Explore a retail revolution with Shopkeeper—an intuitive app redefining stock and point-of-sale dynamics. With a user-friendly interface, real-time reporting, and secure Firebase integration, it adapts seamlessly to various businesses. Shopkeeper isn't just an app; it's your gateway to streamlined inventory control and an enhanced retail experience. Embrace efficiency and engagement with Shopkeeper.

SHOP KEEPER INVENTORY APP.

Muhammad Areeb Dayo

REPORT.

TITLE: SHOPKEEPER INVENTORY & POS APP.

Course: MOBILE APPLICATION DEVELOPMENT

Faculty: DANIYAL NAWAZ **Slot: MONDAY 6:30-9:30**

GROUP MEMBERS & PREFERRED ROLES.

S #	Name	Reg No.	Sap Id	Preferred Role	Assigned Role
1	Muhammad Awais	49537	29277	Webside development,SQA, Project leader had in FYP	Project Manager
2	Hasnain sher	49535	29048	Website developer,SQA,	App Frontend
3	Adil Ali	49130	26743	Website developer, WordPress developer with Php	App Backend
4	Jahangir Ghulam Hyder	51524	31152	Exp: html.css.javascript as a fronted.	App Frontend
5	MUHAMMAD AREEB	53184	28343	Enhancing skill	Business Analyst
6	Hannan	46852	20191	Enhancing skill	App Frontend
7	Amar Chand	27890	57890	Enhancing skill	App Backend
8	Asna Ansari	52791	27952	Enhancing skill	App Backend
9	Updesh Muni	53428	29732	Fresh	Tester
10	Hasnain Shabir	51114	32118	No experience	UI/UX Designer

TABLE OF CONTENTS.

ABSTRACT.

The Shopkeeper Inventory and POS App is a versatile and user-friendly mobile solution crafted to redefine inventory control and streamline point-of-sale operations for businesses. Integrating critical functionalities such as CRUD operations, reporting, billing, and seamless data export through Firebase, the app delivers a comprehensive approach to efficient stock management.

Users can effortlessly add, delete, update, and view items, ensuring real-time accuracy in inventory tracking. The reporting feature empowers users to generate customizable reports, facilitating data-driven decision-making based on stock levels and sales trends.

The robust billing system by using barcode scanner simplifies transaction management, while data export to Firebase guarantees secure and accessible storage. Leveraging Firebase APIs for authentication, real-time database management, storage, and analytics, the Shopkeeper Inventory and POS App stands as a powerful and forward-thinking tool for businesses to optimize their stock operations and enhance the overall retail experience.

INTRODUCTION.

Introducing the Shopkeeper Inventory and POS App—an innovative mobile solution designed to revolutionize stock management and retail operations. With a user-friendly interface, seamless CRUD operations, real-time reporting, and an advanced barcode system, the app ensures precise and efficient inventory control. The robust billing system simplifies transactions, while Firebase integration guarantees secure data storage and accessibility. Leveraging Firebase APIs for authentication and analytics, Shopkeeper stands as a powerful tool, offering businesses a dynamic and intuitive platform to optimize stock operations and enhance the overall retail experience.

Beyond its core features, Shopkeeper adapts to diverse business models, providing customization options for retail stores, cafes, boutiques, and more. The app's versatility is complemented by a real-time analytics dashboard, empowering business owners with insights into performance metrics, sales trends, and inventory turnover. Its integration of machine learning capabilities further enhances forecasting accuracy, anticipating stock demands based on historical data.

Shopkeeper's advanced barcode system facilitates streamlined inventory tracking and faster transactions. By leveraging barcode technology, businesses can enhance accuracy in stock management and expedite the point-of-sale process. Shopkeeper is more than just an inventory management tool; it's a transformative solution, redefining how businesses interact with their customers. Its commitment to user-centric design, data security, and cutting-edge features makes it an indispensable asset for businesses aiming to revolutionize their retail ecosystem. Embrace Shopkeeper and explore a new era in inventory management and customer interaction.

ACKNOWLEDGEMENT.

We would like to express our heartfelt gratitude to all those who have contributed to the development and realization of this project. First and foremost, we extend our sincere thanks to "**sir Daniyal Nawaz"**, our mentor and guide, for providing invaluable insights, guidance, and unwavering support throughout the journey.

We are grateful to the entire our MPD project Team for their collaboration, encouragement, and collective effort in bringing this project to fruition. Special thanks to "**sir Daniyal Nawaz"** for their dedicated contributions.

We also want to acknowledge the support of our families and friends who stood by us during the challenges and milestones of this project. Their encouragement and understanding were pivotal in keeping us motivated.

Last but not least, we extend our appreciation to all those whose work, research, or insights have influenced and shaped this project. Your contributions have been instrumental in its development.

Thank you all for being part of this journey and for making this project a reality.

EXECUTIVE PROJECT SUMMARY.

The Shopkeeper Inventory and POS App is a revolutionary mobile solution designed to enhance stock management and retail operations. With a focus on user-friendly design and seamless integration of features such as CRUD operations, real-time reporting, secure billing, and Firebase functionalities, the app aims to provide businesses with a comprehensive platform for efficient inventory control and an elevated retail experience. The project's key goals include prioritizing a user-centric interface, offering versatility for diverse business models, and ensuring scalability to adapt to evolving industry needs. In summary, the Shopkeeper app aspires to set a new standard in modern retail applications by combining advanced features with innovative technology for optimal stock management and customer engagement.

LITERATURE REVIEW.

