# Cloud Computing Workshop with AWS

## Detailed Project / Product Document

### Project Information

Project Name:  
PrepPal  
  
Student Names:  
Matan Moskovich, Niv Siman Tov, Lilach Kupherstein

This document will fully describe the project content (10-30 pages).

Submission date: Feb 1, 2025.

The below are mandatory details to the document. Feel free to add any additional information as you like.

### 1. Introduction

The grocery shopping experience has remained relatively unchanged for decades, requiring individuals to either visit a supermarket physically or manually search for and select items on online platforms. Both options can be time-consuming and inefficient. For individuals with disabilities, mobility challenges, or those with busy lifestyles, this process becomes even more daunting. Advances in AI and digital services now offer an opportunity to revolutionize grocery shopping by integrating intelligent conversational interfaces with seamless delivery mechanisms. Our project leverages these technologies to create an inclusive, user-friendly platform that simplifies the entire process, from product selection to grocery delivery, ensuring convenience for all users.

Traditional grocery shopping and even online grocery platforms require users to manually select individual items, which can be tedious and time-consuming. Additionally, there is often no integration between meal planning and item selection, leaving users to calculate the ingredients needed for recipes on their own. For individuals with disabilities, the elderly, or those unfamiliar with online systems, these processes can be particularly difficult. Current solutions lack a streamlined, intuitive system that combines product planning, selection, and grocery delivery in a single conversational interface.

Our proposed solution is a conversational, AI-driven grocery shopping platform. Users can engage in a simple chat to specify their meal preferences or shopping needs, and the system will provide tailored product suggestions based on the user's input, generate a complete shopping list of required items, automatically select a nearby supermarket based on the user's location, and enable a seamless order process through an integrated delivery service like Wolt. The platform combines AI, using the ChatGPT API for natural language processing,

and inventory integration from multiple supermarkets to ensure availability and accuracy of items. This creates a more accessible, intuitive, and efficient way to shop for groceries.

Existing grocery platforms primarily focus on traditional online shopping, where users must manually search for products and add them to their cart. Recipe platforms, on the other hand, provide suggestions but do not integrate with grocery delivery services. Some alternative approaches include platforms like Shufersal Online, which allow online grocery shopping but lack product planning integration, and recipe-focused websites such as Foody, which provide recipe suggestions but lack the option for direct grocery ordering. Our solution stands out by integrating these functionalities into a single, seamless conversational interface, specifically catering to accessibility needs and convenience.

The platform introduces several key innovations: conversational shopping, where users interact with the platform in natural language to generate shopping lists and place orders without manual browsing; enhanced accessibility, with features designed to assist individuals with disabilities or limited tech knowledge; seamless integration, combining product planning, selection, and delivery into one streamlined process; and personalization, providing tailored recommendations based on user preferences and dietary needs.

This solution caters to a wide target audience, including individuals with disabilities who require a more accessible shopping experience, busy families or professionals who seek quick and efficient grocery solutions, cooking enthusiasts looking for inspiration and easy ingredient ordering, and elderly individuals who may find traditional shopping platforms overwhelming. By addressing the challenges of traditional and online grocery shopping methods, this project aims to create a truly inclusive, innovative solution that redefines convenience and accessibility in grocery shopping.

### 2. System Main components

primary components

תמונה שמכילה תרשים, קו, טקסט, מלבן

התיאור נוצר באופן אוטומטי

### 3. System Architecture including AWS modules used per module

use AWS Lambda for small-scale operations involving microservices and AWS ECS for managing containers. Here's a more detailed breakdown of what you're doing:

