DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

PHARMACY INVENTORY MANAGEMENT SYSTEM

Database Management System – CD252IA
Experiential Learning (Lab)

Design of Forms, Security, and Validation Report

Submitted by

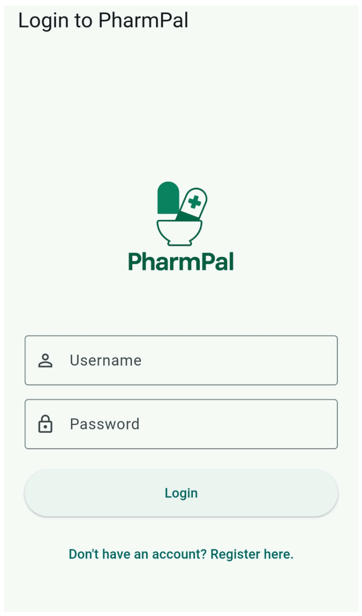
Kushagra Bashisth	1RV23IS064
Mohammad Oweis	1RV23IS072

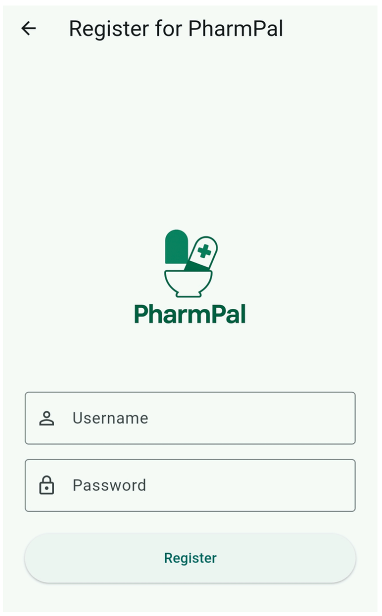
A. Design of Forms & Frontend Screenshots

The PharmPal application is built using Flutter, a modern engineering tool, to provide a clean, responsive, and user-friendly interface. The UI is designed to be intuitive for pharmacy staff, minimizing clicks and simplifying complex tasks. Below are descriptions and representations of four key forms.

1. Login Screen

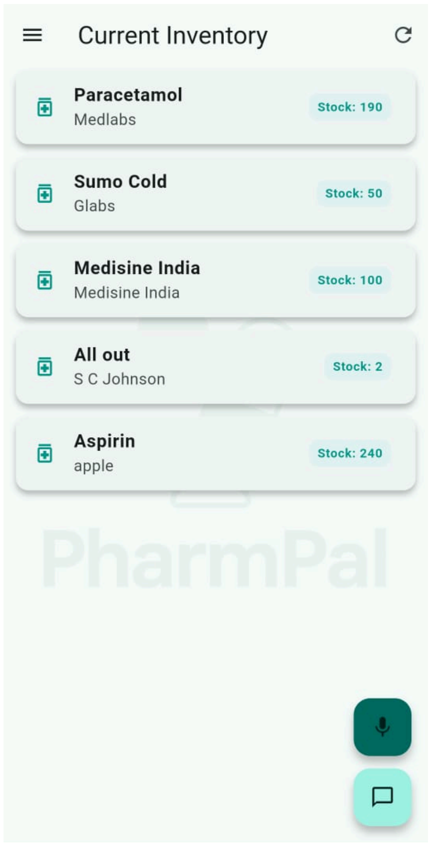
- Purpose: The primary entry point for the application, ensuring only authenticated users can access the system.
- Design: A clean and simple form with fields for "Username" and "Password". It includes the "PharmPal" logo for branding and a clear "Login" button. A text link provides navigation to the Registration Screen for new users. The design is centered and responsive, adapting well to different screen sizes and keyboard appearances. If the user is not already registered, he/she can register using the registration page.



