

FIN-TECH E-COMMERCE APPLICATION

PRELIMINARY REPORT

15th September 2023

Introduction

In an era defined by technological innovation and digital transformation, I embark on a journey to craft a solution that promises to redefine the intersection of fin-tech and e-commerce. This preliminary report serves as a compass, guiding us through the early stages of my ambitious Project **FIN-TECH E-COMMERCE APPLICATION**.

The Vision Unveiled:

In an age where convenience, security, and accessibility are paramount, the convergence of financial technology(fin-tech) and e-commerce has emerged as a promising frontier. My Vision is to create dynamic and user-centric application that seamlessly integrates these two domains, offering users and unparalleled experience in managing their finances and engaging in online shopping.

The Path Ahead:

While the final destination remains on the horizon, this preliminary report provides a glimpse into our aspirations, goals, and the road map we envision to achieve them. While the project is still in its infancy, my commitment is resolute, and my enthusiasm unwavering.

Key Components:

- **Objective Clarity:** I begin by outlining the primary objectives that underscore the project's purpose and relevance.
- **Technological Landscape:** An overview of the technologies and tools at my disposal, including React Native, Django Rest Framework, and many more.
- **Testing and Quality Insurance:** By prioritizing testing and quality assurance, we ensure that our project remains robust and dependable throughout its development life-cycle
- **Project Plan:** A high-level plan that delineates my anticipated journey, though subject to evolution as the Project unfolds.
- **Personal Dynamics:** Introduction to my roles and responsibilities in breathing life into this vision.

- **Challenges and Risks:** An acknowledgment of the challenges I anticipate and my strategies to overcome them.
- **Anticipate Outcomes:** A glimpse into the future, where I share my expectations for the project's impact.

A Solo Odyssey:

This project is not merely a document; it is a testament to my commitment and unwavering resolve to navigate uncharted territory. As I set forth on this path, I invite you to follow along in this pursuit of innovation, discovery, and transformation.

Together, we will overcome challenges, and ultimately, realize the full potential of my project
FIN-TECH E-COMMERCE APPLICATION

Objective Clarity: Defining the Project's Purpose and Relevance

The foundation of any successful project lies in clear and well-defined objectives. In this section, we set out to elucidate the primary objectives that underpin the purpose and relevance of our project. FIN-TECH E-COMMERCE APPLICATION

1. Project Purpose:

The first step in achieving clarity of objectives is to establish the fundamental purpose of the project. This purpose serves as the guiding star, directing all project activities towards a common goal. In the context of our project:

Purpose Statement: Our project, "[Your Project Title]," aims to create a dynamic and user-centric application that seamlessly integrates financial technology (fin-tech) and e-commerce, thereby offering users an unparalleled experience in managing their finances and engaging in online shopping.

2. Key Objectives:

With the purpose articulated, we can now delve into the specific objectives that constitute the core of our project. These objectives are the tangible outcomes we aim to achieve. For "[Your Project Title]," the key objectives include:

Objective 1: Seamless Fin-tech Integration

Objective Statement: To seamlessly integrate fin-tech features into the application, enabling users to manage their finances, make secure payments, and engage in digital transactions effortlessly.

Objective 2: Enhanced User Experience

Objective Statement: To prioritize user experience through intuitive user interfaces, personalized shopping recommendations, and efficient transaction tracking.

Objective 3: Robust Security

Objective Statement: To ensure the highest level of security for user data and financial transactions by implementing industry-standard security protocols and encryption.

Objective 4: Cross-Platform Accessibility

Objective Statement: To make the application accessible across multiple platforms, ensuring users can enjoy a consistent experience on various devices.

Digital Banking

Revolut N26

Digital Payments

PayPal PayPal

Paxoneer

Consumer Finance

MONEY PATROL

intuit mint

Insurtech

BIMA

tröv

FINTECH ECOSYSTEM

Digital Investment

HEDGEABLE

ADDEPAR

Regtech

PassFort

clicks

Digital Lending

Kabbage

calyx

Institutions and organizations applying fin-tech in their business Fig(1)