Extensive research in retail underscores the critical role of efficient inventory management and point-of-sale (POS) systems. Studies often delve into models like just-in-time (JIT) and ABC analysis to optimize supply chain dynamics and minimize costs. Integrating POS systems with inventory management is crucial for real-time updates on sales, stock levels, and transaction data. User-centric design principles, focusing on user experience and interface design, impact employee productivity and customer satisfaction. Emerging technologies, including cloud-based solutions and mobile applications, are researched for their role in enhancing flexibility and scalability. Data security and compliance with industry regulations are emphasized to ensure secure transactions and customer privacy. Analytics and business intelligence tools integrated with inventory and POS systems provide actionable insights into consumer behavior, pricing strategies, and decision-making processes. Efficient inventory practices contribute to streamlined supply chain operations, reducing lead times and overall operational costs. With the rise of mobile and omni-channel retailing, ongoing research explores strategies for adapting inventory and POS systems to multi-channel sales, addressing challenges in synchronizing inventory across various channels for a unified customer experience. The dynamic retail landscape continues to drive research for innovative technologies and strategies, ensuring the continuous optimization of inventory management and POS systems.

Moreover, the literature highlights the significance of technology adoption in retail, with a focus on the transition from traditional systems to cloud-based solutions. Researchers delve into the benefits and challenges associated with this shift, emphasizing the need for retail businesses to embrace scalable and cost-effective solutions. The mobile and omni-channel retailing era has prompted investigations into strategies for synchronizing inventory and POS systems across various channels seamlessly. The integration of data analytics not only offers insights into consumer behavior but also facilitates predictive modeling for inventory optimization. As retail continues to evolve, ongoing research explores innovative technologies and methodologies, aiming to enhance the adaptability, efficiency, and overall performance of inventory management and POS systems.

OBJECTIVES.

The objective of the Shopkeeper Inventory and POS App project is to develop and implement a cutting-edge mobile solution that revolutionizes the traditional approach to stock management and retail operations. The primary goals are:

1) **Efficiency in Stock Management**:
   Intuitive platform that enables businesses to efficiently manage their stock through seamless CRUD operations, ensuring real-time accuracy in inventory tracking.

2) **Enhanced Retail Experience:**
   Elevate the overall retail experience by providing a user-friendly interface, incorporating features such as real-time reporting, and implementing a secure billing system to simplify transactions.

3) **Integration of Firebase Technology:**
   Leverage Firebase APIs for authentication, real-time database management, storage, and analytics to ensure secure data storage and accessibility, setting the foundation for a robust and scalable application.

4) **Versatility and Customization:**
   Tailor the app to meet the diverse needs of various businesses by offering customization options, making it adaptable to different industry models and business requirements.

5) **Scalability:**
   Design the app with scalability in mind, allowing it to evolve with changing business demands and industry trends.

6) **User-Centric Design:**
   Prioritize a user-centric design to make the app accessible across various technical expertise levels, enhancing usability and user satisfaction.

7) **Setting a Standard:**
   Aspire to set a new standard in modern retail applications by combining advanced functionalities with innovative technology, creating a comprehensive solution that optimizes stock operations and customer engagement.

8) **Barcode Implementation:**
   The primary goal of incorporating barcode technology is to enhance operational efficiency by enabling rapid and accurate item identification, thereby streamlining both inventory management and the point-of-sale process.

FUNCTIONAL FEATURES.

1) Effortless CRUD Operations:

Seamlessly add, delete, update, and view items for real-time accuracy in inventory tracking.

2) Real-time Reporting:

Customizable reports to empower data-driven decision-making based on stock levels and sales trends.

3) Secure Billing System:

Simplify transaction management through an intuitive and robust billing system, enhancing the point-of-sale experience.

4) Firebase Integration:

Ensure secure data storage and accessibility by leveraging Firebase for authentication, real-time database management, storage, and analytics.

5) User-Friendly Interface:

Prioritize a user-centric design to enhance accessibility across various technical expertise levels, ensuring an intuitive and seamless user experience.

6) Versatility for Business Models:

Customize the app to meet the diverse needs of different businesses, providing adaptable solutions for retail stores, cafes, boutiques, and more.

7) Scalability:

Design the app to be adaptable and scalable, allowing it to evolve with changing business requirements and industry trends.

8) Machine Learning Integration:

Utilize machine learning capabilities for predictive analytics, anticipating stock demands based on historical data to optimize inventory levels.

9) Multi-Platform Accessibility:

Enable accessibility across various platforms, including smartphones and tablets, to facilitate on-the-go stock management and transactions.

10) Real-time Analytics Dashboard:

Incorporate a dynamic analytics dashboard to offer business owners insights into performance metrics, sales trends, and inventory turnover in real-time.

11) Setting Standards in Retail Technology:

Aspire to set a new standard in modern retail applications by combining advanced features with innovative technology, providing a comprehensive solution for optimized stock operations and customer engagement.

12) Multi-Platform Accessibility:

Enable access across various devices, including smartphones and tablets.

13) Efficient Barcode Scanning.

- Swift identification of items by scanning barcodes.
- Streamlined checkout process, reducing errors and saving time.

14) Enhanced Inventory Accuracy.

- Quick addition and editing of items with barcode scanning.
- Versatile across various product types, ensuring compatibility.

15) User-Friendly Interface.

- Integrates seamlessly with product details.
- Customizable settings for a personalized experience.

16) Real-time Synchronization.

- Instant updates across devices for efficient stock management.
- Accessible on smartphones and tablets for on-the-go operations.

17) Data Security Assurance.

- Securely stores scanned barcode information.
- Maintains confidentiality and integrity of sensitive data.