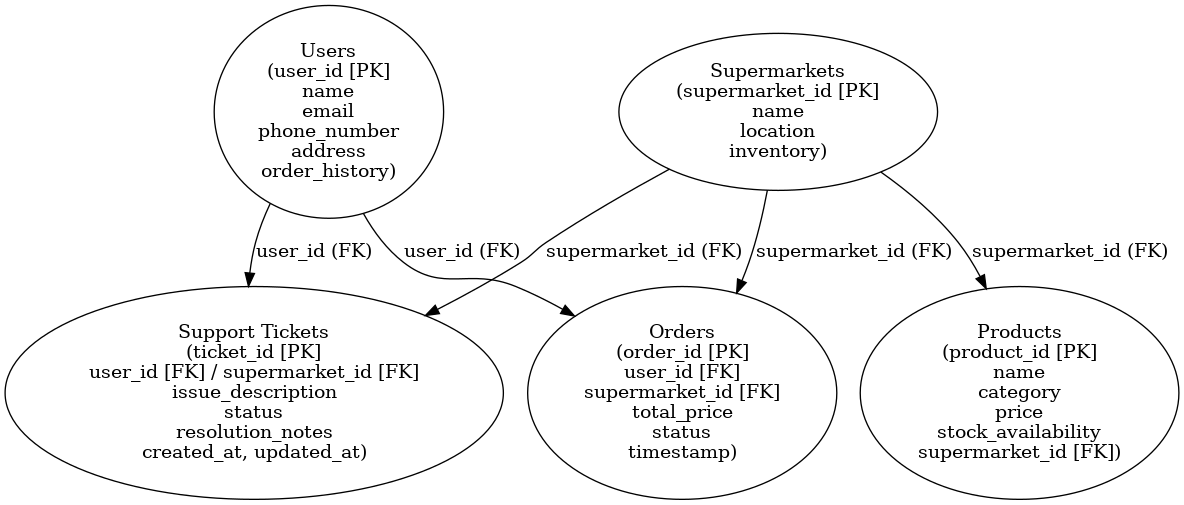
* **AWS Lambda**: Handling small, event-driven tasks with serverless computing. This allows you to run functions in response to specific triggers, without needing to manage the underlying servers.
* **AWS ECS (Elastic Container Service)**: Deploying, managing, and scaling containerized applications. ECS helps you organize workloads into containers, facilitating efficient resource usage and high availability.
* use AWS S3 for storage purposes. It allows you to securely store, retrieve, and manage data at scale, often as part of a larger cloud infrastructure.

### 4. Data Base

**Chosen Database and Rationale**

The PrepPal system utilizes **Amazon DynamoDB** as the primary database, supplemented by SQLite for local storage in certain use cases. This decision is based on several factors:

1. **Scalability and Performance** – DynamoDB provides a fully managed NoSQL **database**, capable of handling high read/write throughput efficiently. This is crucial for real-time inventory updates, user interactions, and order management.
2. **Flexibility** – Since product catalogs and shopping lists vary per supermarket, a NoSQL schema allows dynamic storage of different data structures without predefined table relationships.
3. **Serverless Architecture Integration** – DynamoDB seamlessly integrates with **AWS Lambda and AWS ECS**, reducing operational overhead and ensuring a **highly available, serverless backend**.
4. **Fast Querying** – The ability to perform key-value lookups and secondary index searches enhances system responsiveness, making searches for products and orders quick and efficient.
5. **Local Data Storage (SQLite)** – SQLite is used for offline functionality in cases where temporary data caching is needed before synchronization with DynamoDB.