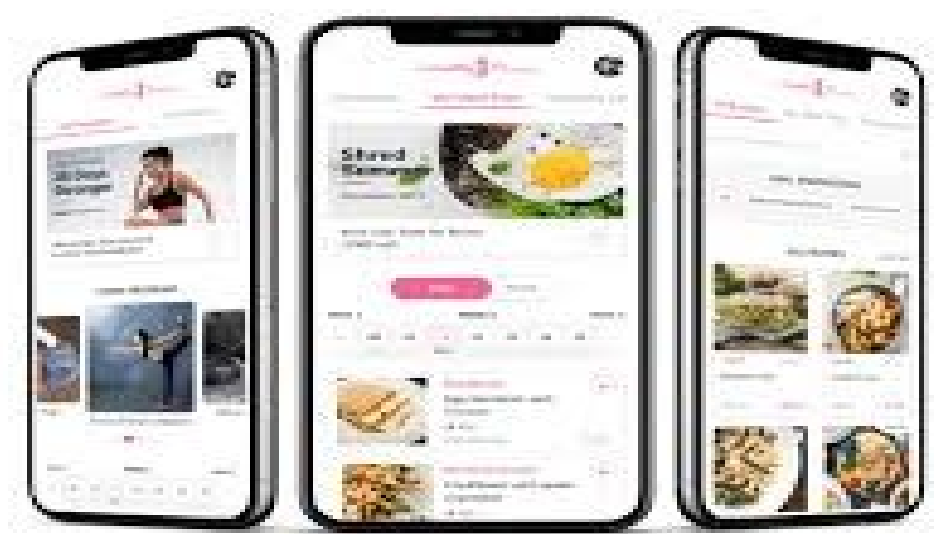
Technological Landscape: Leveraging Tools for Success

In the journey to create "[Your Project Title]," a careful selection of technologies and tools forms the cornerstone of our development strategy. This section provides an in-depth overview of the diverse array of tools and technologies that will empower the project's creation and execution.

1. React Native:

Technology Overview: React Native is an open-source JavaScript framework for building mobile applications. Leveraging its component-based architecture and the power of React, it allows for the development of cross-platform mobile applications with a single code-base (React Native,n.d.).

Role in the Project: React Native serves as the core technology for crafting the user interface (UI) and user experience (UX) of the mobile application. Its cross-platform capabilities enable us to target both iOS and Android platforms efficiently.

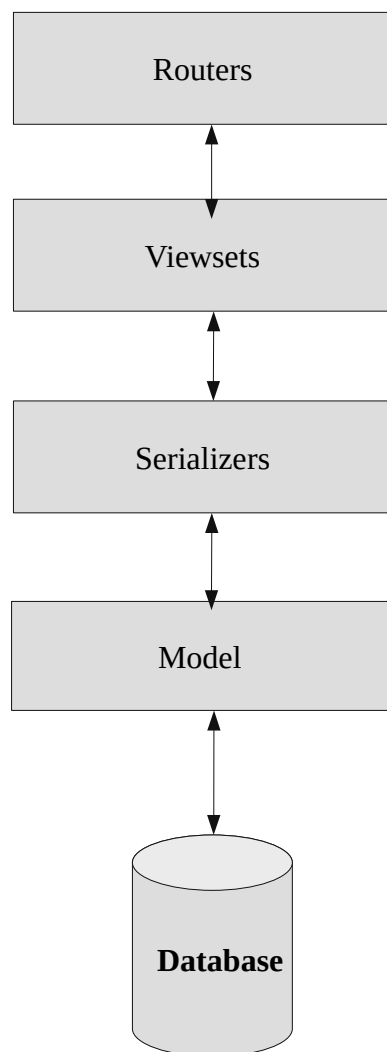


Mobile Application developed with React-Native Fig(2)

2. Django Rest Framework:

Technology Overview: Django Rest Framework is a robust and flexible toolkit for building Web APIs using the Django framework. It simplifies API development by providing tools for serialization, authentication, and viewsets (Django Rest Framework, n.d.).

Role in the Project: Django Rest Framework forms the backbone of our project's backend, enabling us to create secure and feature-rich APIs to interact with our application. It facilitates data exchange and ensures seamless communication between the frontend and backend components.



Django Rest framework used in creating a back-end API for our application Fig(3)