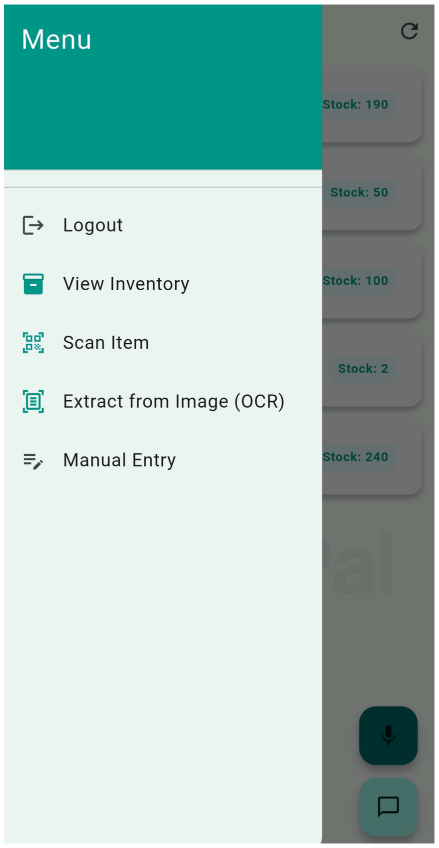


2. Main Inventory Screen

- Purpose: The central dashboard of the application. It displays a real-time list of all medicines in the user's specific inventory.
- Design: Uses a ListView of Card widgets for a modern look. Each card clearly displays the medicine's name, manufacturer, and total stock quantity. The screen features a "pull-to-refresh" gesture, a manual refresh button, a navigation drawer (hamburger menu), and Floating Action Buttons for quick access to the voice entry and chatbot features. The swipe-to-delete gesture provides an intuitive way to remove items.

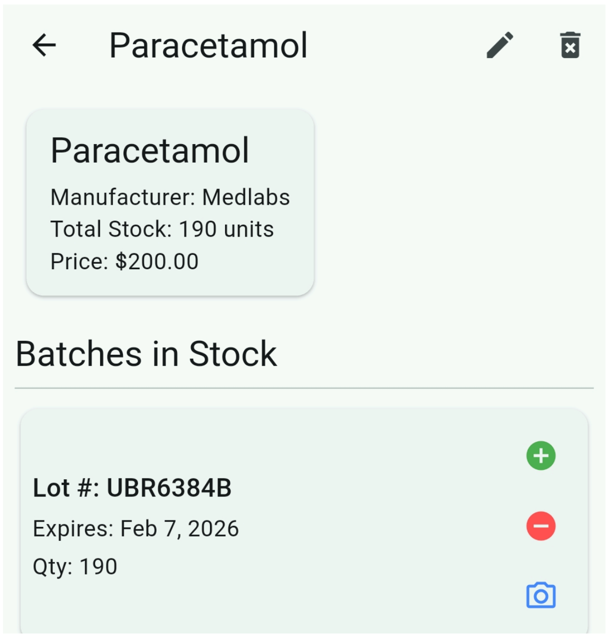

Current Inventory

Paracetamol Medlabs	Stock: 190
Sumo Cold Glabs	Stock: 50
Medisine India Medisine India	Stock: 100
All out S C Johnson	Stock: 2
Aspirin apple	Stock: 240



3.Medicine Detail Screen

- Purpose: Provides a detailed breakdown of a specific medicine, showing all the individual batches in stock.
- Design: This screen displays master information (name, total stock) at the top. The main area is a list of batches, with each card showing the Lot Number, Quantity, and Expiry Date. Each batch has interactive buttons to "Dispense" (decrease quantity) and "Restock" (increase quantity), providing a clear and functional interface for stock management. The AppBar includes "Edit" and "Delete" actions for the master product.


Paracetamol

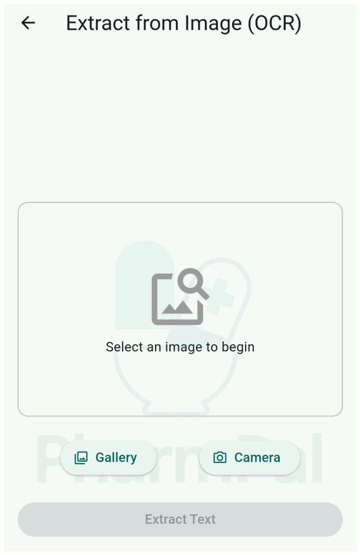
Paracetamol
Manufacturer: Medlabs
Total Stock: 190 units
Price: $200.00

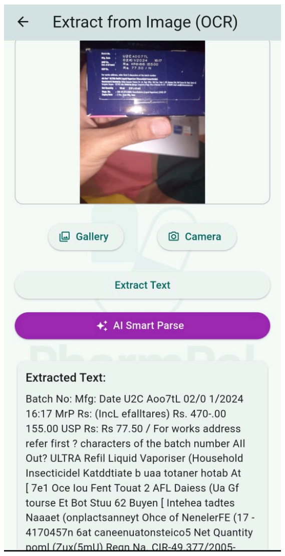
Batches in Stock

Lot #: UBR6384B
Expires: Feb 7, 2026
Qty: 190

4.Manual/OCR Entry Form (CreateMedicineScreen)

