HTML, Tailwind CSS, and vanilla JavaScript (no external frameworks).
- Applied modular code structure for better scalability and maintainability.
- Focused on UI/UX design principles, using consistent spacing, readable fonts, hover effects, and transitions to enhance user interaction.
- Ensured that all components (navbar, banners, cards, buttons) were mobile-first and fully responsive.

Redux Implementation

As part of this task, I learned and integrated Redux for state management:

- Used Redux to manage the global cart state including adding/removing items and tracking totals.
- Implemented Redux actions and reducers to update product lists and cart items across components.
- Ensured real-time cart updates and consistent state across different pages of the app.
- Gained an understanding of the Redux data flow and how to connect components to the store using useSelector() and useDispatch().

Learning Outcomes

- Deepened my understanding of responsive web design using Tailwind CSS.
- Improved skills in DOM manipulation and event handling in JavaScript.
- Understood the importance of product flow and user journey in e-commerce platforms.
- Learned to host and deploy projects using Netlify, and manage source code through GitHub.

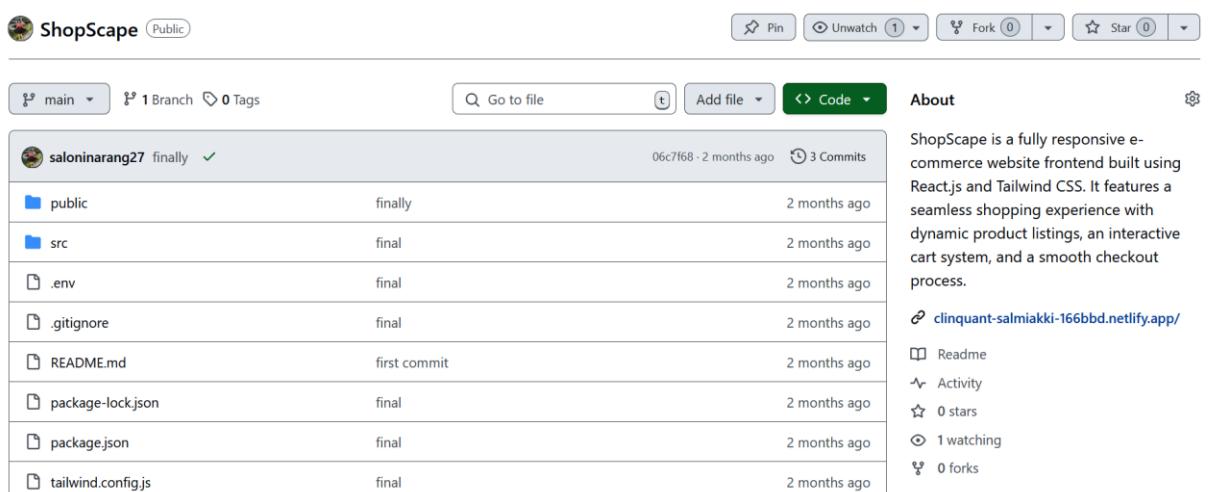


Figure 3.3: Shop-Scape GitHub

The screenshot shows the homepage of the ShopScape website. At the top, there's a navigation bar with links for Home, About, and Contact Us, along with a shopping cart icon showing a count of 2. Below the navigation is a banner for U.S. POLO ASSN. featuring a woman with long hair and a "Flat 50% Offer" for men's wear. The main content area displays four product categories in a grid: Stylish Sneakers (₹2,499), Casual T-Shirt (₹999), Trendy Backpack (₹1,499), and Smart Watch (₹3,999). Each item has an "Add to Cart" button below it. Below this grid are two more rows of products: a white handbag and perfume bottle, and a pair of sunglasses and a pair of earbuds. The footer mirrors the top navigation and includes a "Proceed to Checkout" button.

ShopScape

Home About Contact Us

ShopScape

Find the best products at amazing prices!