4. Integrated Development Environment (IDE):

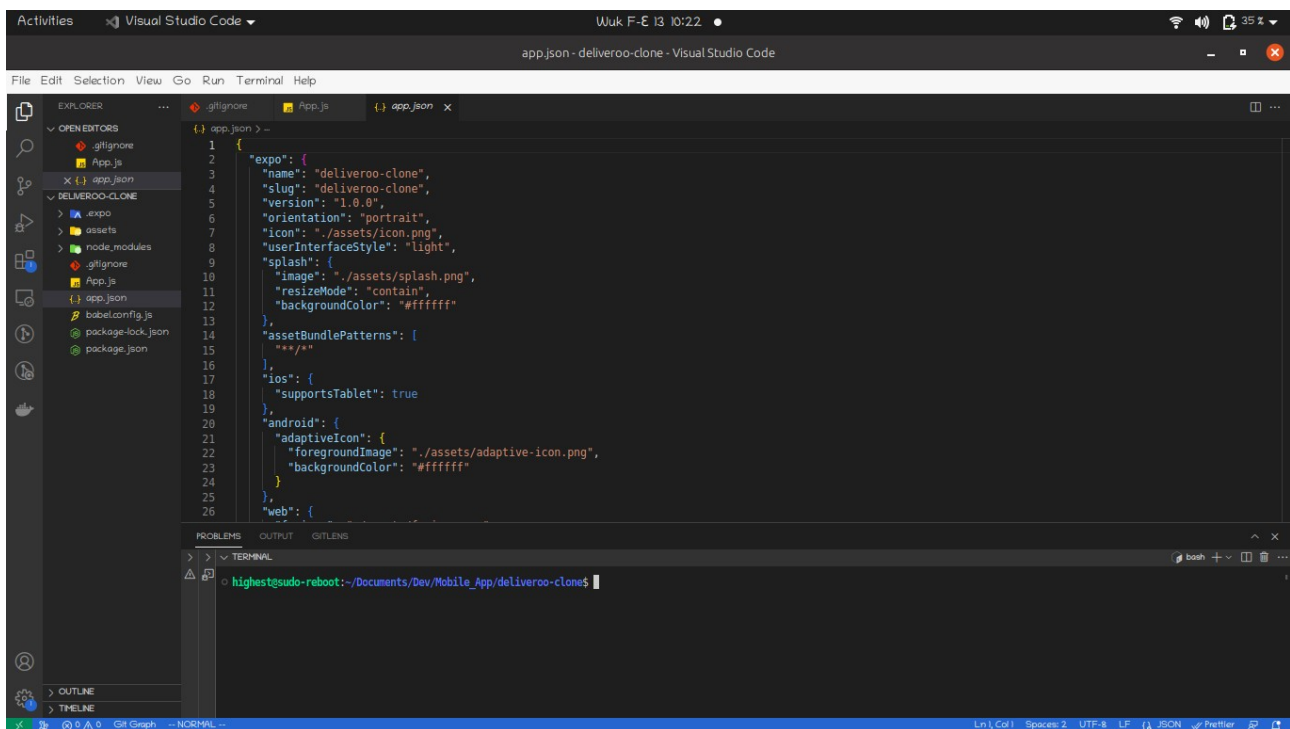
React Native (VS Code):

Technology Overview: Visual Studio Code (VS Code) is the selected IDE for React Native development. It is a lightweight and highly customizable code editor with extensive support for JavaScript and React development (Visual Studio Code, n.d.).

Role in the Project: VS Code plays a pivotal role in the React Native portion of our project. Its key contributions include:

Efficient React Native Support: VS Code provides an array of extensions and plugins tailored for React Native development. These tools streamline the coding process, offering features like auto-completion, debugging, and IntelliSense, which enhance development efficiency (Visual Studio Code, n.d.).

Debugging Capabilities: The integrated debugging features of VS Code simplify the identification and resolution of issues in the React Native codebase. Real-time debugging helps ensure code quality and stability.



Visual Studio Code for Creating mobile Application Fig(4)

Python Django (PyCharm):

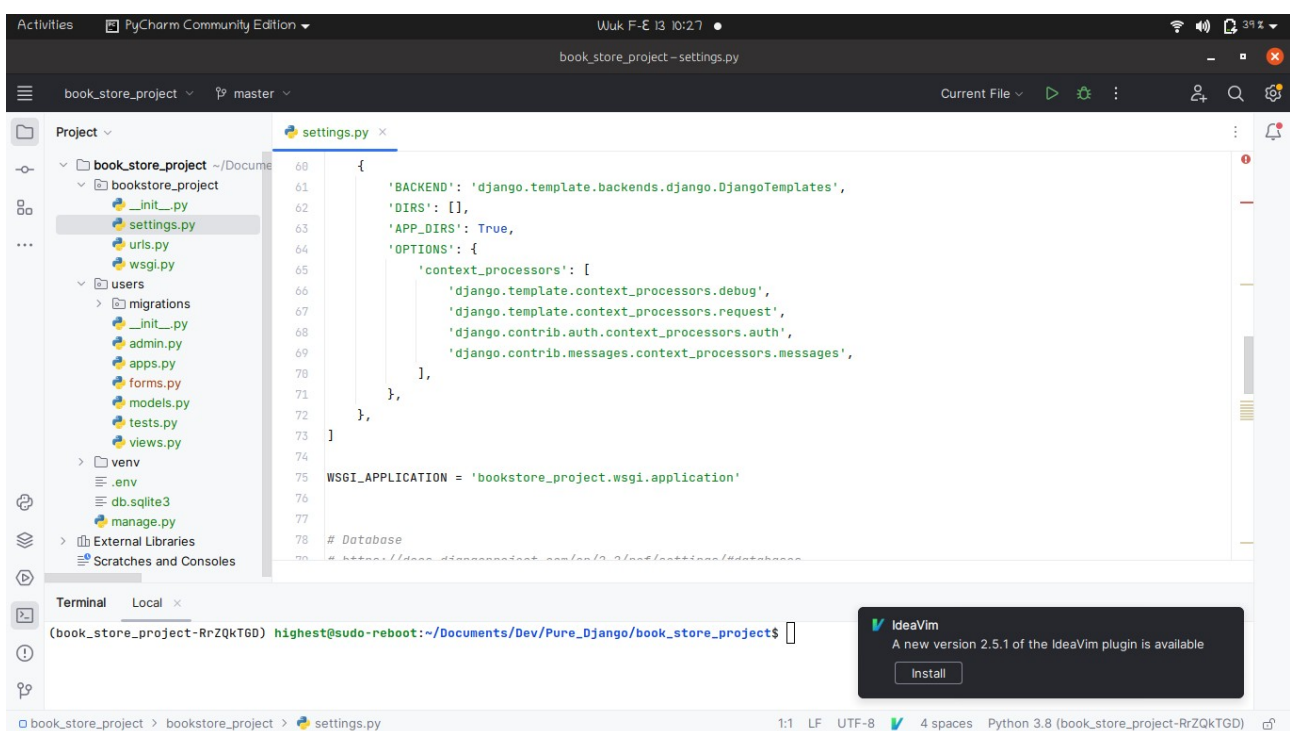
Technology Overview: PyCharm, a robust integrated development environment for Python, is selected for the Python Django component of the project. It offers a comprehensive set of tools for web development, including Django project support (Pycharm n.d.).