NON-FUNCTIONAL FEATURES:

1) Security:

Ensure data security through Firebase integration and industry-standard encryption protocols.

2) Performance:

Optimize app performance for quick response times and efficient data processing.

3) Reliability:

Build a reliable system that minimizes downtime and ensures continuous availability.

4) Usability:

Prioritize a user-friendly design to enhance ease of use and accessibility.

5) Scalability:

Design the app architecture to handle increased data and user load as the business grows.

6) Interoperability:

Ensure compatibility with other systems and technologies commonly used in retail environments.

7) Compliance:

Adhere to industry standards and regulations governing data protection and retail operations.

8) Maintainability:

Facilitate easy maintenance and updates to ensure the app's longevity and adaptability.

9) Documentation:

Provide comprehensive documentation for developers, users, and administrators.

10) Performance Monitoring:

Implement monitoring tools to track and analyze app performance, ensuring optimal functionality.

SYSTEM ARCHITECTURE.

The Shopkeeper Inventory and POS App's system architecture is designed to deliver a robust and scalable solution for efficient stock management and retail operations. The architecture comprises several key components, each playing a vital role in ensuring seamless functionality.

1) Client-Side Application:

The client-side application serves as the user interface, accessible through various platforms such as smartphones and tablets. It interacts with the server-side components to request and display data, manage user input, and facilitate a smooth user experience.

2) Server-Side Application:

The server-side application is responsible for processing client requests, executing business logic, and managing the interaction with the database. It ensures the real-time execution of CRUD operations, report generation, and billing processes.

3) Firebase Integration:

Firebase is integrated into the architecture to handle authentication, real-time database management, storage, and analytics. It ensures secure data storage, user authentication, and facilitates real-time updates to keep the application synchronized.

4) Database:

The database component stores and manages the app's data, including stock information, transaction records, and user details. It is designed to support efficient data retrieval and storage, contributing to the real-time functionality of the application.

5) Machine Learning Module:

The machine learning module is integrated to provide predictive analytics for optimizing inventory levels. Leveraging historical data, it anticipates stock demands and assists in maintaining an optimal balance.

6) Reporting Engine:

The reporting engine component handles the generation of customizable reports. It extracts relevant data from the database, processes it based on user preferences, and delivers comprehensive reports for data-driven decision-making.

7) Authentication Services:

Authentication services ensure secure access to the application, validating user credentials and authorizing actions based on user roles and permissions. Firebase Authentication is utilized to enhance security.

8) Analytics Dashboard:

The analytics dashboard component provides real-time insights into performance metrics, sales trends, and inventory turnover. It enhances business intelligence and aids in strategic decision-making.

9) Communication Middleware:

The communication middleware facilitates seamless communication between different components, ensuring data flows efficiently across the system. It contributes to the real-time synchronization of data.

10) External Integrations:

External integrations enable compatibility with other systems and technologies commonly used in retail environments. This ensures interoperability and a cohesive retail ecosystem.

11) Front-End:

The front-end of the Shopkeeper app serves as the user interface, providing a seamless and intuitive experience for users. Key elements include:

- **User Interface (UI):** The UI is designed for accessibility and ease of use, featuring a clean layout, intuitive navigation, and visually appealing design elements.

- **Client-Side Application:** The client-side application is responsible for rendering the UI on various devices, including smartphones and tablets. It facilitates user interactions, manages input, and communicates with the back-end to request and display data.
- **Reporting Interface:** The reporting interface allows users to generate customizable reports, providing valuable insights into stock levels, sales trends, and other relevant metrics.
- **Billing System UI:** The billing system UI handles transaction-related activities, allowing users to create invoices, manage receipts, and conduct seamless point-of-sale transactions.
- **Authentication Interface:** The authentication interface enables users to securely log in, protecting the application and ensuring that only authorized personnel can access sensitive information.

12) Back-End:

The back-end of the Shopkeeper app consists of server-side components that handle data processing, business logic, and interactions with databases and external services

- **Server-Side Application:** The server-side application processes client requests, executes business logic, and manages the flow of data between the front-end and the database.
- **Database Management System (DBMS):** The DBMS stores and manages the application's data, including information about stock items, transactions, and user details. It supports efficient data retrieval and storage.
- **Firebase Integration:** Firebase integration enhances the back-end by providing services such as authentication, real-time database management, storage, and analytics. It ensures secure data storage and facilitates real-time updates.
- **Machine Learning Module:** The machine learning module, part of the back-end, processes historical data to provide predictive analytics. It aids in optimizing inventory levels by anticipating stock demands.
- **Authentication Services:** Authentication services verify user credentials, ensuring secure access to the application. It controls user permissions and roles, enhancing the overall security of the system.
- **Communication Middleware:** The communication middleware facilitates smooth communication between different components of the back-end, ensuring efficient data flow and synchronization.
- **External Integrations:** External integrations enable compatibility with other systems and technologies commonly used in retail environments. This ensures interoperability and a cohesive retail ecosystem.

DEVELOPMENT PROCESS.

The development of the Shopkeeper Inventory and POS App follows a systematic and iterative process, incorporating best practices to ensure the creation of a robust and user-friendly solution. The development lifecycle can be outlined as follows:

1) **Requirements Analysis:**

   Gather and analyze key features and business objectives.

2) **Design Phase:**

   Architect the system, design UI, and define the database schema.

3) **Front-End Development:**

   Build an intuitive and responsive user interface.

4) **Back-End Development:**

   Develop server-side logic, integrate databases, and implement Firebase services.