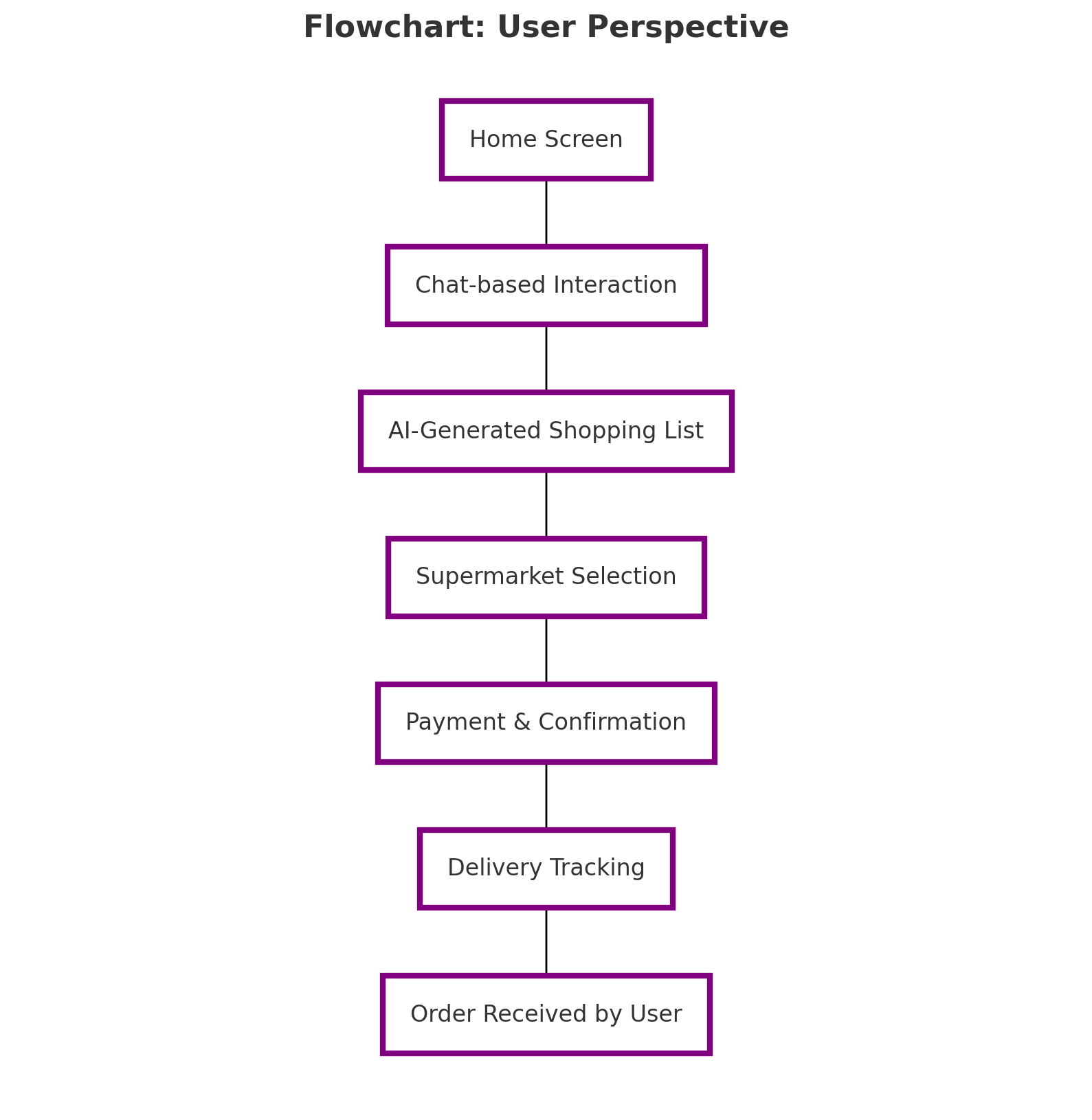


### 5. Flow charts

**Flow Chart – User**

This flowchart outlines the user's journey from entering the platform to receiving their order.