Role in the Project: PyCharm's role in our project is paramount for Django development:

Django Integration: PyCharm provides seamless integration with Django, offering templates, project structure management, and automated code generation. This significantly accelerates Django-based development tasks.

Pythonic Development: PyCharm promotes Pythonic coding practices, enhancing code readability and maintainability. It offers features like code inspection, intelligent code completion, and code navigation.

Database Management: The IDE facilitates database management and ORM (Object-Relational Mapping) support, simplifying the interaction with databases, which is crucial for Django-based applications.



Pycharm IDE being used for creating a backend API Fig(5)

In summary, the combination of Visual Studio Code (VS Code) for React Native and PyCharm for Python Django enhances the development process by providing specialized tools and features tailored to each technology stack. These IDEs contribute significantly to coding, debugging, and overall development efficiency throughout the project.

5. Version Control:

Technology Overview: GitHub

Technology Overview: GitHub is the selected version control system and hosting platform for our project. It is a web-based platform built around the Git version control system, designed to facilitate code collaboration, project management, and the tracking of code changes.

Role in the Project: GitHub serves as the backbone of our project's version control and collaboration efforts, offering a range of essential functionalities:

Code Repository: GitHub hosts our project's central code repository. This repository is the central hub where all project-related code, including React Native, Django, and associated scripts, is stored.

Version Control: Git, integrated with GitHub, provides precise version control capabilities. It allows us to track changes made to the code-base over time, providing a detailed history of commits, branches, and merges.

Collaboration: GitHub facilitates collaboration among project contributors, even in a solo development environment. It offers features like pull requests, which allow for the review and integration of code changes, ensuring code quality and consistency (Github, n.d.).

Issue Tracking: GitHub's issue tracking system enables us to manage and prioritize tasks, bugs, and enhancements. It provides a structured approach to addressing project-related concerns.

Branch Management: We utilize Git branches to work on different project features or bug fixes concurrently. GitHub simplifies branch management, making it easy to create, merge, or delete branches as needed.

Project Documentation: GitHub enables us to create and maintain project documentation, including README files and Wiki pages, to ensure that the project's details, setup instructions, and best practices are readily accessible (Github, n.d.).

Security and Access Control: GitHub offers robust security features, allowing us to control access to the project repository, implement authentication measures, and safeguard sensitive data.

GitHub's role in our project extends beyond version control; it fosters efficient collaboration, code quality, and project management. It ensures that our codebase remains organized, secure, and accessible throughout the development lifecycle, contributing significantly to the project's success.

Activities Google Chrome Wuk F-E 13 10:39

github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Repository template

No template

Start your repository with a template repository's contents.

Owner * Baah-Danso-Kenneth / **Repository name ***

Great repository names are short and memorable. Need inspiration? How about [psychic-octo-computing-machine](#) ?

Description (optional)

☐ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Github will be used to track, host our code base Fig(6)

6. Deployment and Hosting:

Technology Overview: Docker and AWS

Technology Overview: For the deployment and hosting of "[Your Project Title]," we have selected Docker and Amazon Web Services (AWS) as our preferred solutions. These technologies play a pivotal role in ensuring that our project is accessible, scalable, and performs optimally in a production environment.

