**Titles**

*A thesis submitted in partial fulfillment of the requirement for the award of the degree*

*Of*

**Bachelor of Computer Application**

***in***

***Faculty of Computer Technology***

****

***Submitted by***

**Names1**

Roll no: ADTU/0/2024-27/BCAM/049

Names2

Roll no: ADTU/0/2024-27/BCAM/033

*Under the guidance of*

***Dr. Gunikhan Sonowal***

Assistant Professor

Faculty of Computer Technology

**Assam down town University**

**Guwahati-26, Assam**

Session: January-June, 2024

**CONTENTS**

|  |  |
| --- | --- |
| * *Certificate of Approval* | i |
| * *Certificate from Guide* | ii |
| * *Certificate from External Examiner* | iii |
| * *Declaration* | iv |
| * *Acknowledgement* | v |
| * *Abstract* | vi |
| * *List of Figures* | vii |

**Chapter Page No**

|  |  |
| --- | --- |
| 1. **Introduction** |  |
| * 1. Overview of the Project | 1-4 |
| * 1. Motivation |  |
| * 1. Scope and Objective |  |
| * 1. Existing System |  |
| * 1. Problem Definition |  |
| * 1. Proposed System |  |
| 1. **Project Analysis** |  |
| * 1. Project Requirement Analysis | 5 |
| * 1. Gantt Chart | 6 |
| * 1. Advantage & Disadvantage | 7-8 |
| * 1. Project Life Cycle   2. Project Feasibility | 9 |
| 1. **Project Design** |  |
| * 1. System Architecture | 10 |
| * 1. Data Flow Diagram(DFD) | 10-12 |
| * 1. NoSQL Diagram | 13 |
| * 1. Use Case Diagram | 14 |
| * 1. Sequence Diagram | 15 |
| * 1. Schema Diagram | 16 |
| 1. **Project Implementation** |  |
| * 1. Description of the Software Used | 17-18 |
| * 1. User Interface | 18-21 |
| 1. **Testing** |  |
| * 1. Types of Testing | 21 |
| * 1. Test Cases | 22 |
| 1. **Conclusion and Future Scope** |  |
| * 1. Conclusion | 23 |
| *References* | 23 |

****

***Faculty of Computer Technology***

***Assam down town University***

Gandhi Nagar, Panikhaiti, Guwahati- 781026, Assam

**CERTIFICATE OF APPROVAL**

This is to certify that the project report entitled ***“Titles of Project”*** submitted by **Raj Singh**bearing Roll No. ADTU/2021-25/BTech(CTIS)/015, **Pulak Gogoi** bearing Roll No. ADTU/2021-25/BTech(CTIS)/004, **Injamamul Islam**bearing Roll No. ADTU/2021-25/BTech(CTIS)/024 and **Dipjyoti Thakuria** bearing Roll No. ADTU/2021-25/BTech(CTIS)/021, are hereby accorded our approval as a study carried out and presented in a manner required for acceptance in partial fulfilment for the award of the degree of ***Bachelor of Computer Application***under Assam down town University for approval does not necessary endorse or accept every statement made opinion expressed or conclusion drawn as recorded in the report. It only signifies the acceptance of the project report for a purpose which is submitted.

|  |  |
| --- | --- |
| **Date:**  **Place: Guwahati** | **Dr. Utpal Barmam**  Dean, Faculty of Computer Technology  Dean(Additional Charge), Computer Science and Engineering,  Assam down town University  Guwahati |

***Faculty of Computer Technology***

***Assam down town University***

Gandhi Nagar, Panikhaiti, Guwahati- 781026, Assam

**CERTIFICATE FROM GUIDE**

This is to certify that the project report entitled ***“Titles of Project ”*** submitted by **Raj Singh**bearing Roll No. ADTU/2021-25/BTech(CTIS)/015, **Pulak Gogoi** bearing Roll No. ADTU/2021-25/BTech(CTIS)/004, **Injamamul Islam**bearing Roll No. ADTU/2021-25/BTech(CTIS)/024 and **Dipjyoti Thakuria** bearing Roll No. ADTU/2021-25/BTech(CTIS)/021 towards the partial fulfilment of the requirements for the award of the degree of ***Bachelor of Computer Application***under Assam down town University is a bonafide research work carried out by them under my supervision and guidance. This work has not been submitted previously for any other degree of this or any other University.

I recommend that the thesis may be placed before the examiners for consideration of award of the degree of this University.

|  |  |
| --- | --- |
| **Date:**  **Place: Guwahati** | **Dr. Gunikhan Sonowal**  Assistant Professor,  Faculty of Computer Technology  Assam down town University  Guwahati |

****

***Faculty of Computer Technology***

***Assam down town University***

Gandhi Nagar, Panikhaiti, Guwahati- 781026, Assam

**Remove this page**

**CERTIFICATE FROM EXTERNAL EXAMINER**

This is to certify that the project report entitled ***“Titles of Project ”*** submitted by **Raj Singh**bearing Roll No. ADTU/2021-25/BTech(CTIS)/015, **Pulak Gogoi** bearing Roll No. ADTU/2021-25/BTech(CTIS)/004, **Injamamul Islam**bearing Roll No. ADTU/2021-25/BTech(CTIS)/024 and **Dipjyoti Thakuria** bearing Roll No. ADTU/2021-25/BTech(CTIS)/021 towards the partial fulfilment of the requirements for the award of the degree of ***Bachelor of Computer Application*** under Assam down town University is a bonafide research work carried out by him under the supervision and guidance of ***Dr. Gunikhan Sonowal***, Assistant Professor, Department of Computer Technology, Assam down town University, Guwahati has been examined by me and found to be satisfactory.

I recommend the thesis for consideration for the award of the degree of ***Bachelor of Computer Application***under Assam down town University.

**Date:**

**Place: Guwahati**

**DECLARATION**

We, **Raj Singh**bearing Roll No. ADTU/2021-25/BTech(CTIS)/015, **Pulak Gogoi** bearing Roll No. ADTU/2021-25/BTech(CTIS)/004, **Injamamul Islam**bearing Roll No. ADTU/2021-25/BTech(CTIS)/024 and **Dipjyoti Thakuria** bearing Roll No. ADTU/2021-25/BTech(CTIS)/021 hereby declare that the thesis entitled ***“Titles of Project”*** is an original work carried out in the Department of Computer Technology, Assam down town University, Guwahati with exception of guidance and suggestions received from my supervisor, ***Dr. Gunikhan Sonowal***, Assistant Professor, Department of Computer Technology, Assam down town University, Guwahati. The data and the findings discussed in the thesis are the outcome of my research work. This thesis is being submitted to Assam down town University for the degree of ***Bachelor of Computer Application”.***

|  |  |
| --- | --- |
| **Mr. Students Name**  Enrolment:  Semester  Programme  Faculty of Computer Technology  Assam down town University  Guwahati | **Mr. Students Name**  Enrolment:  Semester  Programme  Faculty of Computer Technology  Assam down town University  Guwahati |

**ACKNOWLEDGMENT**

We would like to extend our heartfelt appreciation to everyone who contributed to the successful completion of this project. Our sincere thanks go to our project team members for their dedication and collaboration throughout the project. Each member played a significant role in shaping the outcome. Special thanks to our supervisor, Dr. Gunikhan Sonowal, for her guidance and valuable feedback, which enriched our work. Lastly, we want to thank our friends for their patience and encouragement during this project. Their believe helped us to stayb motivated and to persevere through difficult times.