U.S. POLO ASSN.
SINCE 1890

Flat 50% Offer
MEN'S WEAR

Stylish Sneakers
₹2,499

Casual T-Shirt
₹999

Trendy Backpack
₹1,499

Smart Watch
₹3,999

Add to Cart **Add to Cart** **Add to Cart** **Add to Cart**

RENEE

Leather Boots

Classic Leather Belt

Total: ₹5,198

Proceed to Checkout

Figure 3.4: Shop-Scape view

Task-3: API Data Fetching Using Redux

As part of my continued learning during the second week of my internship at Agbe Technologies, I was assigned a task focused on fetching API data and managing it using Redux. This helped me gain deeper hands-on experience with asynchronous data handling in modern frontend development.

Live Project: [Visit Here](#)

Source Code: [GitHub Repository](#)

Objective

The goal was to fetch data from a public API (<https://dummyjson.com/products>) and display it in a clean, interactive UI while managing the state efficiently using Redux. This task not only tested my understanding of Redux Toolkit but also deepened my knowledge of handling side effects using redux-thunk.

What I Built

- A functional UI that fetches product data from the Dummy JSON API.
- Displayed each product as a card component with key details like product name, image, category, and price.
- Implemented a clickable detail view for each product that shows extended information.
- Designed a responsive layout using Tailwind CSS.

Redux Integration Highlights

- Created a Redux store using Redux Toolkit to manage global state.
- Used async actions (thunks) to perform API calls and dispatch results to the reducers.
- Utilized Redux slices to handle loading, success, and error states.
- Accessed and manipulated the global state using useSelector and useDispatch.

Key Learnings

- Gained hands-on experience in handling asynchronous operations with Redux Toolkit and Thunks.
- Learned how to create reusable Redux slices for scalable state logic.
- Understood how to implement loading indicators and error handling based on async Redux states.
- Strengthened my skills in API integration, data parsing, and conditional rendering.

Takeaways

This project helped me understand how Redux can simplify the complexity of state management, especially when dealing with API-driven applications. It also gave me a clearer view of how to maintain clean separation between UI and business logic in larger codebases.

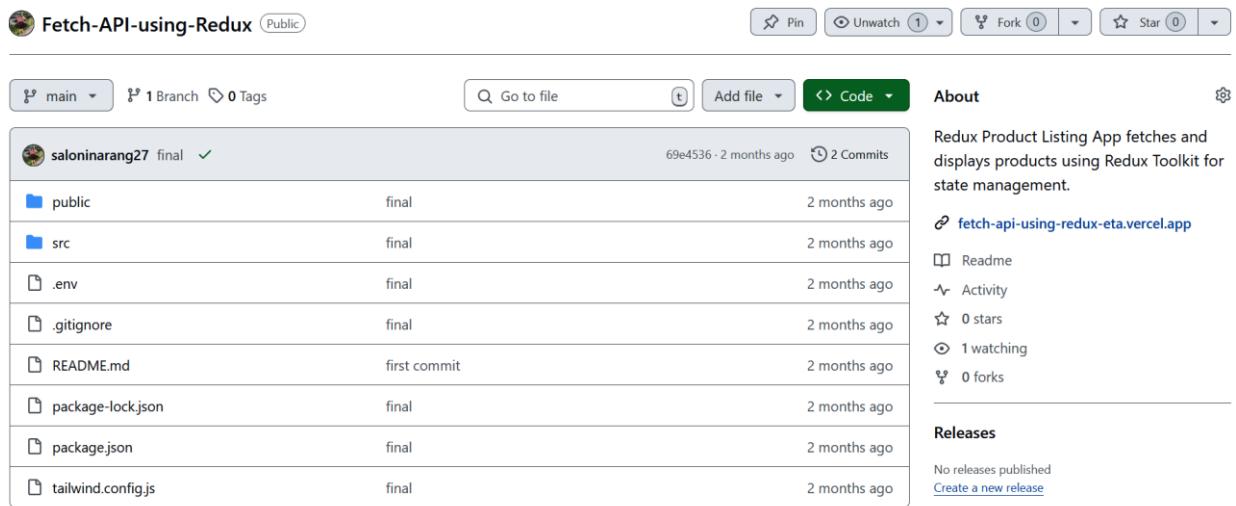


Figure 3.5: Fetch Api GitHub Products

The page displays four product cards:

- Essence Mascara Lash Princess**: \$9.99. Description: The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening effects. Achieve dramatic lashes with this long-lasting and cruelty-free formula.
- Eyeshadow Palette with Mirror**: \$19.99. Description: The Eyeshadow Palette with Mirror offers a versatile range of eyeshadow shades for creating stunning eye looks. With a built-in mirror, it's convenient for on-the-go makeup application.
- Powder Canister**: \$14.99. Description: The Powder Canister is a finely milled setting powder designed to set makeup and control shine. With a lightweight and translucent formula, it provides a smooth and matte finish.
- Red Lipstick**: \$12.99. Description: The Red Lipstick is a classic and bold choice for adding a pop of color to your lips. With a creamy and pigmented formula, it provides a vibrant and long-lasting finish.