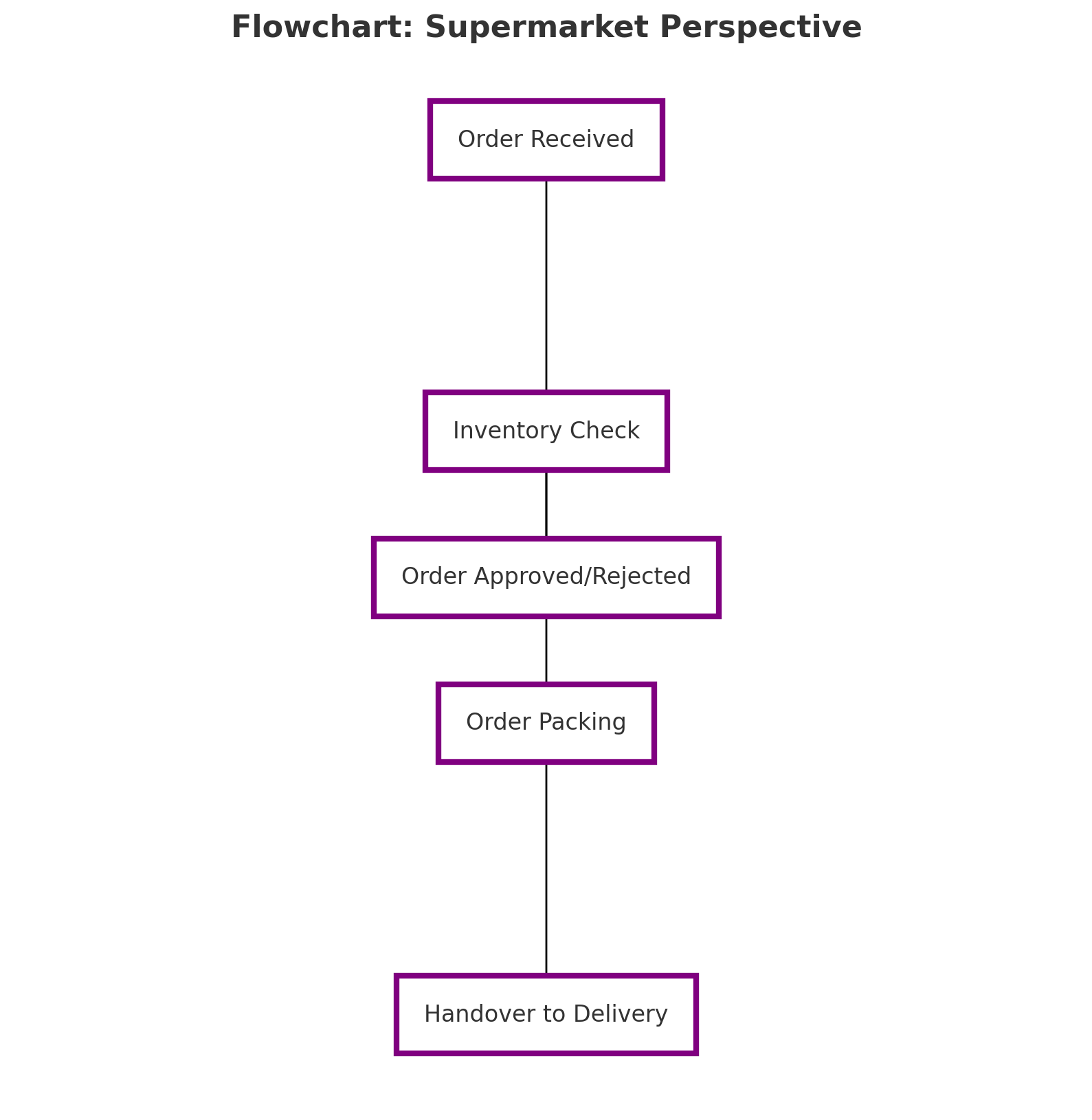
The steps include:

* Home Screen – The user starts their experience from the main interface.
* Chat-Based Interaction – The user communicates their shopping needs through an AI-powered chat system.
* AI-Generated Shopping List – The system generates a personalized shopping list based on user preferences.
* Supermarket Selection – The system selects the appropriate supermarket based on availability and location.
* Payment & Order Confirmation – The user confirms their order and completes the payment process.
* Delivery Tracking – The user can track the real-time status of their order.
* Order Received by User – The final stage, where the user receives the purchased items.