Role in the Project: Docker and AWS contribute significantly to the project's success in several key aspects:

Containerization with Docker: Docker allows us to containerize our application components, encapsulating them into lightweight, portable containers. This containerization simplifies the deployment process, as each container includes all the dependencies required to run the application. It ensures consistency across various environments, from development to production (Docker,n.d.).

AWS Hosting: We leverage AWS as our hosting platform, taking advantage of its wide array of cloud services and infrastructure options. AWS provides scalable, reliable, and secure hosting solutions that align perfectly with our project's requirements.

Scalability: AWS offers elastic scaling capabilities, allowing us to adapt to changing user loads effortlessly. We can scale our resources up or down based on demand, ensuring that the application remains responsive and performance even during peak usage

Performance: AWS provides a high-performance infrastructure with options for load balancing, content delivery, and database services. This infrastructure enhances our application's responsiveness and reliability, delivering a seamless user experience.

Security: AWS offers robust security features, including Identity and Access Management (IAM), encryption, and monitoring tools. These features help us secure user data and maintain compliance with industry standards.

Accessibility: Our choice of Docker and AWS ensures that our application is accessible globally. AWS's worldwide network of data centers and content delivery services helps reduce latency and improve accessibility for users regardless of their geographic location.

In summary, the combined use of Docker for containerization and AWS for hosting provides a robust foundation for FIN-TECH E-COMMERCE APPLICATION. These technologies align with our project's objectives by offering scalability, performance, security, and accessibility. They empower us to deliver a dynamic and user-friendly fin-tech and e-commerce solution that meets the demands of modern users in a cloud-based environment.

The screenshot shows the AWS 'Deploy Docker Containers on Amazon ECS' how-to guide page. The page features a dark blue header with the AWS logo and navigation links. Below the header, there's a section titled 'Deploy Docker Containers on Amazon ECS' with the subtitle 'HOW-TO GUIDE'. The main content area is divided into two columns. The left column contains an 'Overview' section with a brief description of the tutorial. The right column contains a table with requirements for the tutorial.

Requirement	Details
✓ AWS experience	Beginner
⌚ Time to complete	10 minutes
\$ Cost to complete	Cost will vary by region, and will be around \$0.004 / hour of running the container
🔧 Requires	<ul style="list-style-type: none">• AWS account• Recommended browser: The latest version of Chrome or Firefox <p>[**]Accounts created within the past 24 hours might not yet have access to the...</p>

Hosting our project with docker on AWS Fig(7)

Testing and Quality Assurance:

1. Testing Tools:

React Native Testing:

Testing Framework: For React Native testing, we employ the popular testing framework Jest. Jest is well-suited for testing JavaScript and React Native applications. It provides robust testing utilities, including test runners, assertion libraries, and mocking capabilities.

Testing Library: To facilitate UI testing in React Native, we utilize tools like React Testing Library and Detox. React Testing Library simplifies component testing by focusing on user interactions and ensuring that our components work seamlessly in a real-world user environment. Detox, on the other hand, is dedicated to end-to-end testing, allowing us to simulate user interactions and validate application behavior.