**ABSTRACT**

The project delves into the realm of serverless application development using Amazon Web Services and its various services. With a focus on practical implementation, we explore the architecture, design principles and deployment strategies for building serverless applications. Leveraging AWS lambda, dynamodb and other serverless components, we developed a scalable and effective application. Furthermore, we address key considerations such as security, monitoring in the context of serverless architectures. By sharing insights gained from real-world experimentation ,this project equips with foundational knowledge and practical skills essential for leveraging AWS for serverless application development.

**List of Figures**

|  |  |
| --- | --- |
| **Sl no.** | **Name of the figure/chart** |
| **1** | **Gantt Chart** |
| **2** | **System Architecture** |
| **3** | **Data Flow Diagram** |
| **4** | **Context diagram** |
| **5** | **Nosql diagram** |
| **6** | **Use case Diagram** |
| **7** | **Sequence diagram** |
| **8** | **Schema Diagram** |
| **9** | **Wireframes/Ui** |

**1. INTRODUCTION**

**1.1 Overview of the project**

This project delves into the realm of serverless application development using Amazon Web Services as its primary platform. This application will leverage various AWS services such as AWS lambda, API Gateway, DynamoDB and others to provide a scalable and cost effective solution. Through practical implementation and experimentation, key components such as architecture design, development, data management and deployment will be explored. This project will dive in the creation of a fully functional serverless application deployed on AWS. By providing insights and recommendations for adopting serverless architecture this project aims to equip participants with valuable skills and knowledge in leveraging AWS for building scalable and cost-effective applications.

**1.2 Motivation**

The motivation behind this project stems from the burgeoning significance of serverless computing and its transformative impact on modern application development. With the exponential growth of digital services and the increasing demand for scalable, cost-effective solutions, serverless architecture has emerged as a compelling paradigm shift. By choosing to focus on serverless application development using Amazon Web Services (AWS), this project aims to delve into the core principles, advantages, and practical implications of this innovative approach. Motivated by the opportunity to explore cutting-edge technologies and industry best practices, as well as the potential to contribute to the broader understanding and adoption of serverless computing, this project seeks to equip participants with invaluable skills and insights into the future of cloud-native development.

**1.3 Scope and Objective**

The scope of this project encompasses the exploration and implementation of serverless architecture utilizing Amazon Web Services (AWS) for the development of a targeted application. Key areas within this scope include the selection and utilization of AWS Lambda, API Gateway, DynamoDB, and related services to build scalable, reliable, and cost-effective functionalities. The project aims to provide participants with hands-on experience in designing, developing, and deploying serverless applications within the AWS environment, emphasizing scalability, reliability, and cost-efficiency. Objectives include demonstrating the feasibility of serverless architecture through a proof of concept, evaluating its performance and scalability, and documenting best practices for dissemination to academic and professional audiences. Throughout the project, attention will be given to constraints such as time, resources, and technical limitations, ensuring realistic expectations and achievable outcomes within the defined scope.

**1.4 Existing system**

The existing system encompasses client-server architectures commonly used in application development of serverless computing. In these systems, applications typically rely on a fixed infrastructure of physical or virtual servers to handle various tasks, such as processing requests, managing databases, and serving content to users. However, such architectures often face challenges related to scalability, as provisioning and managing resources to handle fluctuating workloads can be complex and costly. Additionally,systems may struggle to adapt to rapidly evolving user requirements and technological advancements due to their rigid and tightly coupled nature. These limitations underscore the need for a more flexible, scalable, and cost-effective approach to application development, thus motivating the exploration of serverless architecture on AWS as a modern alternative.

**1.5 Problem Definition**

The problem addressed by this project lies in the inefficiencies and limitations of traditional application development approaches, particularly in the context of serving the dynamic needs of college students. Existing systems often struggle to adequately scale and adapt to fluctuating demand, leading to performance decrease and increased operational costs. Moreover, the complexity of managing infrastructure and handling administrative tasks detracts from the focus on delivering innovative features and enhancing user experiences. As such, there is a pressing need for a more agile, scalable, and cost-efficient solution. By leveraging serverless architecture on AWS, this project seeks to address these challenges by providing a flexible, scalable, and cost-effective platform for developing and deploying applications tailored to the needs of various users.

**1.6 Proposed System**

The proposed system for this project centers on leveraging serverless architecture within Amazon Web Services (AWS) to develop a highly scalable, cost-effective, and flexible platform for the users. By harnessing AWS Lambda, API Gateway, DynamoDB, and other serverless services, the system aims to development, deployment, and management processes. Additionally, automatic scaling and resource allocation based on demand will ensure optimal performance and cost efficiency, addressing scalability challenges commonly encountered in traditional architectures. Through comprehensive documentation, the proposed system aims to not only deliver a robust and user-centric solution but also serve as a model for adopting serverless computing paradigms in the development of applications for educational institutions and beyond.

**2. PROJECT ANALYSIS**

**2.1 Project Requirement Analysis**

The project entails the development of a serverless application using Amazon Web Services (AWS). The analysis phase involves a thorough examination of stakeholder requirements, encompassing both functional and non-functional aspects. Through interviews, surveys, and workshops with stakeholders, requirements are gathered, prioritized, and documented, ensuring alignment with user needs and project objectives. Use cases are defined to illustrate Srequirements are detailed to guide the design and development process. Additionally, non-functional requirements such as performance, scalability, security, and usability are identified and documented to ensure that the application meets quality standards and user expectations. Throughout the analysis phase, validation techniques such as prototyping and user feedback sessions are employed to validate requirements and refine the project scope. Ultimately, the analysis phase lays the groundwork for designing and developing a robust and user-centric serverless application that meets the dynamic needs of college students while leveraging the capabilities of AWS for scalability, reliability, and cost-efficiency.

**2.2 Gantt Chart**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **March** | **April** | **May** |
| Information Gathering |  |  |  |
| Analysis |  |  |  |
| Design |  |  |  |
| Coding |  |  |  |
| Testing |  |  |  |
| Analysis |  |  |  |

**2.3 Advantage and Disadvantage**

**Advantage:**

1. Scalability: Leveraging serverless architecture on AWS allows for automatic scaling based on demand, ensuring optimal performance.

2.Flexibility: Serverless architecture offers greater flexibility in development and deployment, allowing for rapid iteration and experimentation with new features and functionalities.

3.Reduced Operational Overhead: With AWS managing the infrastructure and server maintenance, developers can focus more on writing code and less on managing servers, leading to increased productivity and faster time-to-market.

**Disadvantages:**

1.Vendor Lock-in: Developing a serverless application on AWS may result in vendor lock-in, making it difficult to migrate to another cloud provider in the future.