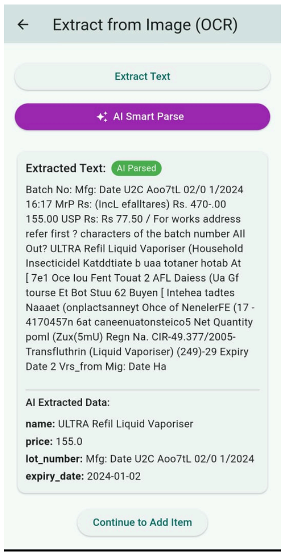
- Purpose: A versatile form used for manual entry, for creating a new item after a barcode scan fails to find a match, and for adding an item after OCR extraction.
- Design: A scrollable ListView containing clearly labeled TextField widgets for all required information (Name, Manufacturer, Price, Lot Number, Quantity, etc.). It uses a date picker for selecting the expiry date. When used in the OCR workflow, the Price, Expiry Date, and Lot Number fields are intelligently pre-filled, saving the user time.





Extracted Text:

Batch No: Mfg: Date U2C Aoo7tL 02/0 1/2024 16:17 MrP Rs: (IncL efalltares) Rs. 470-.00 155.00 USP Rs: Rs 77.50 / For works address refer first ? characters of the batch number All Out? ULTRA Refil Liquid Vaporiser (Household Insecticidel Katddtiate b uaa totaner hotab At [7e1 Oce Iou Fent Touat 2 AFL Daiess (Ua Gf tourse Et Bot Stuu 62 Buyen [Intehea tadtes Naaaet (onplactsanneyt Ohce of NenelerFE (17 - 4170457n 6at caneenuatonsteico5 Net Quantity poml (Zux(5mU) Regn Na. CIR-49.377/2005-

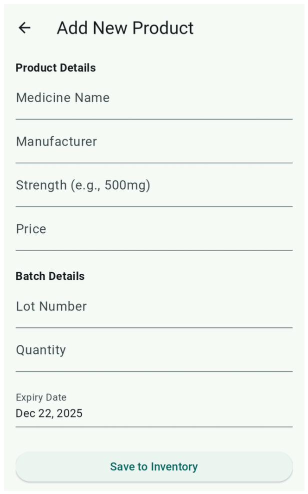


Extracted Text: AI Parsed

Batch No: Mfg: Date U2C Aoo7tL 02/0 1/2024 16:17 MrP Rs: (IncL efalltares) Rs. 470-.00 155.00 USP Rs: Rs 77.50 / For works address refer first ? characters of the batch number All Out? ULTRA Refil Liquid Vaporiser (Household Insecticidel Katddtiate b uaa totaner hotab At [7e1 Oce Iou Fent Touat 2 AFL Daiess (Ua Gf tourse Et Bot Stuu 62 Buyen [Intehea tadtes Naaaet (onplactsanneyt Ohce of NenelerFE (17 - 4170457n 6at caneenuatonsteico5 Net Quantity poml (Zux(5mU) Regn Na. CIR-49.377/2005- Transfluthrin (Liquid Vaporiser) (249)-29 Expiry Date 2 Vrs_from Mig: Date Ha

AI Extracted Data:

name: ULTRA Refil Liquid Vaporiser
price: 155.0
lot_number: Mfg: Date U2C Aoo7tL 02/0 1/2024
expiry_date: 2024-01-02