**Flow Chart – Store**

This flowchart outlines the process the supermarket follows from receiving an order to handing it over for delivery.

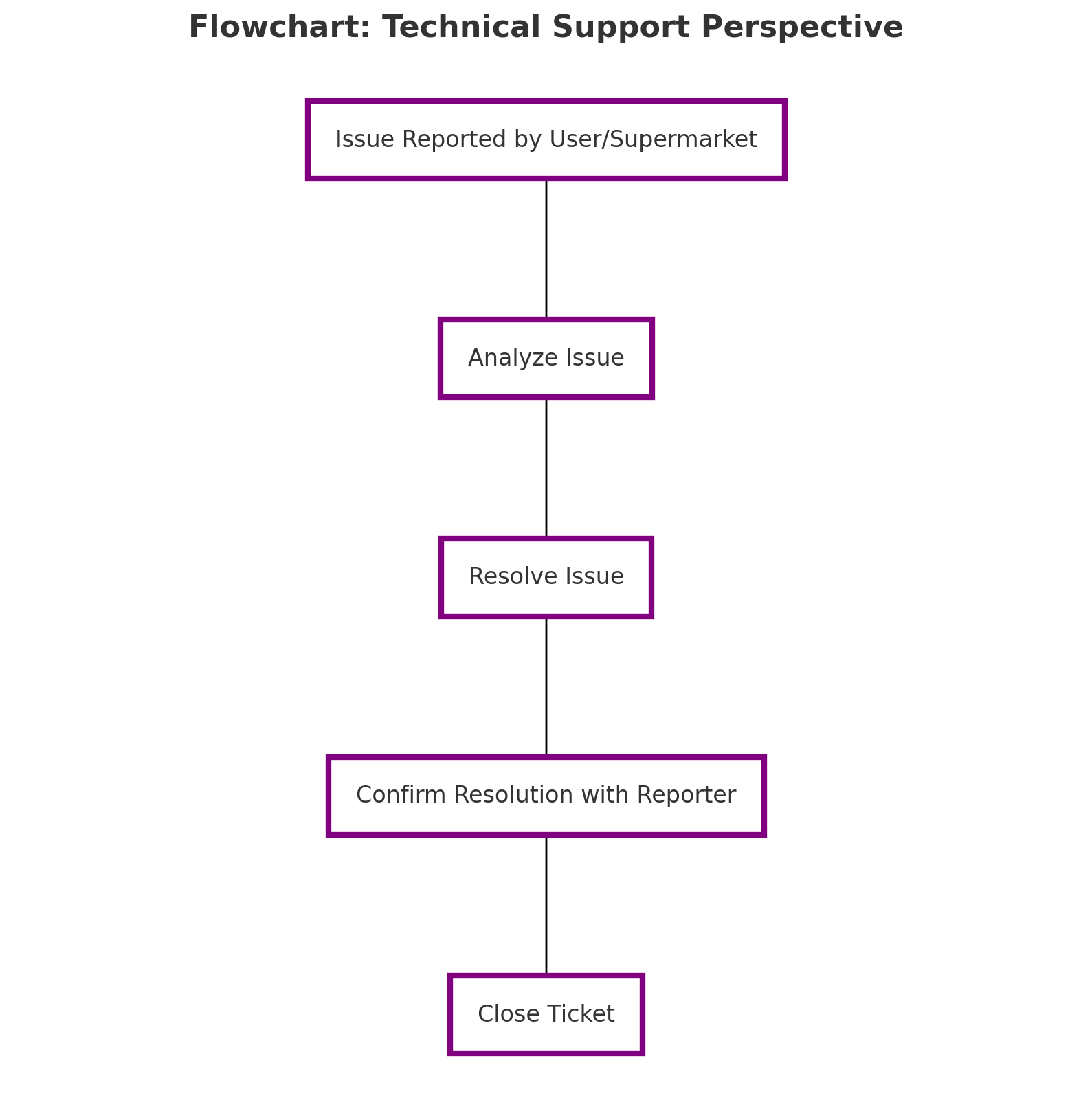
The steps include:

* Order Received – The supermarket receives the order placed by the user.
* Inventory Check – The system verifies the availability of the requested items.
* Order Approved/Rejected – The order is either confirmed if all items are available or rejected if items are out of stock.
* Order Packing – If approved, the supermarket staff packs the items for delivery.
* Handover to Delivery – The packed order is handed over to the designated delivery service.