2.Monitoring and Debugging: Debugging and monitoring serverless applications can be more challenging compared to traditional architectures, as you have less visibility into the underlying infrastructure and execution environment.

3.Potential Security Risks: Serverless applications may introduce new security risks, such as data exposure and unauthorized access, if not properly configured and secured.

**2.4 Project Lifecycle**

The project lifecycle involves several key phases from initiation to closure.i.e,

i)Initiation: Conceptualizing the project and gathering initial requirements.

ii) Planning: Detailed planning of tasks, resources, and timelines, including architecture and design.

iii) Execution: Development and implementation of the serverless application on AWS.

iv) Monitoring and Control: Monitoring progress, tracking performance metrics, and controlling project activities.

v) Testing and Quality Assurance: Thorough testing to ensure functionality, performance, and security.

vi) Deployment: Deploying the application to the AWS environment using deployment tools.

vii) Operations and Maintenance: Monitoring, managing, and maintaining the application post-deployment.

viii) Closure: Handing over deliverables, conducting project review, and transitioning the application to long-term support.

**2.5 Project feasibility**

It typically involves evaluating various aspects to determine whether the project is feasible and worth pursuing. Here are the key components of a project feasibility study for this project:

i)Technical Feasibility: Assess if the project is technically achievable with the available resources and technology.

ii)Market Feasibility: Determine if there is sufficient demand and potential for the project in the target market.

iii)Financial Feasibility: Evaluate the project's financial viability, considering costs, revenue potential, and return on investment.

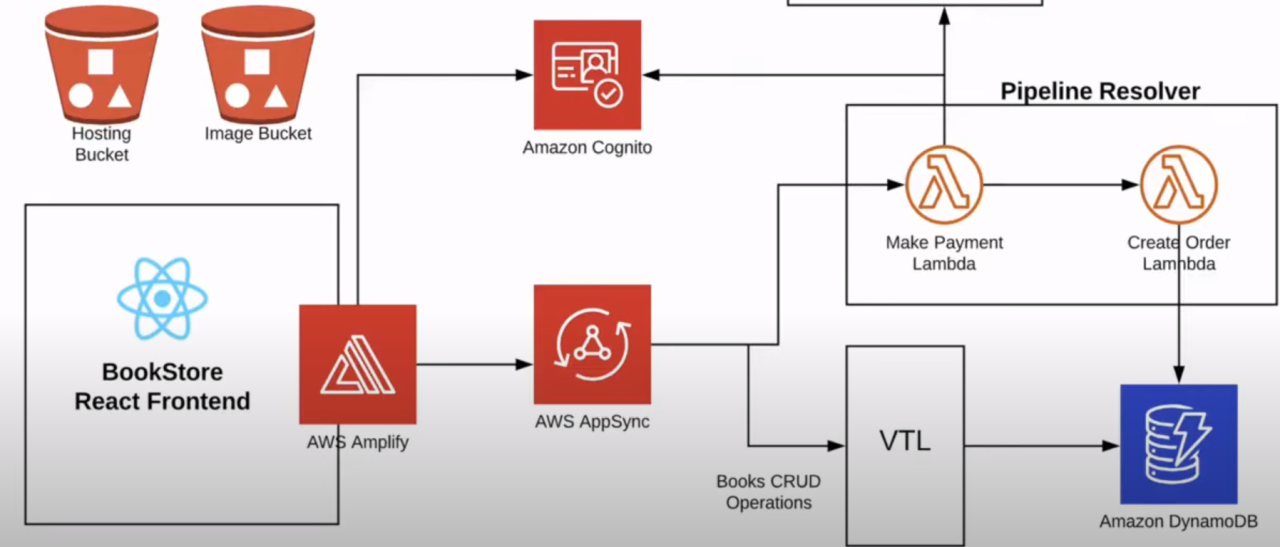
iv)Operational Feasibility: Determine if the project can be effectively implemented and maintained within operational constraints.

v)Legal and Regulatory Feasibility: Ensure compliance with legal and regulatory requirements relevant to the project.

vi)Risk Analysis: Identify and assess potential risks and uncertainties associated with the project.