Powder Canister

ID: 3

Description: The Powder Canister is a finely milled setting powder designed to set makeup and control shine. With a lightweight and translucent formula, it provides a smooth and matte finish.

Category: beauty

Price: \$14.99

Discount: 18.14%

Rating: ★ 3.82

Stock: 59 units

Brand: Velvet Touch

SKU: 9EN8WLT2

Weight: 8 kg

Dimensions: 24.16 x 10.7 x 11.07 cm

Warranty: 2 year warranty

Shipping: Ships in 1-2 business days

Availability: In Stock



Figure 3.6: Fetch Api view

Task-4: UI/UX Design Phase – Client Project

Role & Involvement

As part of the frontend design team, I actively contributed to a real-world client project that moved from requirement gathering to final UI/UX design handoff.

Key responsibilities included:

- Attending client briefings to gather expectations and feature priorities.
- Contributing to team discussions for ideating layout structures and component hierarchies.
- Designing low- and mid-fidelity wireframes in Figma aligned with business goals.
- Collaborating with developers to ensure feasibility and responsiveness of designs.

Key Design Expectations & Features

The **MVP Audit Pro** platform is a comprehensive, feature-rich solution designed to facilitate efficient audit management and organizational oversight. The platform is equipped with a range of functionalities that enable seamless operation, customizable features, and real-time reporting.

Feature Overview:

1. **Login Page:**
The secure entry point of the platform, enabling authenticated access for users to engage with the audit management system.
2. **Dashboard:**
The **Dashboard** provides a high-level overview of audit performance, presenting key metrics in a visual format. It includes:
 - **Checksheet Status Overview:** A diagrammatic representation of the audit progress, showcasing the total count of completed (30), submitted (20), and pending (20) checks.
 - **Organization Score:** A detailed view of the organization's performance across different parameters, broken down by month for trend analysis.
 - A **Select** button that allows users to filter and view data by geographical regions (city, state, country), ensuring location-specific insights are easily accessible.
3. **Organization Management:**
This section enables the creation of new organizations and the addition of departments, which are displayed in a structured table format. The feature allows for the seamless integration of departments into the audit system, ensuring accurate task allocation and departmental oversight.
4. **Outlets:**
The **Outlets** feature allows users to assign specific departments to various audit outlets. This section also provides the ability to create internal users for each department, with user profiles that include key details such as department, role, and status, thereby streamlining internal audit assignments and responsibilities.

5. Action Plans:

The **Action Plans** table tracks the status of check sheets across audits. It includes the following columns for detailed monitoring:

- **Checksheet**
- **Reviewer**
- **Date**
- **Location**
- **Status**

This functionality ensures that action items are systematically managed, reviewed, and updated in a timely manner.

6. Schedule Audit:

The **Schedule Audit** feature allows for comprehensive audit planning by specifying critical parameters, such as:

- What will be audited
- The location of the audit
- The designated auditor and reviewer
- The scheduled date for the audit

This feature enables meticulous audit preparation, ensuring that audits are conducted as per the defined timelines and responsibilities.

7. Settings:

The **Settings** section provides users with the ability to make platform customizations, such as:

- Modifying fonts and themes for a personalized user experience
- Changing account passwords to maintain system security and privacy

8. Logout:

A simple yet essential feature for securely logging out of the platform, ensuring data protection and confidentiality upon session completion.

The **MVP Audit Pro** platform combines ease of use with robust functionality, making it an indispensable tool for managing audits, tracking organizational performance, and ensuring effective internal controls.