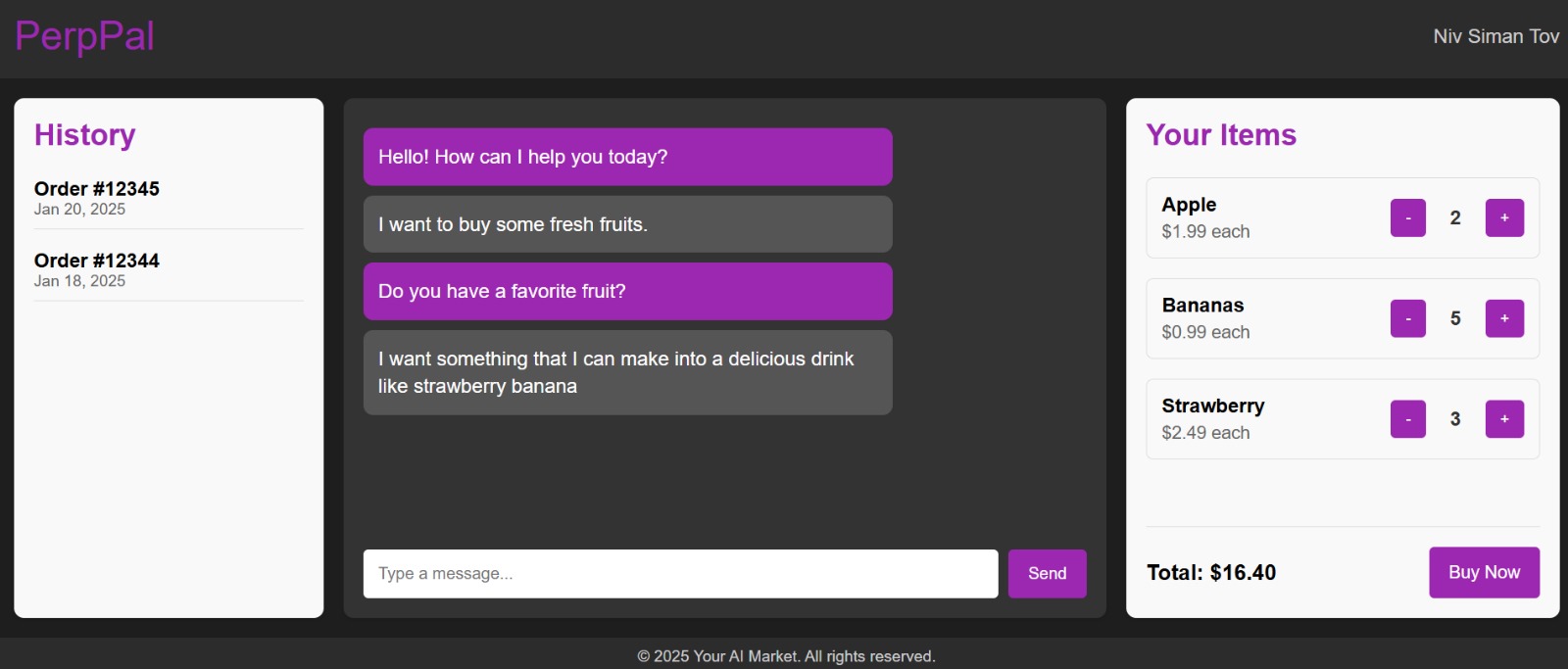
**Flow Chart – Technical Support**

This flowchart illustrates the process followed by the technical support team when handling issues reported by users or supermarkets.