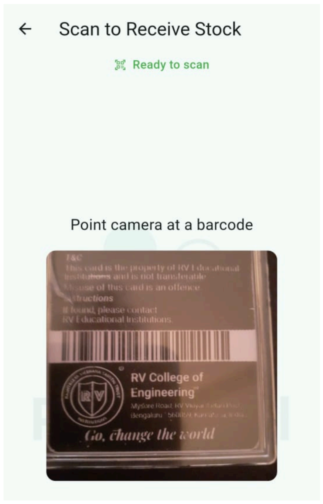


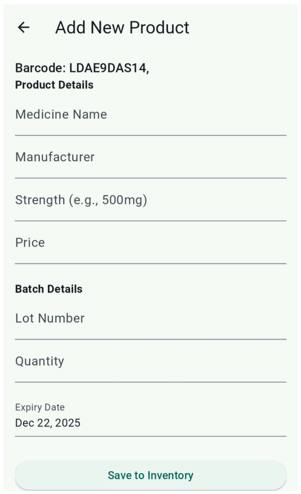
Product Details

Medicine Name
Manufacturer
Strength (e.g., 500mg)
Price

Batch Details

Lot Number
Quantity
Expiry Date: Dec 22, 2025





Barcode: LDAE9DAS14,
Product Details

Medicine Name
Manufacturer
Strength (e.g., 500mg)
Price

Batch Details

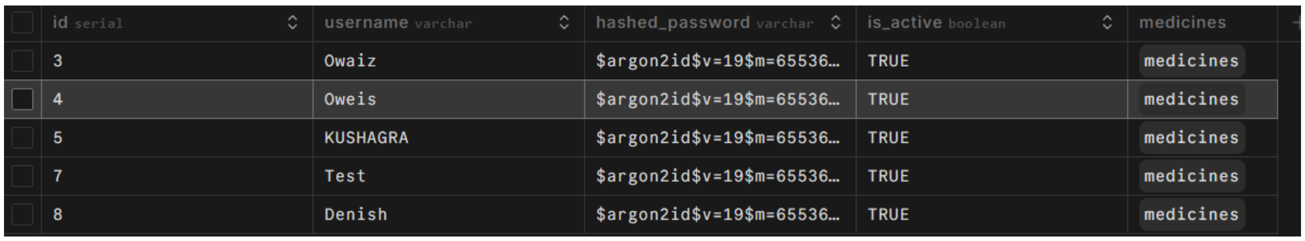
Lot Number
Quantity
Expiry Date: Dec 22, 2025

B. Security and Validation Proof

Security:

- Authentication: The system is built on a robust multi-user authentication model using JSON Web Tokens (JWT). Users must register and log in to access any part of the application. The Flutter app securely stores the JWT on the device using flutter_secure_storage.
- Password Hashing: User passwords are never stored in plain text. They are securely hashed on the backend using the modern Argon2 algorithm via the passlib library before being stored in the database.
- Data Segregation (Multi-Tenancy): This is the core security principle. Every data--related table in the database (e.g., medicines) has a user_id foreign key. Every single API endpoint is protected by a dependency that verifies the user's JWT and loads their user profile. All subsequent database queries are strictly filtered by the current user's ID, making it architecturally impossible for one user to view, edit, or delete another user's data.

```
app.post("/medicines/smart-create", response_model=schemas.Medicine)
def smart_create_medicine_and_inventory(
  request: schemas.SmartCreateRequest,
  db: Session = Depends(get_db),
# This dependency ensures the user is logged in
  current_user: models.User = Depends(auth.get_current_active_user)
):
  """Smart create medicine with inventory (secured to current user)"""
# The query is filtered by the logged-in user's ID
  return _smart_create_db_entry(request, db, user_id=current_user.id)
```



id serial	username varchar	hashed_password varchar	is_active boolean	medicines
3	Owaiz	$argon2id$v=19$m=65536…	TRUE	medicines
4	Oweis	$argon2id$v=19$m=65536…	TRUE	medicines
5	KUSHAGRA	$argon2id$v=19$m=65536…	TRUE	medicines
7	Test	$argon2id$v=19$m=65536…	TRUE	medicines
8	Denish	$argon2id$v=19$m=65536…	TRUE	medicines

Validation:

- Frontend Validation: The Flutter forms have basic client-side validation, such as ensuring that the "Quantity" field is not empty and contains a valid number before the API call is made.
- Backend Validation: The FastAPI backend uses Pydantic for powerful, automatic data validation. Every incoming request is validated against a schema. If a client sends data of the wrong type (e.g., sending the string "ten" for a quantity field that expects an integer), the request is automatically rejected with a 422 Unprocessable Entity error, ensuring that no invalid data can ever reach the database.

```
class SmartCreateRequest(BaseModel):
  # Fields for the Medicine catalog item
  barcode: Optional[str] = None
  name: str
  manufacturer: Optional[str] = None
  strength: Optional[str] = None
  price: float

  # Fields for the first InventoryItem batch
  lot_number: str
  quantity: int
  expiry_date: date
```

Innovative Experiment: AI Chatbot with Real-Time Database Tool Use

The most innovative component of the PharmPal project is the PharmPal Assistant, an AI-powered chatbot that functions as a natural language interface to the user's live inventory database. This experiment goes beyond simple Q&A by implementing a "Tool Use" (or "Function Calling") architecture, allowing the AI to intelligently use the backend API as a tool to answer questions factually and accurately.