Key capabilities:

1. **User-Friendly Interface:** A secure login system and a well-structured navigation to allow users to easily access and manage audit-related tasks.
2. **Organizational and Departmental Management:** The ability to create and manage organizations and departments, assign internal users to specific roles, and track audit progress at a departmental level.
3. **Geographical Customization:** The ability to filter and view audit data by specific regions (city, state, country), allowing for detailed analysis and reporting based on location.
4. **Audit Tracking and Status Overview:** Visual representation of check sheet progress, including completed, submitted, and pending tasks, alongside detailed organizational performance metrics.
5. **Action Plan Management:** A structured table to track and manage the status of checks, with clear columns for key audit details such as reviewer, date, location, and status.

6. **Audit Scheduling:** The ability to schedule audits by defining key parameters, including what to audit, where, who will perform the audit, who will review it, and the scheduled time.
7. **Customizability:** The option to adjust the platform's visual theme and other settings to meet organizational preferences, as well as the ability to change user passwords for security purposes.
8. **Secure User Logout:** A straightforward and secure logout option to ensure data protection when users finish their sessions.

My Contributions

Throughout this phase, I was deeply involved in both design ideation and handoff preparation:

- Converted client inputs into structured wireframes and user flows using Figma.
- Focused on mobile-first responsive layout, optimizing for smaller screen usability.
- Proposed UI improvements like horizontal scroll product sliders, sticky cart, and icon-based filters.
- Participated in design audit meetings, noted feedback, and applied revisions iteratively.
- Helped create a component library (e.g., buttons, input fields, modals, cards) for modular development.
- Maintained continuous coordination with the dev team to ensure design implementability and adherence.

Tools & Design Techniques Used

- Figma for wireframing, prototyping, and style guide management.
- Colour Psychology & Typography Matching for visual hierarchy and brand alignment.
- Flexbox/Grid-based Layout Planning for responsiveness.
- UI/UX Best Practices like Hick's Law (simple choices), Fitt's Law (target size), and accessibility (WCAG basics).

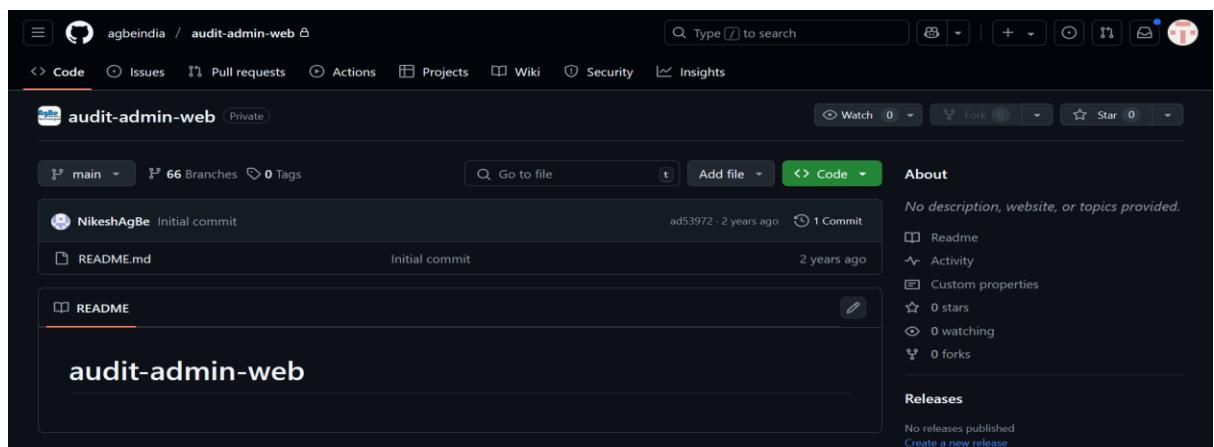


Figure 3.7: Admin Audit GitHub AgBe

The image displays three screenshots of the MVP Audit Pro application, showing the login screen, the dashboard, and the organization creation screen.

Login Screen: The first screenshot shows a login form titled "MVP AUDIT PRO" with the sub-instruction "Welcome Back" and "Please enter your login details." It features two input fields for "Email" and "Password" with placeholder text "Enter Email" and "Enter Password" respectively, and a blue "Login" button.