The steps include:

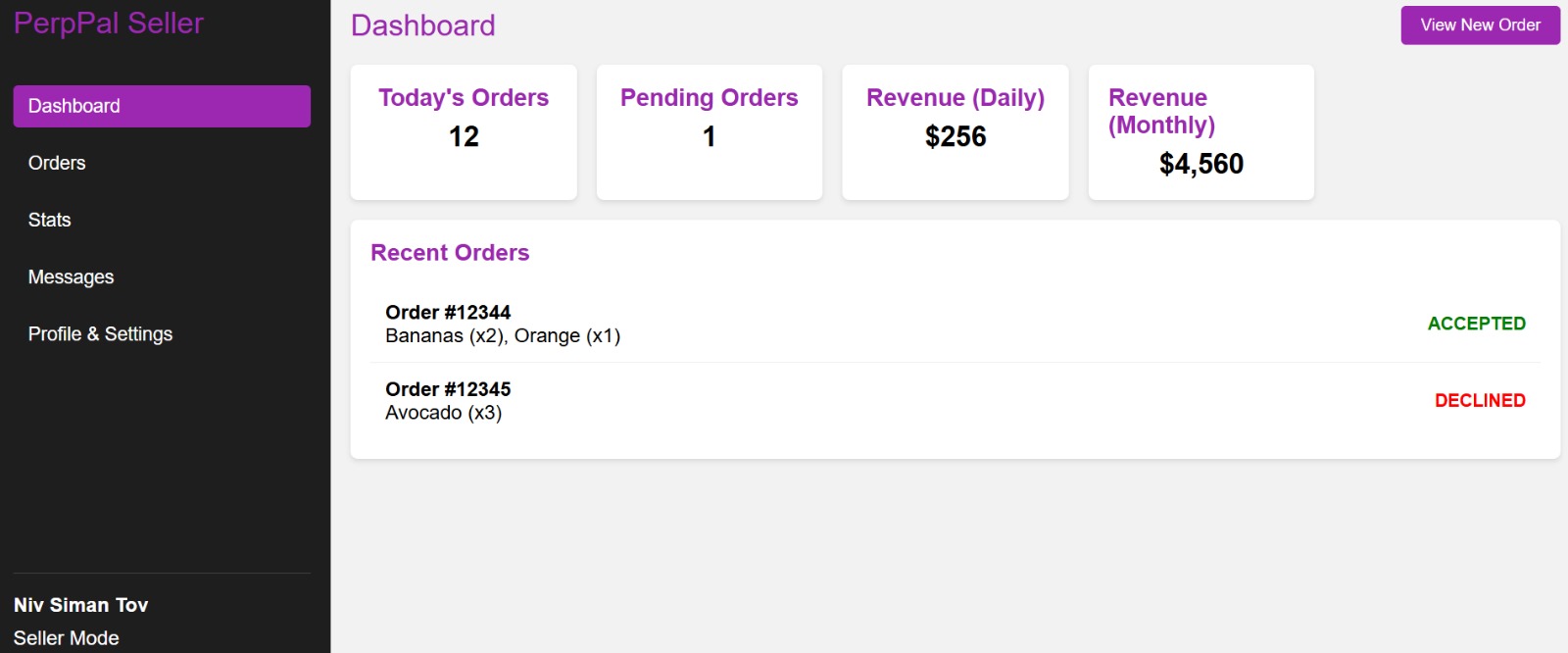
* Issue Reported by User/Supermarket – The support team receives a complaint or issue from either a user or a supermarket.
* Analyze Issue – The team investigates the problem to determine its cause and severity.
* Resolve Issue – A solution is implemented to address the problem, whether technical or operational.
* Confirm Resolution with Reporter – The support team follows up with the user or supermarket to ensure the issue has been resolved.
* Close Ticket – Once confirmed, the issue is marked as resolved and officially closed.