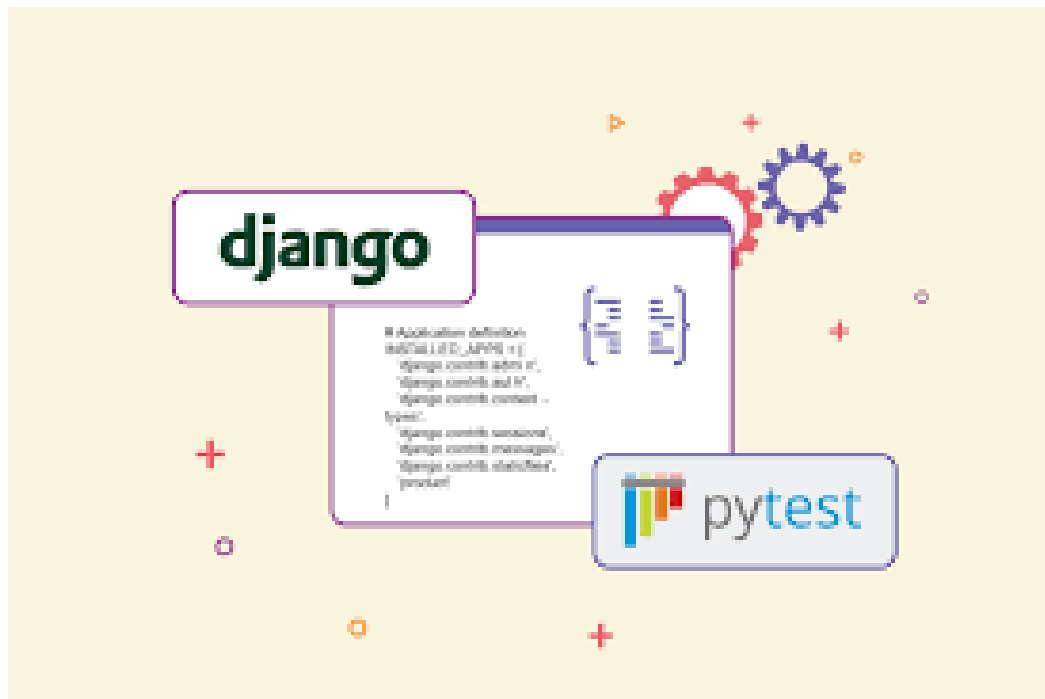


Unit testing with Jest on React-Native Application Fig(8)

Django Rest Framework Testing:

Testing Framework: Django Rest Framework (DRF) includes a built-in testing framework that is integral to our testing strategy. It offers features for testing views, serializers, authentication, and more (Django Rest Framework, n.d.).

PyTest: In addition to DRF's built-in testing capabilities, we leverage PyTest, a versatile Python testing framework. PyTest simplifies test case writing, parametrization, and reporting, enhancing the comprehensiveness and efficiency of our tests.



Pytest will be used in testing django back-end Fig(9)

Quality Assurance:

Quality Assurance Process: Our quality assurance process encompasses a comprehensive set of practices, including code reviews, continuous integration (CI), and automated testing. These practices ensure code quality and reliability throughout the development lifecycle.

Code Reviews: Code reviews are a fundamental part of our quality assurance process. All code changes, whether in React Native or Django Rest Framework, undergo peer review. This collaborative approach helps identify issues, enforce coding standards, and ensure code correctness.

Continuous Integration (CI): We implement CI pipelines using tools like Jenkins or Travis CI to automate build and test processes. CI ensures that code changes are tested automatically whenever new code is pushed, preventing regressions and maintaining project stability.

Automated Testing: Automated testing, as mentioned earlier, is a cornerstone of our quality assurance efforts. It includes unit testing, integration testing, and end-to-end testing, carried out using the tools and frameworks described above.

Project Plan: FIN-TECH E-COMMERCE APPLICATION

Timeline: September 18, 20223- January 18, 2024

Phase	Objectives	Activities
Phase1: September 18 – September 24	Define project scope, objectives, and requirements	<ul style="list-style-type: none">I. Research and finalize project requirementsII. Create a detailed project planIII. Set up development environment (VS Code for React Native, PyCharm for Django)IV. Establish GitHub repository for version control
Phase2: UI Creation (Weeks 2-4)	Design the user interface (UI) of the application.	<ul style="list-style-type: none">I. Create wire frames and mock-ups for UI elements.II. Develop UI components using React Native.III. Implement basic navigation and layout.
Phase3: Frontend Development(Weeks 5-8)	Build the frontend of the application using React Native	<ul style="list-style-type: none">I. Implement application logic and user interactions.II. Integrate APIs for data retrieval and display.III. Perform initial testing of the frontend components.
Phase 4: Backend Development (Weeks 9-12)	Develop the backend API using Django Rest Framework.	<ul style="list-style-type: none">I. Set up Django project and configure database.II. Create API endpoints for data retrieval and manipulation.III. Implement authentication and security measures.IV. Begin testing the API.
Phase 5: Integration and Testing (Weeks 13-15)	Integrate the frontend and backend components and perform comprehensive testing.	<ul style="list-style-type: none">I. Connect the React Native frontend to the Django backend.II. Conduct unit tests, integration tests, and end-to-end tests.III. Identify and resolve bugs and issues.
Phase 6: Refinement and Optimization (Weeks 16-18)	Optimize application performance and refine user experience.	<ul style="list-style-type: none">I. Refactor code for efficiency and maintainability.II. Optimize database queries and API responses.III. Enhance UI/UX based on user feedback (if available).
Phase 7: Final Testing and Debugging (Weeks 19-21)	Perform final testing and debugging to ensure a stable and	<ul style="list-style-type: none">I. Conduct rigorous testing, including load testing and

	error-free application.	performance testing. II. Address any remaining issues and optimize performance. III. Prepare for deployment.
Phase 8: Deployment and Hosting (Weeks 22-23)	Deploy the application on AWS using Docker for containerization.	I. Create Docker containers for the frontend and backend. II. Deploy containers on AWS, configure server settings. III. Ensure secure and scalable hosting.
Phase 9: Final Review and Documentation (Week 24)	Review the entire project, prepare documentation, and finalize the project.	I. Conduct a comprehensive review of the application. II. Create project documentation, including setup instructions and user guides. III. Prepare for the project's completion.

plan for execution of project Fig(10)

Challenges and Risks: Navigating the Path Ahead

In the pursuit of bringing FIN-TECH E-COMMERCE APPLICATION to life, it's crucial to recognize that challenges and risks are an integral part of any development journey. This section delves into the anticipated challenges and outlines the strategies devised to overcome them.