5) **Database Implementation:**

   Set up and optimize the database for efficient data handling.

6) **Machine Learning Integration:**

   Integrate machine learning for predictive analytics on inventory levels.

7) **Testing:**

   Conduct thorough testing, addressing identified issues.

8) **Deployment:**

   Deploy to staging and then to the production environment.

9) **Monitoring and Optimization:**

   Implement monitoring tools and optimize performance based on insights.

10) **Documentation:**

    Create comprehensive documentation for users and administrators.

11) **Training and Support:**

    Conduct training sessions and establish a support system.

12) **Continuous Improvement:**

    Gather feedback and implement regular updates for ongoing improvement.

CHALLENGES AND SOLUTIONS.

1) **Integration Complexity:**
   - **Challenge:** Integrating various components like Firebase, machine learning, and external services can be complex.
   - **Solution:** Adopt a modular approach, ensuring clear interfaces and thorough testing during integration phases.

2) **Security Concerns:**
   - **Challenge:** Safeguarding sensitive data, especially during transactions and storage.
   - **Solution:** Implement robust encryption, secure authentication, and regular security audits to identify and address vulnerabilities.

3) **User Interface Optimization:**
   - **Challenge:** Balancing a feature-rich UI with simplicity for diverse users.
   - **Solution:** Conduct usability testing, gather user feedback, and iterate on the design for an intuitive interface.

4) **Scalability Demands:**
   - **Challenge:** Meeting the scalability requirements as the user base grows.
   - **Solution:** Design the architecture with scalability in mind, employ cloud services, and regularly assess performance metrics for optimization.

5) **Machine Learning Accuracy:**
   - **Challenge:** Ensuring the accuracy of machine learning predictions for optimal inventory management.
   - **Solution:** Train the model with quality data, fine-tune algorithms, and continuously update the model based on real-world performance.

   \
6) **User Adoption:**
   - **Challenge:** Ensuring user adoption and providing adequate training.
   - **Solution:** Offer comprehensive training sessions, create user-friendly documentation, and provide ongoing support channels.

7) **Data Migration Challenges:**
   - **Challenge:** Migrating existing data to the new system without disruptions.
   - **Solution:** Develop a robust data migration plan, conduct thorough testing, and have backup mechanisms in place.

8) **Performance Optimization:**
   - **Challenge:** Optimizing app performance for responsiveness and minimal downtime.
   - **Solution:** Implement monitoring tools, conduct performance tests, and continuously optimize based on feedback.

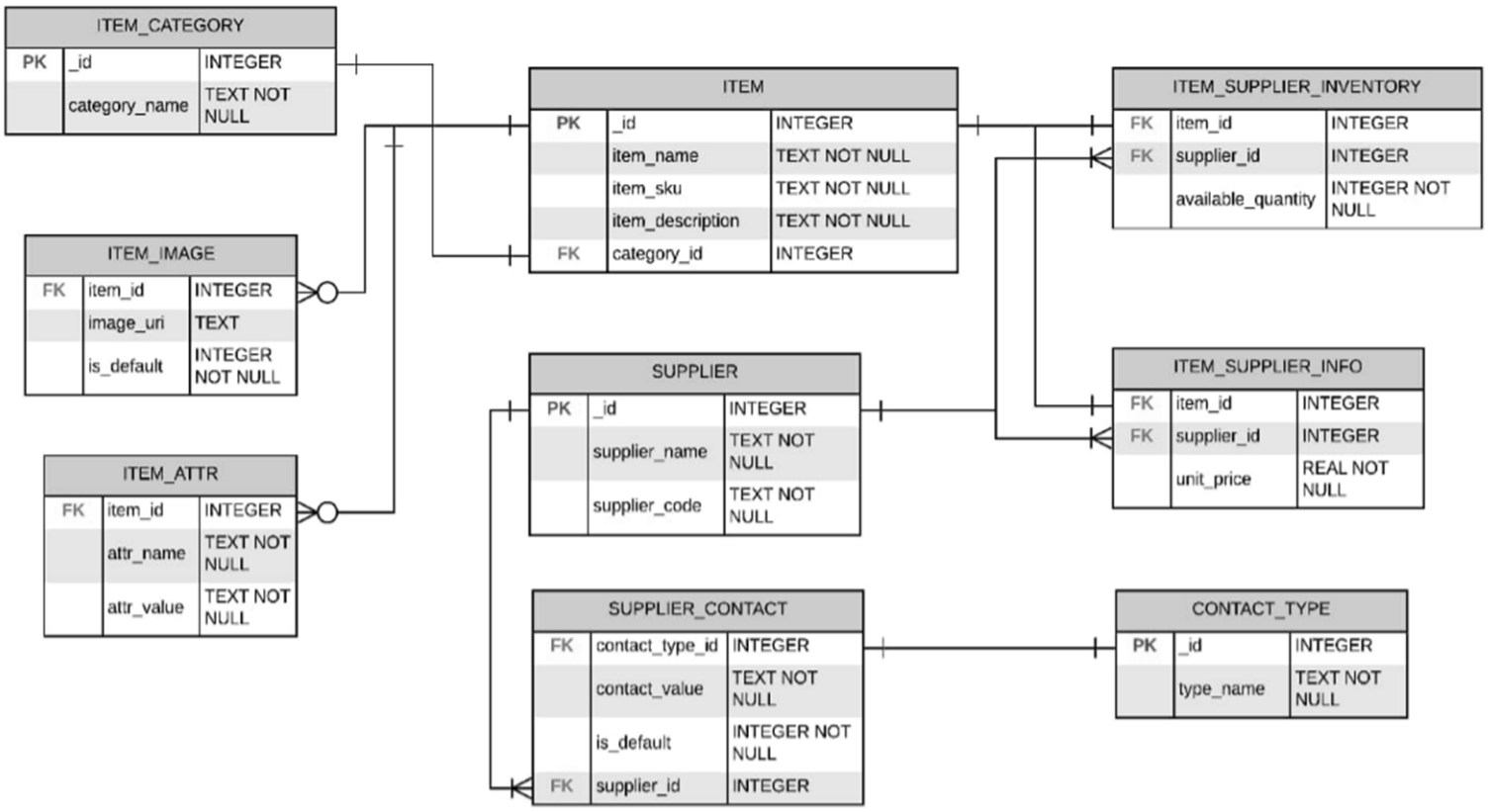
9) **Regulatory Compliance:**
   - **Challenge:** Adhering to industry regulations for data protection and retail operations.
   - **Solution:** Stay informed about relevant regulations, implement compliance measures, and conduct regular audits.