Dashboard: The second screenshot shows the main dashboard. On the left is a sidebar with navigation links: Dashboard, Organizations, Outlets, Organization Setup, Check Sheet Builder, and Schedule Audit. The top right corner shows a user profile "InnoSoft" and a greeting "Hi, Randhir". The dashboard area contains four cards with statistics: "Total Users" (130), "Total Reviewers" (64), "Total Auditors" (56), and "Total Organizations" (25). Below these are two charts: a donut chart titled "Check Sheet Status" showing a total of 50, and a line graph titled "Organization Score" showing a month-over-month increase from approximately 105 to 140.

Organization Creation: The third screenshot shows the "Create Organizations" and "Create Departments" sections. The sidebar remains the same. In the "Create Organizations" section, there is a field for "Organization Name" with placeholder "Enter Organization Name" and a "Create New Organization" button. In the "Create Departments" section, there is a note: "You can create separate departments in the organization for which you want to create audits." Below it are fields for "Select Organization" (a dropdown menu) and "Department Name" (with placeholder "Enter Department Name") along with a "Create New Department" button. At the bottom of this section is a link "Organizations List".

The figure consists of three vertically stacked screenshots of the MVP AUDIT PRO application, showing different sections of the platform:

- Screenshot 1: Outlets Management**
The URL is localhost:3000/outlets. The page title is "MVP AUDIT PRO". On the left sidebar, the "Outlets" option is selected. The main area is titled "Outlets" and contains fields for "Outlet Name" (with placeholder "Outlet 1"), "Outlet City" (with placeholder "City 1"), and "Outlet State" (with placeholder "State 1"). A "Create New Outlet" button is at the top right. A note at the bottom says: "You can add departments to the outlets for which you want to create audits".
- Screenshot 2: Organization Setup**
The URL is localhost:3000/organization-setup. The page title is "MVP AUDIT PRO". On the left sidebar, the "Organization Setup" option is selected. The main area is titled "Setup Your Organization" and contains a note: "Create your Company's network offices and allocate auditors". Below it is a dropdown menu titled "Select your Organization" with options: "Select Organization" (selected), "Organization 1", and "Organization 2".
- Screenshot 3: Scheduler Audit**
The URL is localhost:3000/audit-schedulers. The page title is "MVP AUDIT PRO". On the left sidebar, the "Scheduler Audit" option is selected. The main area is titled "Scheduler Audit" and contains four expandable sections: "What to audit?", "Select a Check Sheet", "Where to audit?", and "Who will audit and review?". Below these is another section titled "When to audit?".

Figure 3.8: MVP audit view

CHAPTER 4: RESULTS AND DISCUSSION

4.1: RESULTS

During the course of my internship at AGBE Technologies, I acquired hands-on experience with several modern frontend technologies and development practices. I honed my skills in HTML, CSS, and JavaScript, while gaining proficiency in React.js for building modular and reusable UI components. I also learned how to manage application state effectively using Redux Toolkit, and implemented real-time data fetching from public APIs. My exposure to these tools significantly strengthened my understanding of scalable web application architecture, user interface optimization, and responsive design. Additionally, I explored version control with Git, and practiced converting client feedback into design iterations, thus bridging the gap between design mock-ups and working applications.

On the implementation side, I completed four major tasks that showcased my frontend development capabilities. First, I designed a responsive Amazon website clone, which provided an excellent foundation for understanding layout structuring and navigation. Next, I developed an independent e-commerce platform named Shop-Scape, which included features like product listing, detailed product views, cart management, and order confirmation—all built using React and styled with Tailwind CSS. For the third task, I implemented API fetching using Redux Toolkit, dynamically displaying product data with proper state handling and loading/error mechanisms. Lastly, I actively contributed to a client-based UI/UX design project, where I created wireframes using Canva, collaborated with designers and backend developers, and ensured the UI adhered to client expectations. These projects collectively reflect my learning outcomes and ability to apply them to real-world applications.