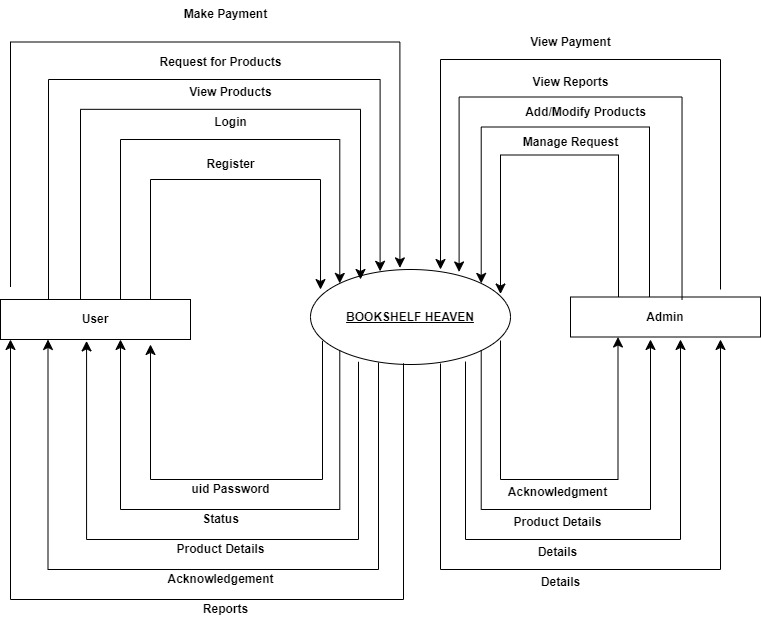
**3. PROJECT DESIGN**

**3.1 System Architecture**

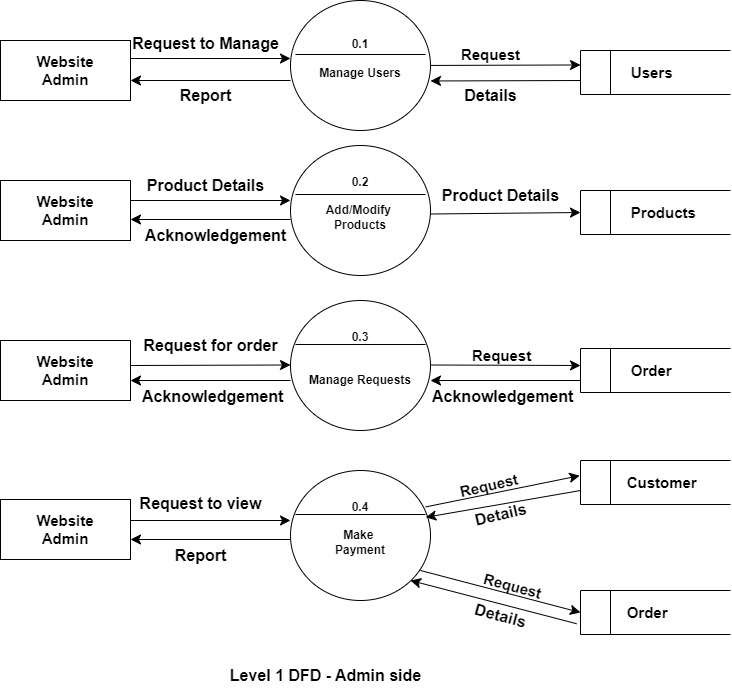


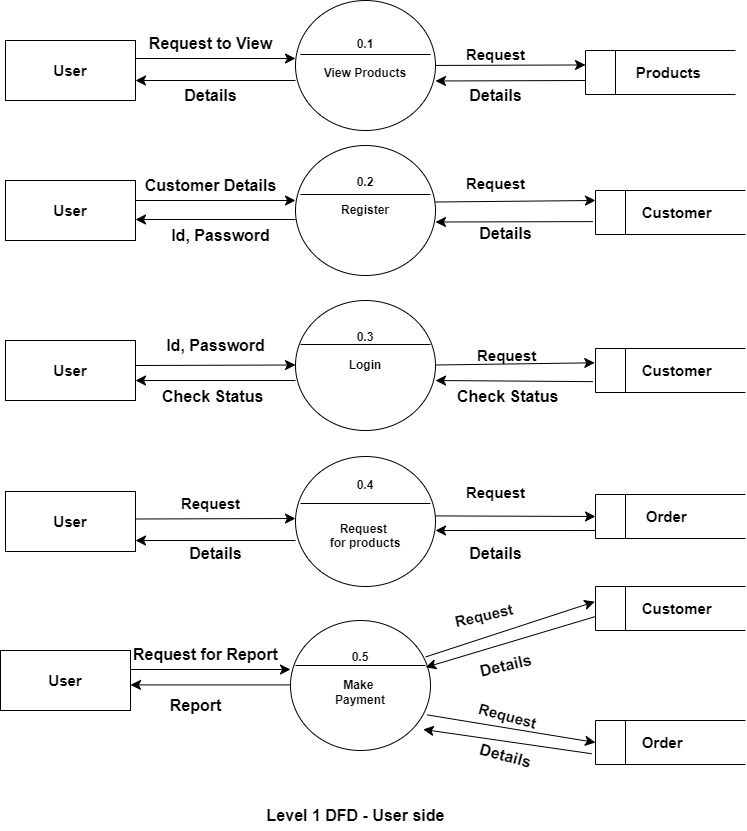
**3.2 Data Flow Diagram**

**3.2.1 Context Diagram**

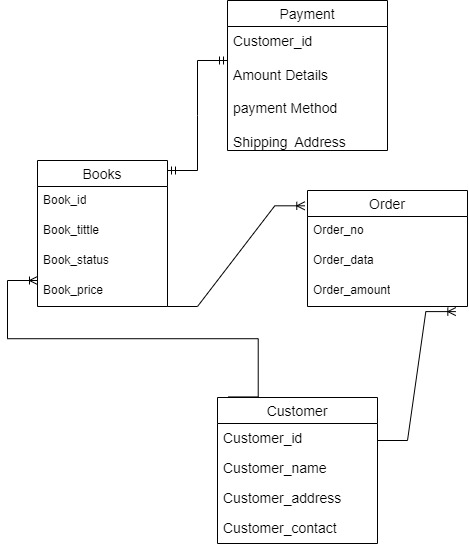


3.2.2 DFD level 1

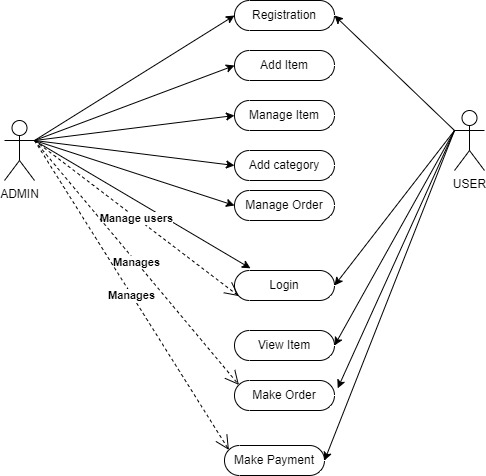




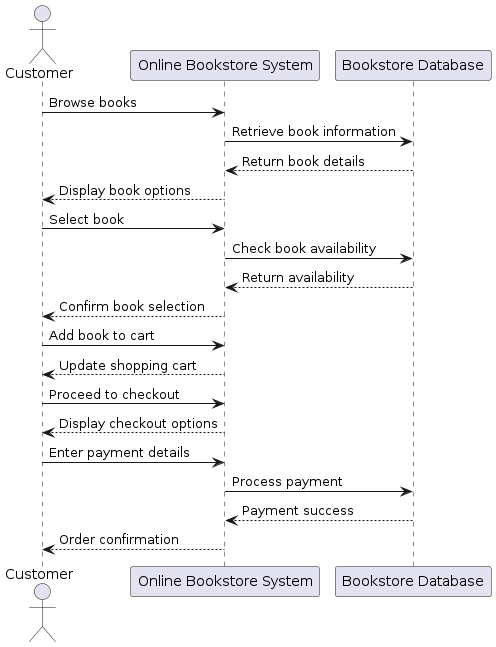
**3.3 NoSQL Diagram**

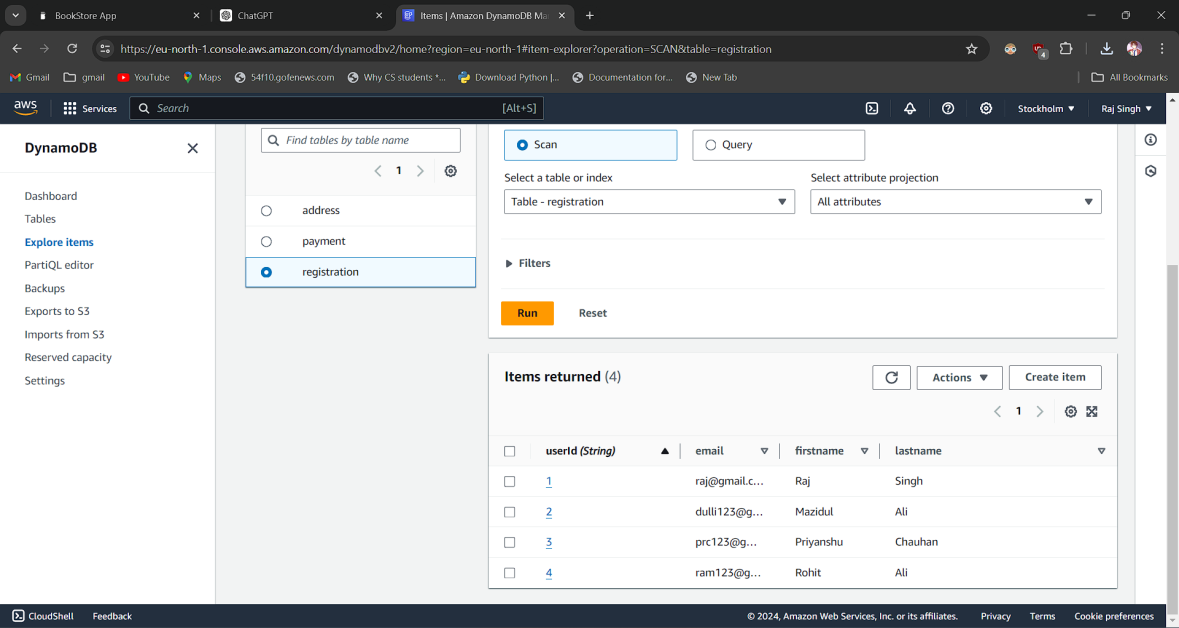
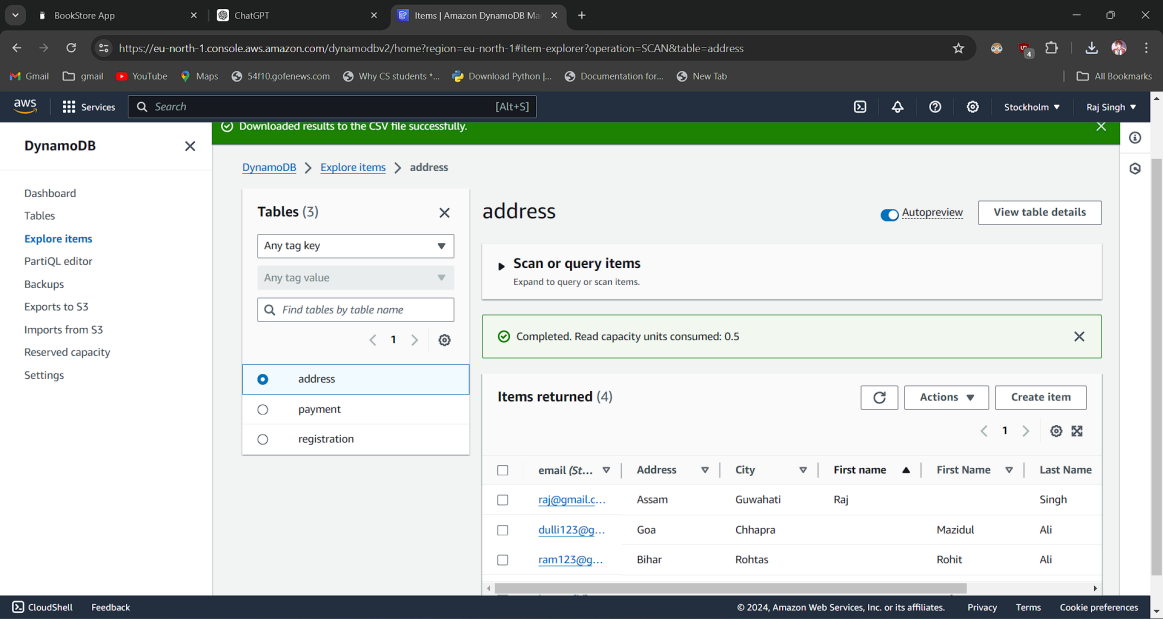
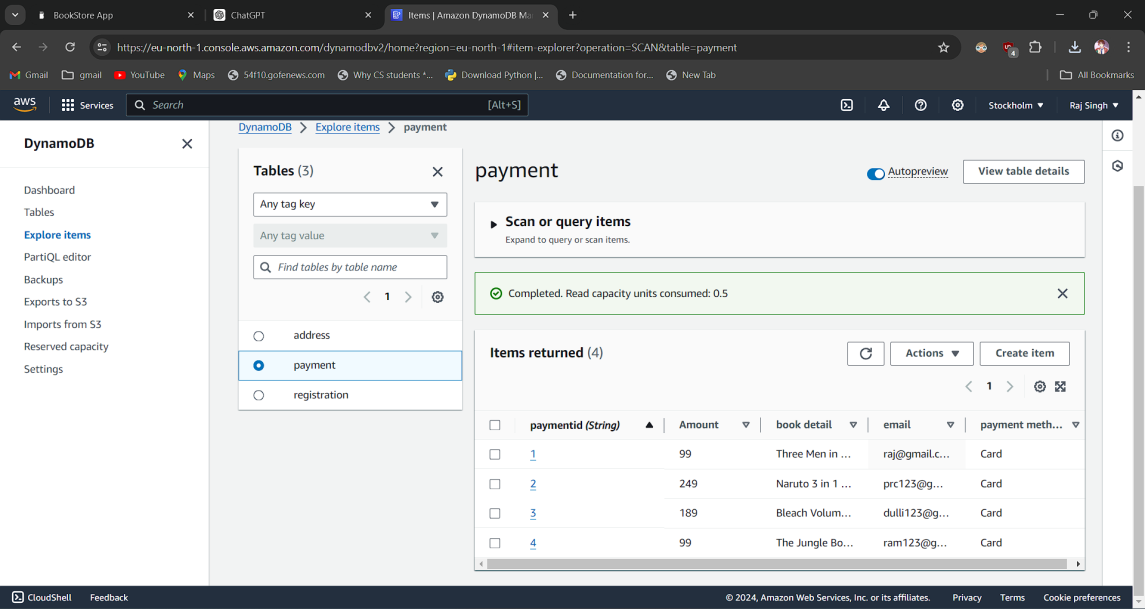


**3.4 Use case Diagram**



**3.5 Sequence Diagram**





**4. PROJECT IMPLEMENTATION**

**4.1 Description of the software used**

We have used React for the front-end development and AWS for the back-end and infrastructure:

i) Bookstore app(front-end):

* Developed using react, a popular javascript library for building user interfaces.
* It allows developers to create reusable UI components for building a dynamic and interactive user interfaces.
* Renders the bookstore's user interface in the web browser, allowing users to interact with the application to browse, search, and view books.

ii) Amazon DynamoDB:

* It’s a fully managed NoSQL database service provided by AWS.
* Used for storing and retrieving book data in a flexible, scalable and high performance manner.
* Integrated with the React application using AWS Amplify’s libraries for database operations.

iii) Amazon S3(Simple Storage Service):

* A scalable object storage service provided by AWS.
* Used for storing assets such as book images, cover photos or any other contents related to the books.

iv) AWS Amplify:

* A service provided by AWS for building and deploying cloud-based applications.
* Enables developers to quickly set up backend resources such as database, authentication services directly from the react applications.
* Facilitates seamless deployment and hosting of React applications on AWS infrastructure, ensuring scalability, reliability and security.

v) Commerce.js:

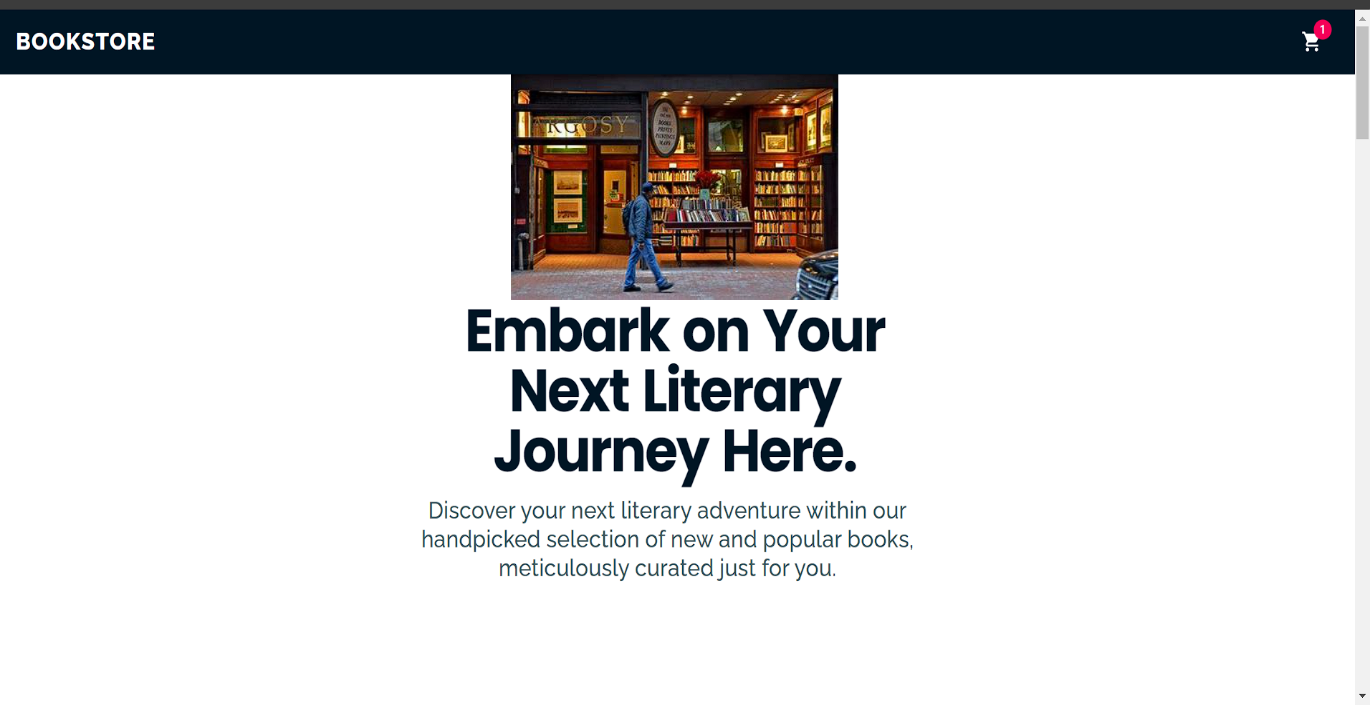
* It provides a set of APIs that allow developers to build web, mobile, augmented, and virtual shopping experiences.
* Operates as a headless eCommerce backend, meaning the frontend presentation is decoupled from the backend logic and processes.
* Integration with various payment gateways like Stripe, Omise, and Braintree is supported.

vi) Stripe:

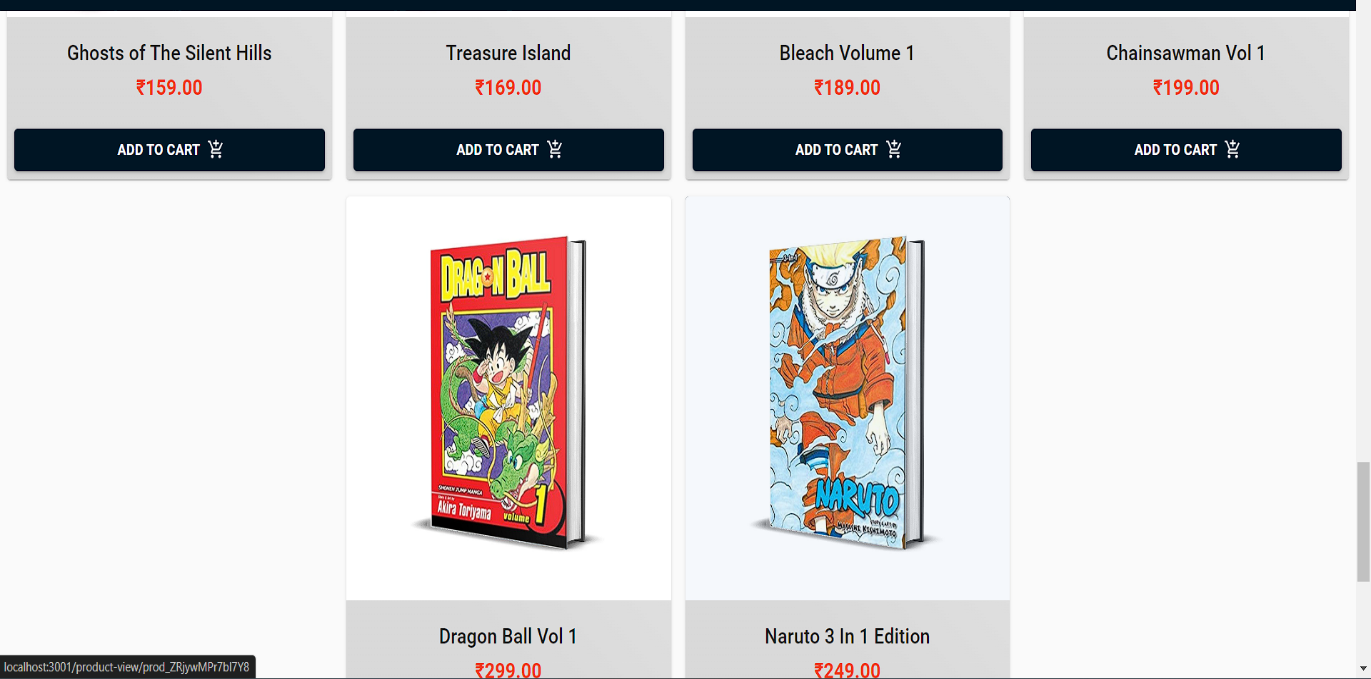
* Provides features such as card tokenization, fraud detection, and PCI compliance to ensure the security of payment data.
* Integrated with the Commerce.js SDK for handling secure payment transactions, including credit/debit card payments.
* A payment processing platform that allows businesses to accept online payments.