10) **Continuous Improvement:**
   - **Challenge:** Maintaining the app's relevance over time.
   - **Solution:** Gather user feedback, analyze performance metrics, and implement regular updates to add features and address emerging challenges.

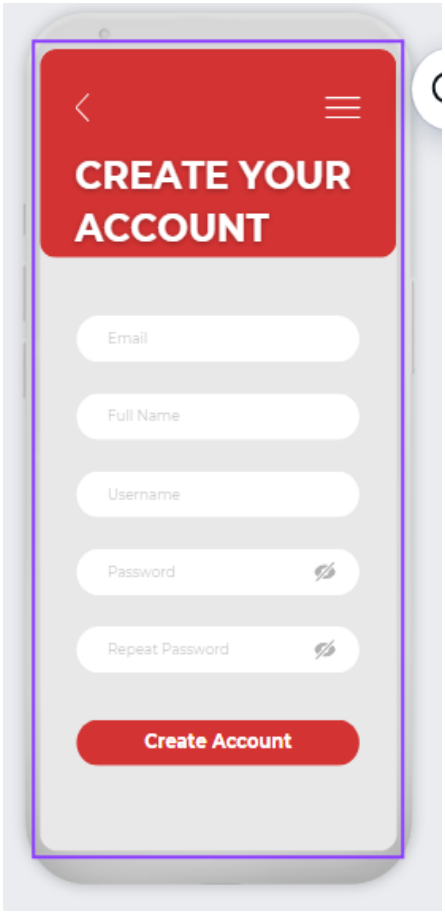
CONCLUSION.

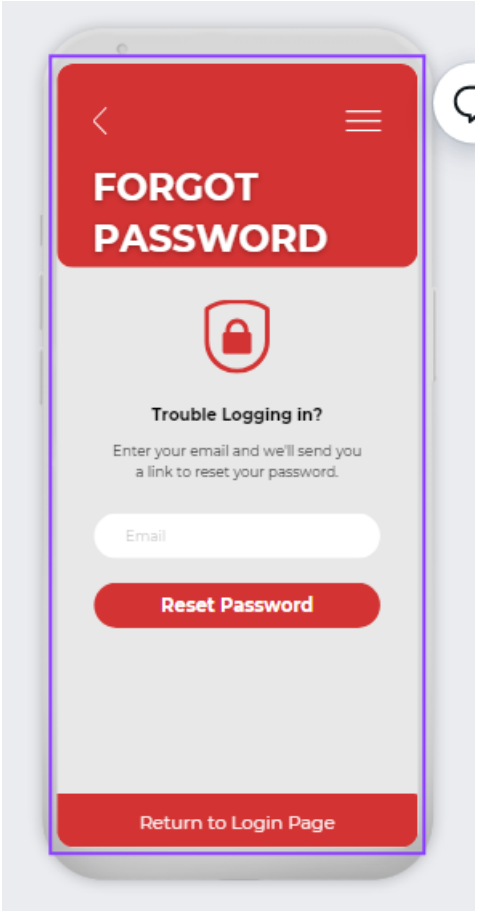
In conclusion, the Shopkeeper Inventory and POS App represents a transformative solution in the realm of retail technology. With a focus on efficient inventory management, seamless point-of-sale operations, and integration with advanced features such as real-time reporting and billing, the app addresses critical needs in the retail sector. The user-centric design ensures accessibility and ease of use, promoting a positive experience for both employees and customers. Leveraging Firebase for secure data storage, authentication, and real-time database management adds a layer of reliability to the application. The incorporation of machine learning for predictive analytics further distinguishes the app, providing businesses with insights into stock demands and optimizing inventory levels. The emphasis on scalability, versatility, and multi-platform accessibility positions the app as a dynamic tool adaptable to various business models and retail environments. As the Shopkeeper Inventory and POS App integrates seamlessly with emerging technologies and addresses the evolving challenges in the retail landscape, it stands as a testament to the continuous pursuit of innovation in inventory management and point-of-sale systems. Moving forward, the app holds the promise of setting new standards in user-friendly, adaptable, and efficient retail applications, contributing to the evolution of modern retail technology.