4.2: DISCUSSION

The internship experience at AGBE Technologies was deeply enriching from both technical and professional development perspectives. Here are the key insights and discussions:

- ✓ **Real-World Exposure to Agile Development:** Being a part of weekly sprints and stand-ups helped me understand agile workflows, task prioritization, and iterative feature development.
- ✓ **Importance of UI/UX in Client Satisfaction:** I learned that even the smallest details in the user interface can significantly impact client feedback and user retention. Tools like Canva proved useful for mock-ups, while Tailwind CSS ensured speed in styling and responsiveness.
- ✓ **Responsiveness as a Core Goal:** Ensuring responsiveness across multiple screen sizes was a priority. I encountered several layout-breaking challenges which were solved using a combination of CSS Grid, Flexbox, and conditional rendering in React.
- ✓ **Redux for Scalable Frontend Architecture:** Implementing Redux enhanced my understanding of state management in scalable applications.
- ✓ **Professional Communication and Documentation:** Regular communication with the team helped improve my documentation, version control (via Git), and ability to interpret client expectations and transform them into technical features.
- ✓ **UI/UX Client Project Learnings:** The collaborative nature of the UI/UX phase taught me the importance of iterative feedback loops. From designing low-fidelity wireframes to implementing final designs, each round of feedback added clarity and polish to the product.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

5.1: CONCLUSION

The internship at AGBE Technologies was an immensely enriching journey that provided me with hands-on experience in real-world software development, specifically in the area of frontend web technologies. During this six-week period, I was entrusted with meaningful tasks that helped me gain practical exposure to tools such as HTML, CSS, JavaScript, React.js, Redux, and API integration. I not only learned how to implement individual components and manage state using Redux, but also how to optimize performance, maintain clean code structures, and debug issues effectively.

This opportunity allowed me to work on real client requirements and understand the importance of UI/UX design, responsive layouts, and cross-browser compatibility. Working collaboratively with both design and backend teams helped improve my communication skills and taught me how to work efficiently in a team-oriented and deadline-driven environment. I also learned to gather requirements, iterate based on client feedback, and maintain a user-centric approach to development. These experiences collectively helped strengthen my technical foundation and made me more confident as a frontend developer.

5.1: FUTURE SCOPE

Building upon the knowledge and experience gained during my internship at AGBE Technologies, I plan to:

- ✓ **Advance Frontend Expertise**
Deepen my understanding of modern frontend technologies such as Next.js and TypeScript to build faster and more scalable user interfaces.
- ✓ **Transition into Full-Stack Development**
Learn backend technologies like Node.js, Express.js, and MongoDB to handle server-side logic and databases, enabling full-stack development capabilities.
- ✓ **Improve UI/UX Design Skills**
Enhance my proficiency in Figma, Adobe XD, and motion design principles to create intuitive, user-friendly, and visually engaging web interfaces.
- ✓ **Explore Mobile App Development**
Learn **React Native** for developing cross-platform mobile applications that can reach wider audiences efficiently.
- ✓ **Contribute to Open Source**
Start contributing to open-source projects to gain real-world collaborative experience and become more involved in the developer community.
- ✓ **Build a Strong Developer Portfolio**
Continue working on personal and collaborative projects to showcase my development skills and practical understanding of full-cycle software development.
- ✓ **Stay Updated with Industry Trends**
Follow blogs, attend webinars, and complete online courses to stay up to date with the latest trends, frameworks, and best practices in web development.

CHAPTER-6: REFERENCES

Books

- [1] E. Freeman and E. Robson, *Head First HTML and CSS*, 2nd ed. Sebastopol, USA: O'Reilly Media, 2012.
- [2] M. Haverbeke, *Eloquent JavaScript*, 3rd ed. San Francisco, USA: No Starch Press, 2018.
- [3] C. N. S. Horstmann, *Modern JavaScript for the Impatient*, 1st ed. Boston, USA: Addison-Wesley, 2020.

Online Sources

- [4] Redux Documentation, “Redux Essentials – Learn Redux Toolkit,” [Online]. Available: <https://redux.js.org/tutorials/essentials/part-1-overview-concepts>
- [5] ReactJS Documentation, “React – A JavaScript library for building user interfaces,” [Online]. Available: <https://reactjs.org/>
- [6] Tailwind CSS Docs, “Tailwind CSS – Utility-First CSS Framework,” [Online]. Available: <https://tailwindcss.com/>
- [7] DummyJSON API, “Free Fake REST API for Testing and Prototyping,” [Online]. Available: <https://dummyjson.com/>