1. Technical Challenges:

Complex Integration: Integrating the React Native frontend with the Django backend can present technical challenges. Ensuring smooth data flow and synchronization between these components will be essential.

Performance Optimization: Achieving optimal performance for both frontend and backend components, especially in a resource-constrained environment, is a challenge that demands careful attention.

Strategies:

- Conduct rigorous testing and optimization throughout the development process.
- Leverage performance profiling tools to identify bottlenecks and areas for improvement.
- Seek guidance from the developer community and online resources to address integration challenges.

2. Resource Constraints:

Time Constraints: The solo development approach may present time limitations, making it essential to manage time efficiently to meet project milestones.

Scope Creep: Balancing project scope with available resources can be challenging. Expanding features beyond the project's original scope may lead to delays.

Strategies:

- Adhere to the project timeline and prioritize tasks based on their criticality.
- Maintain a clear project scope and manage stakeholder expectations.
- Regularly assess progress and adjust priorities as needed to stay on track.

3. Testing and Quality Assurance:

Testing Complexity: Rigorous testing, including UI, API, and end-to-end testing, can be time-consuming and complex.

Bugs and Issues: Identifying and addressing bugs and issues effectively to maintain code quality and user satisfaction is paramount.

Strategies:

- Implement a robust testing strategy with a focus on automation to streamline testing processes.
- Continuously monitor for issues and conduct thorough testing at each development phase.
- Allocate time for debugging and issue resolution in the project plan.

4. User Feedback and Iteration:

User Expectations: Meeting user expectations and adapting to user feedback can be challenging as user needs may evolve over time.

Strategies:

- Establish channels for gathering user feedback and iterate on the application based on user input.
- Prioritize user experience and user-centric design principles.

5. Security and Data Protection:

Data Security: Ensuring the security of user data, including sensitive information, is a critical aspect of the project.

Strategies:

- Implement robust authentication and authorization mechanisms.
- Regularly update security measures based on industry best practices and security advisories.

6. Scalability:

Scalability Demands: As the project gains users, the demand for scalability may increase. Ensuring the application can handle growth is a consideration.

Strategies:

- Design the application architecture with scalability in mind.
- Explore cloud-based hosting options to accommodate increased usage.

In conclusion, while challenges and risks are inherent in any development project, FIN-TECH E-COMMERCE APPLICATION is equipped with strategies to overcome them. By addressing technical complexities, managing resources efficiently, emphasizing testing and quality assurance, staying responsive to user feedback, prioritizing security, and planning for scalability, this project is poised to navigate challenges successfully and emerge as a robust fintech and e-commerce solution.

Anticipated Outcomes: Shaping the Future

The development and deployment of FIN-TECH E-COMMERCE APPLICATION are driven by a vision of creating a transformative solution at the intersection of fintech and e-commerce. These anticipated outcomes encapsulate the expectations for the project's impact on users, stakeholders, and the broader ecosystem.

1. Enhanced User Experience:

At the forefront of our expectations is the enhancement of the user experience for individuals engaging with our application. We anticipate:

Streamlined Financial Management: Users will experience a seamless integration of fintech tools, simplifying tasks such as budgeting, investing, and financial planning.

Effortless E-commerce: Users will enjoy a frictionless online shopping experience, featuring personalized recommendations, secure transactions, and convenient checkout processes.

2. Financial Empowerment:

FIN-TECH E-COMMERCE APPLICATION is expected to empower users by providing them with tools and insights to take control of their finances:

Financial Literacy: We aim to foster financial literacy and empower users with knowledge to make informed financial decisions.

Investment Opportunities: Our platform will enable users to explore and access investment opportunities, potentially leading to improved financial stability and growth.

3. Accessibility and Inclusivity:

Our project is committed to ensuring accessibility and inclusivity for a wide range of users:

Accessibility: The application will be designed to be accessible to users with diverse needs, including those with disabilities, ensuring that everyone can benefit from its services.

Inclusivity: The platform will be inclusive, catering to users from different demographic backgrounds and financial situations.

4. Industry Impact:

We anticipate that FIN-TECH E-COMMERCE APPLICATION will have a broader impact on the fintech and e-commerce industries:

Innovation Catalyst: Our project aims to inspire innovation in these sectors, prompting other stakeholders to explore new possibilities.