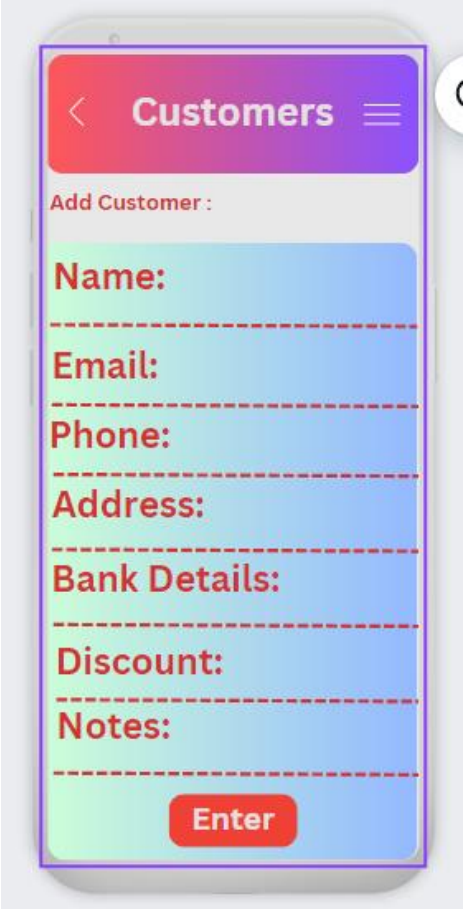
ER DIAGRAM.


ITEM_CATEGORY
PK _id INTEGER
category_name TEXT NOT NULL
ITEM
PK _id INTEGER
item_name TEXT NOT NULL
item_sku TEXT NOT NULL
item_description TEXT NOT NULL
FK category_id INTEGER
ITEM_SUPPLIER_INVENTORY
FK item_id INTEGER
FK supplier_id INTEGER
available_quantity INTEGER NOT NULL
ITEM_IMAGE
FK item_id INTEGER
image_uri TEXT
is_default INTEGER NOT NULL
SUPPLIER
PK _id INTEGER
supplier_name TEXT NOT NULL
supplier_code TEXT NOT NULL
ITEM_SUPPLIER_INFO
FK item_id INTEGER
FK supplier_id INTEGER
unit_price REAL NOT NULL
ITEM_ATTR
FK item_id INTEGER
attr_name TEXT NOT NULL
attr_value TEXT NOT NULL
SUPPLIER_CONTACT
FK contact_type_id INTEGER
contact_value TEXT NOT NULL
is_default INTEGER NOT NULL
FK supplier_id INTEGER
CONTACT_TYPE
PK _id INTEGER
type_name TEXT NOT NULL

UI/UX DESIGNS.