Concept:

The primary challenge in inventory management is the rapid retrieval of specific information. A user might need to quickly know their stock levels or identify expiring products. While a UI can provide this, a natural language interface is faster and more intuitive. The innovative experiment was to create a chatbot that doesn't just "know" about medicine but can actively query the user's personal, real-time database to provide live, accurate answers.

Working:

The system uses the high-speed Groq API (running Llama 3) and is orchestrated by our FastAPI backend.

- User Input: The user asks a question in the Flutter app, e.g., "What medicines are expiring in the next 90 days?"

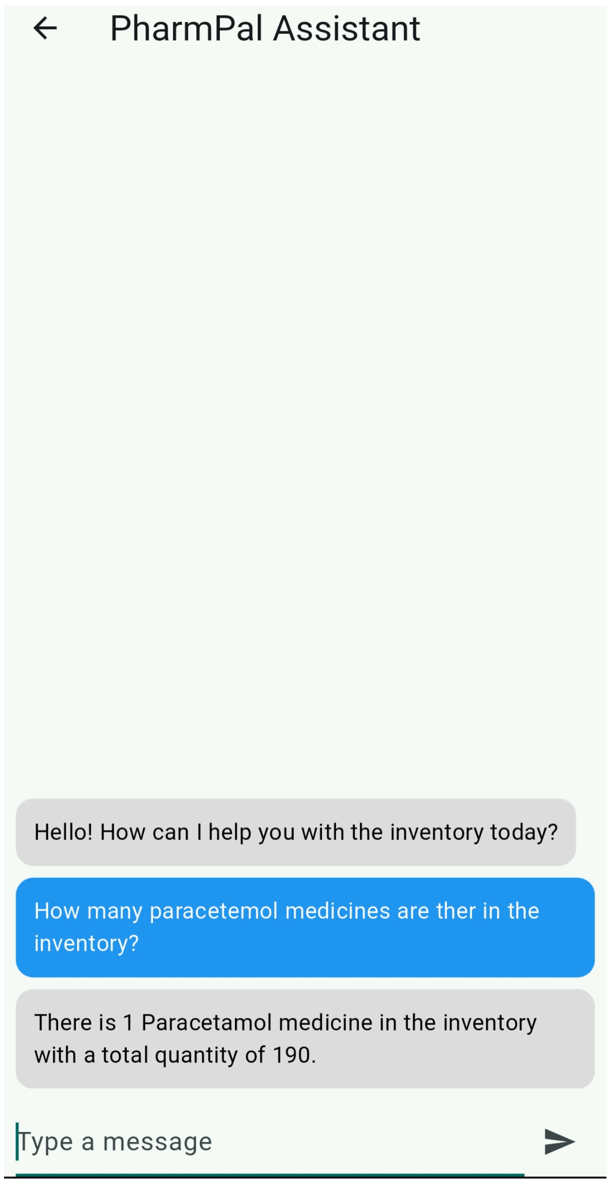
- First AI Call (Intent Detection): The FastAPI backend does not try to answer the question. Instead, it sends the user's query to the Groq API along with a "menu" of available tools (our database functions, like find_expiring_medicines).

- AI Response (Tool Selection): The Groq LLM analyzes the query and, instead of a conversational reply, it returns a structured JSON command: {"tool_name": "find_expiring_medicines", "parameters": {"days_limit": 90}}.

- Backend Execution (Database Query): The FastAPI backend parses this JSON command. It calls its internal find_expiring_medicines_from_db function with the parameter 90. This function executes a real SQL query against the Neon PostgreSQL database.

- Data Retrieval: The database returns a list of medicines that match the criteria.
- Second AI Call (Response Formulation): The backend sends the raw data from the database back to the Groq API with a new prompt: "The user asked 'What's expiring...?' and the database returned this data: [...list of medicines...]. Now, formulate a friendly, human-readable answer."
- AI Response (Final Answer): The Groq LLM generates the final, conversational response: "Here are the items expiring in the next 90 days: - Paracetamol (Lot: ABC-123), - Aspirin (Lot: GHI-789)..."
- Display: This final answer is sent to the Flutter app and displayed to the user.

This architecture combines the vast language understanding of a powerful LLM with the factual, real-time "ground truth" of a structured database. The AI is not guessing; it's using the database as its source of knowledge, preventing hallucinations and ensuring the answers are always accurate and relevant to the user's specific inventory. This creates a truly intelligent and reliable assistant that dramatically enhances the user experience.



Hello! How can I help you with the inventory today?

How many paracetemol medicines are ther in the inventory?

There is 1 Paracetamol medicine in the inventory with a total quantity of 190.