**4.2 Wireframes/ Ui**

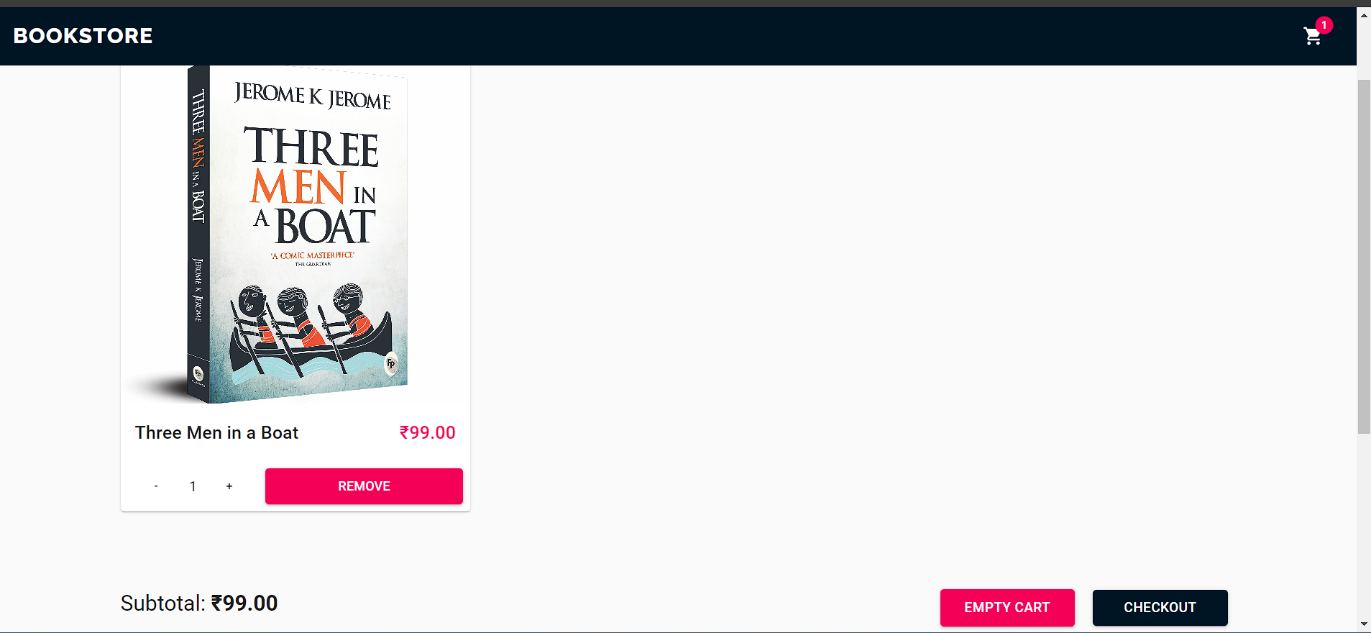
**Home Page**



**Books**

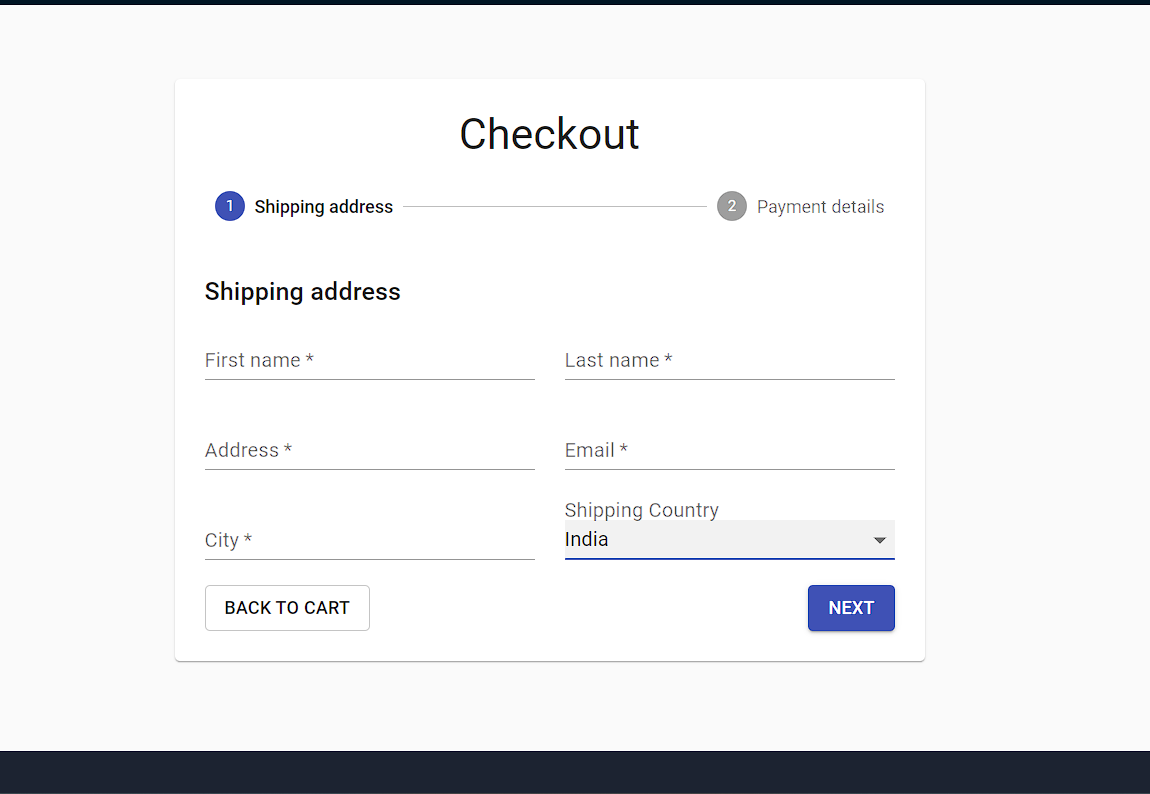


**Cart**

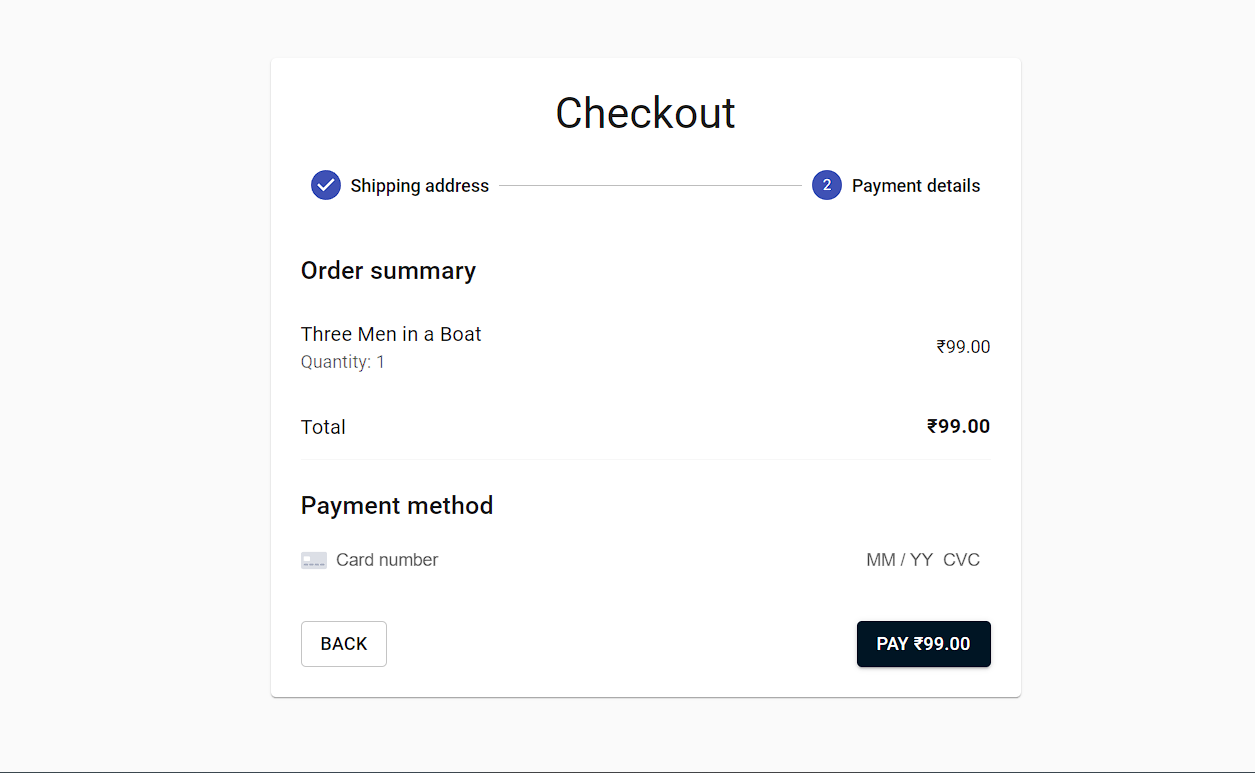


**Checkout**

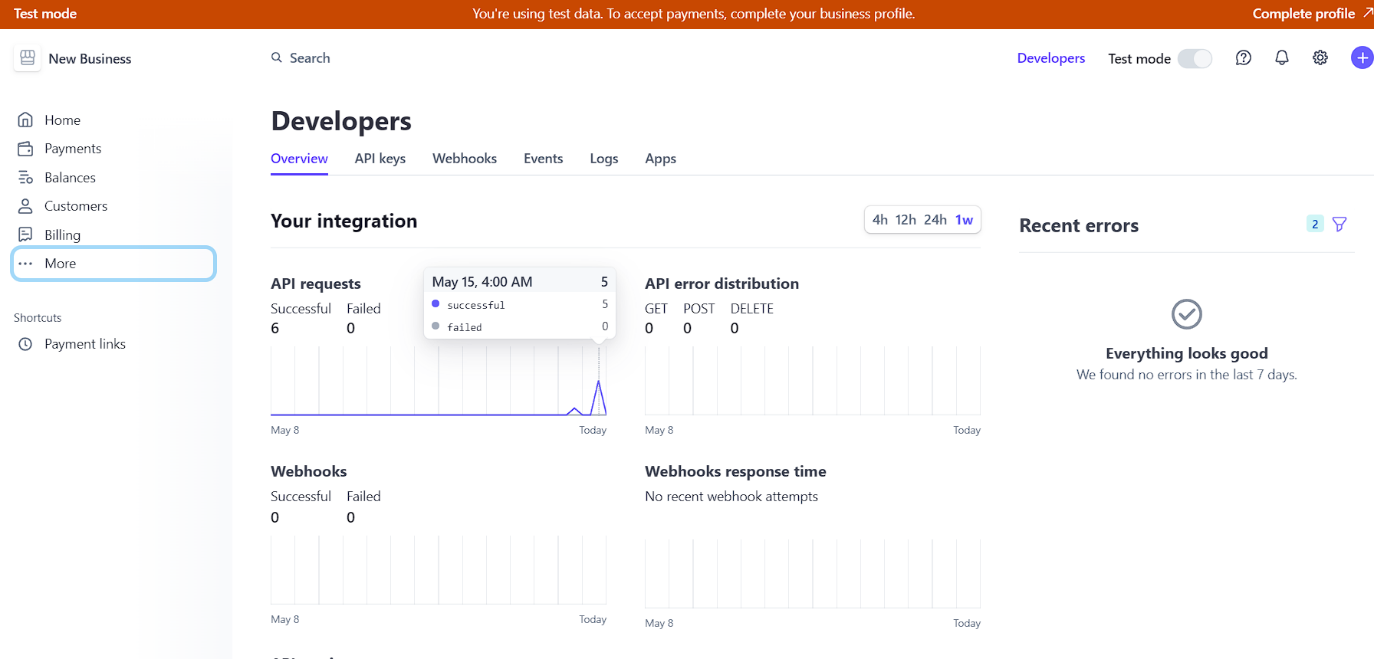
**(Shipping details)**



( **Payment)**



**Stripe**



**5. Testing / Result Analysis**

**5.1 Types of Testing**

Some common types of testing to ensure its reliability, functionality, security and performance are:

i) Unit testing:

* Test individual modules or components of the system to ensure that they function as intended.
* Ensures that each component behaves as expected and functions correctly.

ii) Integration testing:

* Ensures that integrations with external services, such as AWS services or Commerce.js APIs, function correctly.
* Tests the interaction between different components or modules of the application.

iii)End-to-End Testing:

* Tests the entire application flow from start to finish.
* Tests UI interactions, form submissions, navigation, and other user actions.

iv) Performance Testing:

* Tests the performance and scalability of the application under various conditions.
* Measures response times, throughput, and resource utilization to identify performance

**5.3 Test Cases**

i)Unit Test - React Component Rendering:

* Test Case: Verify that the homepage component renders without crashing.
* Expected Result: The homepage renders successfully, and all UI elements are displayed correctly.

ii)Unit Test - Commerce.js Integration:

* Test Case: Verify that product data is fetched from Commerce.js API.
* Expected Result: Product data, including titles, descriptions, and prices, is successfully retrieved from the Commerce.js API.

iii)Integration Test - Stripe Payment Processing:

* Test Case: Simulate a payment transaction using Stripe.