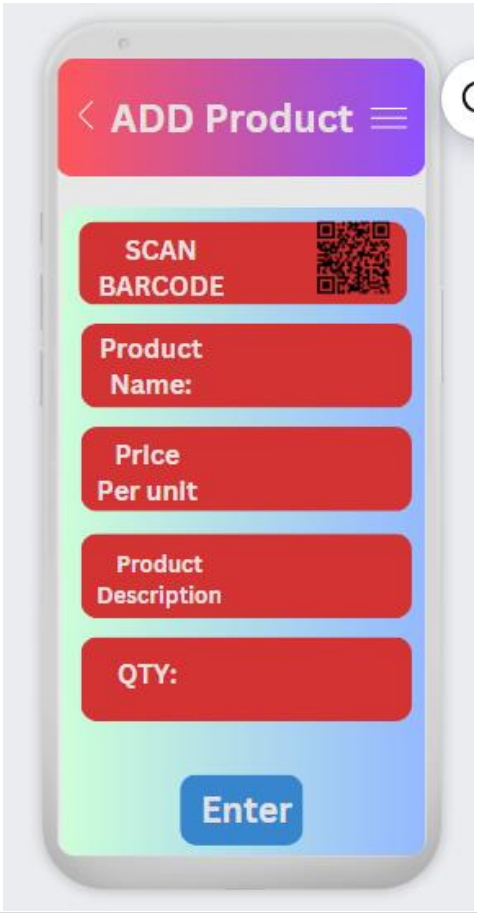


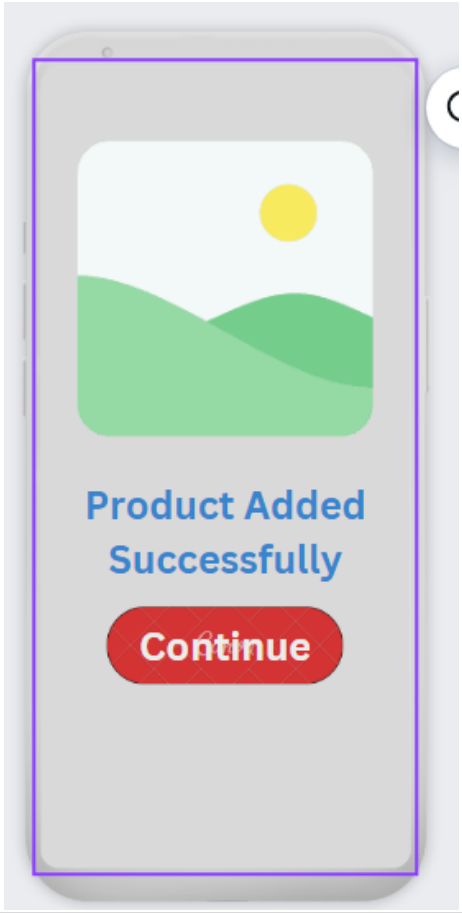


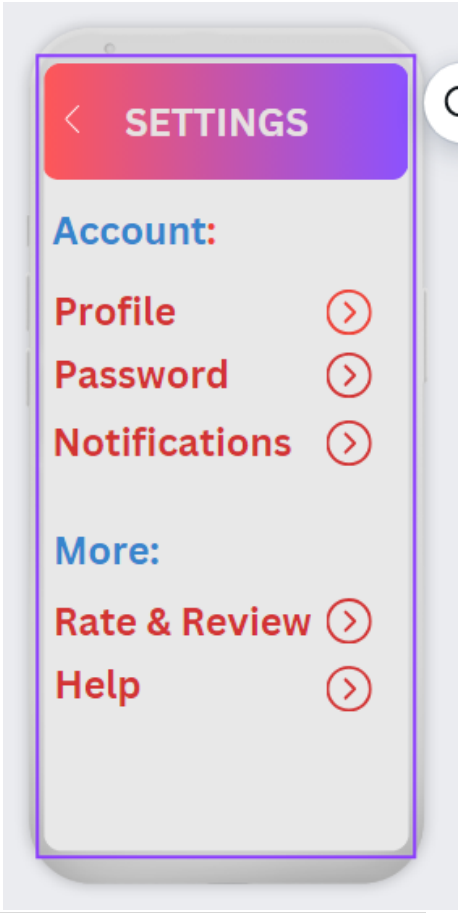

EXPANSES
Books QTY: 20
Dice QTY: 20
Pen QTY: 20
Water QTY: 20
Other QTY: 20
Edit
Add


Performance
Books QTY: 20
Dice QTY: 20
Pen QTY: 20
Water QTY: 20
Other QTY: 20
Edit
Add


BILLING
Outgoing:
Prdt Name:
QTY:
Remaining:
Approved:
Click Here


ADD Product
SCAN BARCODE
Product Name:
Price Per unit
Product Description
QTY:
Enter


Product Added Successfully
Continue


SETTINGS
Account:
Profile
Password
Notifications
More:
Rate & Review
Help

TEST CASES.

TEST CASE: LOG IN Case No: 1.

Test Case ID	01
Test Case	LOG IN
Description	Verify that users can successfully log in to the POS & Inventory App.
Preconditions	1. The POS & Inventory App is installed and accessible. 2. A valid user account with the provided email and password exists in the system.
Steps	1. Open the POS & Inventory App. 2. Enter the email into the "Email" field. 3. Enter the password into the "Password" field. 4. Click on the "Login" button.
Expected Results	i. The login process should be successful without errors. ii. The user should be redirected to the main dashboard or landing page of the POS & Inventory App.
Test Results	Pass /Fail
Post conditions	i. The user is successfully logged in. ii. The main dashboard or landing page of the POS & Inventory App is displayed.

TEST CASE: Create Your Account Case No: 2.

Test Case ID	02
Test Case	Create Your Account
Description	Verify that users can successfully create a new account.
Preconditions	1. The account creation page is accessible. 2. The system is in a state where it can accept new account registrations.
Steps	1. Open the "Create Your Account" page. 2. Enter a valid email address into the "Email" field. 3. Enter the full name into the "Full Name" field. 4. Enter a unique username into the "Username" field. 5. Enter a secure password into the "Password" field. 6. Repeat the password in the "Repeat Password" field. 7. Click on the "Create Account" button.
Expected Results	1. The account creation process should be successful without errors. 2. The user should receive a confirmation message indicating that the account has been created.
Test Results	Pass /Fail
Post conditions	1. The user account is successfully created. 2. The user may receive a confirmation email if email verification is part of the process. 3. The user can now log in with the created credentials.

TEST CASE: **Forgot Password** **Case No: 3.**

Test Case ID	03
Test Case	Forgot Password
Description	Verify that users can successfully reset their password using the "Forgot Password" feature.
Preconditions	i. The "Forgot Password" page is accessible. ii. The user has a registered email in the system.
Steps	1. Open the "Forgot Password" page. 2. Enter the registered email address into the "Email" field. 3. Click on the "Reset Password" button.
Expected Results	1. The system should process the request without errors. 2. The user should receive a password reset email containing a link to reset the password.
Test Results	Pass/Fail
Post conditions	1. The user receives an email with a password reset link. 2. The link should redirect the user to a page where they can enter a new password. 3. After setting a new password, the user should be informed that the password has been successfully reset.

TEST CASE: Add New Item **Case No: 4.**

Test Case ID	04
Test Case	Add New Item
Description	Verify that a new item can be successfully added to the inventory.
Preconditions	1. The user is logged into the system. 2. The "ITEMS" module is accessible.
Steps	1. Navigate to the "ITEMS" module. 2. Click on the "Add Item" button. 3. Fill in the required information for the new item: 4. Item Name: "Test Item" 5. Quantity: 10 6. Price: Rs 100.0 (or any desired price) 7. Other relevant details 8. Click on the "Save" button.
Expected Results	1. The system should save the new item without errors. 2. The item "Test Item" with the specified details should be added to the inventory.
Test Results	Pass/Fail
Post conditions	1. The item "Test Item" is displayed in the list of items with the provided details. 2. The overall stock quantity and value are updated to reflect the addition of the new item.

Case No: 5.

TEST CASE: ITEMS	
Test Case Id	5
Test Case	Add New Item
Description	Verify that a new item can be successfully added to the system.
Preconditions	The user is logged into the system. The Items module is accessible.
Steps	Navigate to the "Items" module. Click on the "Add Item" button. Fill in the required information for the new item: Name: "Test Item" Quantity: 10 Category: Any category of your choice Other relevant details Click on the "Save" button.
Expected Results	The system should save the new item without errors. The item "Test Item" with a quantity of 10 should be added to the list.
Test Results	Pass/Fail
Post Conditions	The item "Test Item" is displayed in the list of items with the specified quantity. The overall count of items is updated to reflect the addition of the new item.

Case No: 6.