Market Influence: As the application gains traction, it has the potential to influence industry trends and best practices.

5. Stakeholder Satisfaction:

The satisfaction of stakeholders, including users, investors, and team members, is of paramount importance:

User Satisfaction: We anticipate high user satisfaction rates, driven by a user-centric design, reliability, and continuous improvement.

Investor Confidence: We aim to attract investor confidence and support for the project's growth and sustainability.

6. Community Building:

FIN-TECH E-COMMERCE APPLICATION is not just a product; it's a community-building endeavor:

User Community: We envision a vibrant user community that shares insights, feedback, and experiences, fostering a sense of belonging.

Industry Collaboration: Collaborations with industry partners and experts are anticipated to enhance the project's capabilities and reach.

In summary, the anticipated outcomes of FIN-TECH E-COMMERCE APPLICATION extend far beyond the boundaries of a software application. They encompass the transformation of user experiences, financial empowerment, industry impact, stakeholder satisfaction, inclusivity, and community building. These expectations serve as a driving force, propelling the project toward its ultimate goal of reshaping the landscape of fintech and e-commerce.

Conclusion: Pioneering a New Frontier

As we conclude this preliminary report, we find ourselves at the threshold of a transformative journey—a journey that holds the promise of reshaping the way we engage with fintech and e-commerce. "[Your Project Title]" is more than just a project; it is a vision that seeks to pioneer a new frontier where financial empowerment and e-commerce convenience converge.

Our exploration into the realms of React Native and Django Rest Framework, our commitment to user-centric design, and our unwavering dedication to excellence have brought us to this pivotal moment. While the road ahead is lined with challenges and uncertainties, it is also adorned with opportunities for innovation and growth.

In the coming months, we will breathe life into this vision. We will transcend technical complexities, overcome resource constraints, and navigate the ever-evolving landscape of user expectations. We will embark on a journey of continuous learning, adaptation, and collaboration.

But FIN-TECH E-COMMERCE APPLICATION is not just about technology; it's about people—people seeking better financial management, seamless online shopping experiences, and a future where inclusivity and accessibility are paramount. It's about stakeholders who believe in the potential of this project to influence industries and change lives.

Together, we stand at the precipice of innovation, driven by a shared commitment to this vision. As the sole developer, my role is clear—to steer this ship through uncharted waters, to adapt to the winds of change, and to lead us toward the realization of our shared aspirations.

In the pages of this preliminary report, we've outlined our objectives, detailed our technology stack, and anticipated challenges and outcomes. We've glimpsed into the future we envision—a future where FIN-TECH E-COMMERCE APPLICATION plays a pivotal role in shaping the landscape of fintech and e-commerce.

This is not the end; it is merely the beginning. We invite you to join us on this remarkable journey of exploration, innovation, and transformation. Together, we will navigate the complexities of technology and human experience, ultimately giving life to the vision we hold dear.

As we set forth on this path, let us remember that it is not just about reaching the destination; it is about the journey itself—the challenges we overcome, the innovations we foster, and the impact we create along the way.

In unity and determination, we shall pioneer this new frontier. The future awaits, and we are ready to shape it.

Thank you for being a part of this vision.

With unwavering resolve,

BAAH DANSO KENNETH

References:

Online Resources:

React Native. (n.d.). Retrieved from <https://reactnative.dev/>

Django Documentation. (n.d.). Retrieved from <https://docs.djangoproject.com/en/stable/>

Jest. (n.d.). Retrieved from <https://jestjs.io/>

React Testing Library. (n.d.). Retrieved from <https://testing-library.com/docs/react-testing-library/intro/>

Detox. (n.d.). Retrieved from <https://github.com/wix/Detox>

Django Rest Framework. (n.d.). Retrieved from <https://www.django-rest-framework.org/>

Visual Studio Code. (n.d.). Retrieved from <https://code.visualstudio.com/>

PyCharm. (n.d.). Retrieved from <https://www.jetbrains.com/pycharm/>

Git. (n.d.). Retrieved from <https://git-scm.com/>

GitHub. (n.d.). Retrieved from <https://github.com/>

Docker. (n.d.). Retrieved from <https://www.docker.com/>

Amazon Web Services (AWS). (n.d.). Retrieved from <https://aws.amazon.com/>

Online Images:

Fig2: Found via Google Images with keywords "React-Native testing"

Fig9: Found via Google Images with keywords "Django-testing"

Fig7: Found via Google Images with keywords "AWS Docker hosting"

Local Images (Author's Own Work):

Fig6: Screenshot of GitHub page

Fig5: Screenshot of PyCharm IDE

Fig4: Screenshot of Visual Studio Code IDE

Fig3: Diagram of Django Rest-framework scope