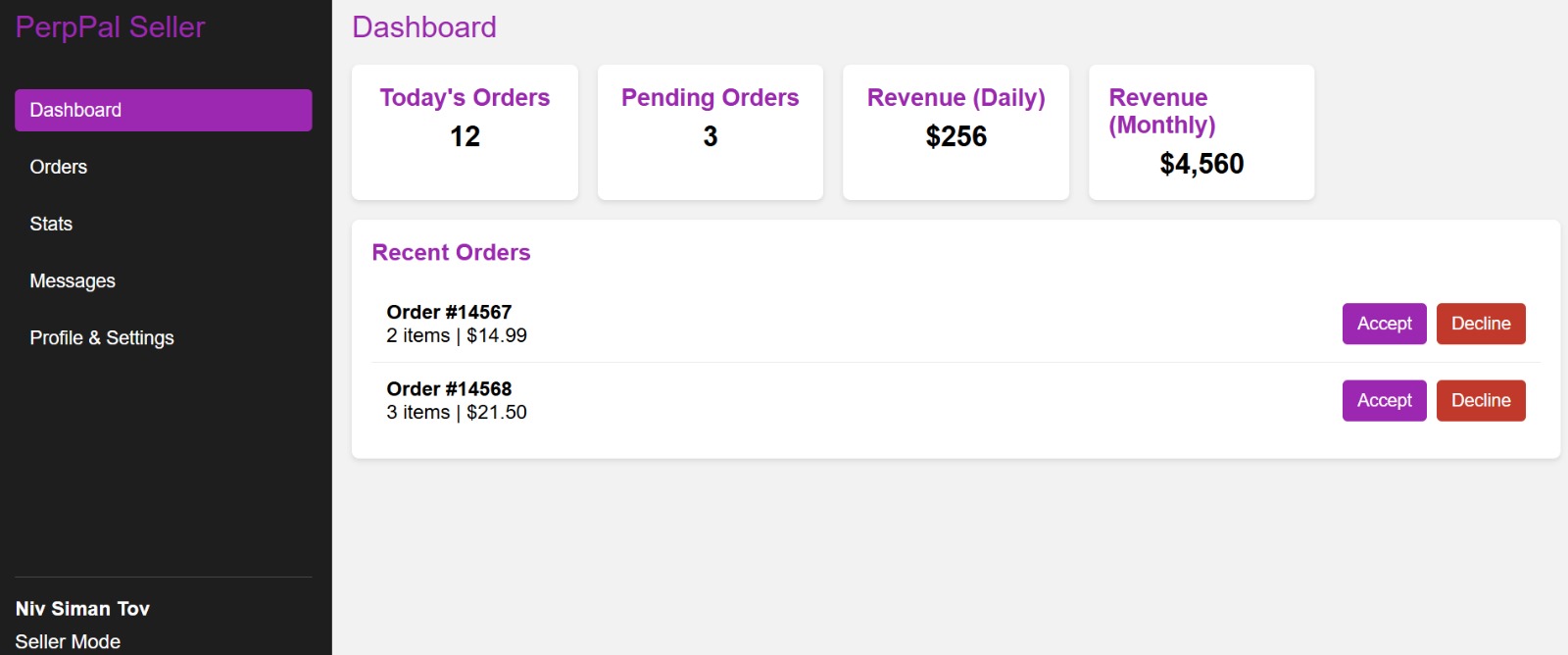
**6. Screen mockups**

Home Page

### 

Shipping Details:

Seller:



### 7. Development tools

Programming languages:

* Java Script
* C#
* Python

Frameworks:

* React
* ASP.Net
* Entity framework

IDEs:

* Visual Studio
* VS code
* WebStorm

APIs:

* AWS
* OpenAI
* Wolt
* Bit

Database:

* sqLite
* Amazon DynamoDB

Tools:

* GPT
* CoPilot

### 8. Submission Details

### <https://github.com/lilachKup/PrepPal.git>