TEST CASE: CUSTOMERS	
Test Case Id	6
Test Case	Add New Customer
Description	Verify that a new customer can be successfully added to the system.
Preconditions	The user is logged into the system. The Customers module is accessible.
Steps	Navigate to the "Customers" module. Click on the "Add Customer" button. Fill in the required customer information: Name: "Test Customer" Email: "test.customer@example.com" Phone: "123-456-7890" Address: "123 Main Street, City, Country" Bank Details: Any relevant bank information Discount: 10% (or any desired discount value) Notes: "Test customer for verification purposes" Click on the "Save" button.
Expected Results	The system should save the new customer without errors. The customer "Test Customer" with the provided details should be added to the customer list.
Test Results	Pass/Fail
Post Conditions	The customer "Test Customer" is displayed in the list of customers with the specified details. The overall count of customers is updated to reflect the addition of the new customer.

Case No: 7.

TEST CASE: EXPANSES	
Test Case Id	7
Test Case	Add Expanses
Description	Verify that a new expense can be successfully added to the system.
Preconditions	The user is logged into the system. The Expenses module is accessible.
Steps	Navigate to the "Expenses" module. Click on the "Add Expense" button. Fill in the required information for the new expense: Category: "Books" Quantity: 20 (or any desired quantity) Other relevant details: Any additional information related to the expense. Click on the "Save" button.
Expected Results	The system should save the new expense without errors. The expense entry for "Books" with the specified quantity should be added to the list of expenses.
Test Results	Pass/Fail
Post Conditions	The expense entry for "Books" is displayed in the list of expenses with the provided details. The overall count of expenses is updated to reflect the addition of the new expense.

Case No: 8.

TEST CASE: PERFORMANCE	
Test Case Id	8
Test Case	Performance - Add/Edit Items
Description	Evaluate the system's performance when adding or editing items in large quantities.
Preconditions	The user is logged into the system. The Performance module for adding/editing items is accessible.
Steps	Navigate to the "Performance" module. Click on the "Add" or "Edit" button to initiate the process. For adding items: Add a large number of items (e.g., 100, 500, or more). Observe the time it takes for the system to process and save the items. For editing items: Select a category with a large quantity (e.g., 100). Edit the quantity to a new value (e.g., 150). Observe the time it takes for the system to process and save the edits.
Expected Results	The system should handle the addition or editing of a large number of items without significant delays. The processing time for adding or editing items should be within acceptable performance limits.
Test Results	Pass/Fail
Post Conditions	The added/edited items are successfully saved in the system. The overall performance of the system remains stable.

TEST CASE: BILLING Case No: 9.

Test Case ID	09
Test Case	BILLING
Description	Verify that the shopkeeper mobile app effectively records outgoing products in the billing system.
Preconditions	3. the shopkeeper is logged into the mobile application. 4. Confirm a stable network connection.
Steps	5. Launch the mobile app and access the "Billing" module. 6. Select the "Outgoing" option. 7. Input product details: Name, Quantity. 8. Submit by tapping "Click Here." 9. Review the updated "Remaining" quantity. 10. If necessary, endorse the action by tapping "Approved." 11. Confirm the displayed approval status.
Expected Results	iii. Successful submission without errors. iv. "Remaining" quantity updates based on the submitted quantity.
Test Results	Pass
Post conditions	iii. The outgoing product is recorded in the system with updated details. iv. Approval status reflects the recent action.

TEST CASE: Add Product Case No: 10.

	10
Test Case	Add Product
Description	the shopkeeper mobile app adds a new product by scanning a barcode.
Preconditions	The app has access to the device's camera
Steps	8. Open the app and navigate to the "Add Product" section. 9. Choose the option to "Scan Barcode." 10. Scan the barcode on the product. 11. Confirm or enter the product name. 12. Enter the price per unit. 13. Save or submit the product details.
Expected Results	3. Successful scanning of the barcode without errors. 4. Accurate retrieval of product information from the barcode. 5. Proper recording of product name and price per unit.
Test Results	Pass
Post conditions	4. The new product is added to the inventory with the scanned details. 5. Price per unit is recorded accurately.

TEST CASE: **Product Added Successfully** **Case No: 11.**

Test Case ID	11
Test Case	Product Added Successfully
Description	Verify that the shopkeeper mobile app accurately confirms the successful addition of a product.
Preconditions	iii. Shopkeeper is logged into the mobile application. iv. Stable network connection
Steps	4. Open the app and navigate to the "Product Addition" section. 5. Enter product details: Name, Price, Quantity, etc. 6. Submit the product details. 7. Confirm the displayed message: "Product added successfully."
Expected Results	3. The product submission process completes without errors. 4. The confirmation message "Product added successfully" is displayed.
Test Results	Pass
Post conditions	4. The product submission is successful. 5. The confirmation message is visibly presented

TEST CASE: SETTING **Case No: 12.**

Test Case ID	12
Test Case	SETTING
Description	Verify that the shopkeeper mobile app correctly updates account settings including profile, password, and notifications.
Preconditions	3. Shopkeeper is logged into the mobile application. 4. Stable network connection
Steps	9. Open the app and go to the "Settings" section. 10. Select "Account." 11. Update the profile information. 12. Change the password. 13. Adjust notification preferences. 14. Save the changes.
Expected Results	3. Profile details, password, and notification preferences are updated without errors. 4. The system confirms the successful update
Test Results	Pass
Post conditions	3. The shopkeeper's account settings are updated in the system. 4. Confirmation of successful changes is displayed.

X

DANIYAL NAWAZ
FACULTY