* Expected Result: The payment process is successful, and the user receives a confirmation of the order.

iv)End-to-End Test - User Registration and Login:

* Test Case: Simulate the user registration and login process.
* Expected Result: The user is able to register an account, login with the credentials, and access their account dashboard.

v)End-to-End Test - Add to Cart and Checkout:

* Test Case: Add a book to the cart and proceed to checkout.
* Expected Result: The selected book is added to the cart, and the user is able to complete the checkout process successfully.

vi)Performance Test - Page Load Time:

* Test Case: Measure the page load time of the homepage under various network conditions.
* Expected Result: The homepage loads within an acceptable time frame, ensuring a smooth user experience.

**6.1 Conclusion**

The serverless application built using AWS services offers several advantages, including scalability, cost-effectiveness, and ease of management. By leveraging AWS Lambda, API Gateway, DynamoDB, S3, and other services, the application architecture is highly resilient and can handle varying workloads efficiently. Additionally, integration with Commerce.js and Stripe enables e-commerce functionality, allowing users to browse, search, and purchase books securely.

The implementation of various types of testing ensures the reliability, functionality, security, and performance of the application. Unit tests, integration tests, end-to-end tests, and other testing methodologies help identify and address potential issues early in the development process, ensuring a robust and reliable application.

**6.2 References:**

<https://www.youtube.com/watch?v=nIrpSxMUAsk>

<https://commercejs.com/>

<https://dashboard.stripe.com/>

<https://www.plantuml.com/plantuml/uml/SyfFKj2rKt3CoKnELR1Io4ZDoSa70000>

<https://aws.amazon.com/lambda/>