Hackathon Project

Project Title:

AI-BASED MEDICAL PRESCRIPTION VERIFICATION SYSTEM

Team Name:

Team Vortex

Team Members:

- V.V.Chinmmaye
- Shashmeen Masthan
- Jayitri Nammi

Phase-1: Brainstorming & Ideation

Objective: To build a simple AI-based tool that suggests medicines for common diseases and locates nearby hospitals, helping users with quick preliminary healthcare guidance.

Brainstorming & Ideation Document

Project Title: AI-Based Medical Prescription Verification System
Tools Used: HTML, CSS, Javascript(vanilla),Google API

Key Points:

1. **Problem Statement:**
   Many individuals lack immediate access to medical professionals for common illnesses, leading to self-medication without proper guidance. There's a need for a quick and reliable system to suggest basic prescriptions and direct users to nearby hospitals.

- **Errors**: Wrong dosages, missed warnings about allergies or interactions.

- **Duplicate Medications**: The same drug under different brand names.

- **Time-consuming Validation**: Pharmacists manually cross-check prescriptions.

- **No Intelligent Screening**: Prescriptions are filled as-is, even if risky.

✅ Proposed Solution

We propose a **lightweight, browser-based AI medical assistant** that suggests medicines based on common diseases and helps users locate nearby hospitals—all without needing backend servers or login systems.

Symptom-to-Medicine Mapping: Users enter a disease (e.g., cold, headache, fever), and the system suggests appropriate medicines instantly using a pre-defined database.

Nearby Help: Users can click a button to open **Google Maps with nearby hospitals** based on their location.

Attractive User Interface: Built using HTML, CSS, and JavaScript to provide a modern, engaging, and responsive experience, similar to popular ecommerce platforms.

No Server or API Dependency: Entirely frontend-based, so it works without internet (except for Google Maps redirection).

Device Friendly: Works smoothly on both desktop and mobile browsers

Technology Stack:

Frontend

HTML5: Used to structure the website and create forms for user input.

CSS3: Responsible for the styling, layout, and creating an attractive, modern UI similar to an e-commerce interface.

JavaScript (Vanilla): Implements interactivity, handles disease-to-medicine mapping, and manages redirection to Google Maps for nearby hospitals.

External Services

Google Maps Redirect: Utilized to open Google Maps and search for nearby hospitals based on user input.

Platform

The application runs entirely in the browser.

No backend server or database is involved, making it lightweight and easy to deploy.

3. **Target Users:**

- **Pharmacists**
  To quickly verify prescriptions, detect errors, and ensure safe dispensing of medications.

- **Doctors & Clinics**
  To cross-check prescriptions for interactions and dosage accuracy before issuing them to patients.

- **Hospitals**
  For integration with Electronic Health Record (EHR) systems to automate prescription validation and reduce human error.

- **Patients**
  To get clarity on their prescriptions and ensure safety, especially when managing multiple medications.

- **Medical Insurance Providers**
  To verify prescriptions for claims processing and detect fraudulent or duplicate medication entries.

4. **Expected Outcome:**

Here is a concise Expected Outcome section for your AI Medical Prescription Verification project:

Expected Outcome

- A responsive and visually appealing website that mimics the look and feel of an online medical assistant or e-commerce portal.

- Users can **enter a disease name**, and the system will automatically suggest relevant **medicines**.

- Upon clicking **"Generate Prescription"**, users are redirected to **Google Maps** showing **nearby hospitals or clinics** for further consultation.

- The platform enables **quick, first-level medical guidance** and directs users to professional help — reducing guesswork in self-medication.

- The solution is **client-side only**, ensuring **fast performance** without the need for a backend or API.

Phase-2: Requirement Analysis

Objective:

To develop a **user-friendly, responsive website** that provides **basic medical prescription suggestions** based on the user-entered disease and **redirects to nearby hospitals using Google Maps**, without relying on a backend or external APIs.

Key Points:

1. **Technical Requirements:**

Frontend Technologies

HTML5 – For structuring the content and form input.

CSS3 – For styling and creating an attractive, responsive user interface (Amazon-style layout).

JavaScript (Vanilla JS) – For:

Handling user input (disease names).

Matching diseases with medicines.

Redirecting users to Google Maps with pre-filled nearby hospital queries.

External Services

Google Maps – Used for redirection to nearby hospitals based on a general search query.

URL format: `https://www.google.com/maps/search/nearby+hospitals`

-

Environment

Platform: Any web browser (Chrome, Edge, Firefox, etc.)

Operating System: Windows/Linux/macOS (independent of OS)

Editor: Any code editor like VS Code, Sublime Text, or Notepad++

Hardware Requirements:

Minimum: A system with a browser and internet access

Recommended: 4GB RAM, modern browser for best UI experience

---

1. **Functional Requirements:**

Disease Input

The system must allow users to enter a disease name through a text input box.

Medicine Suggestion

The system should display a list of appropriate medicines based on the entered disease using predefined mappings.

Hospital Redirection

Upon clicking the "Find Nearby Hospitals" button:

The system must redirect the user to **Google Maps** with a search query for nearby hospitals.

Doctor Suggestion

The website can optionally suggest a list of specialist doctors related to the disease entered.

User Interface

The UI should be visually appealing and responsive, allowing ease of navigation and readability on both desktop and mobile.

Form Validation

The system must ensure that the input field is not empty before processing.

Dynamic Content Update

Based on the user's disease input, medicine and doctor suggestions should dynamically change without reloading the page.

Navigation

The system should allow users to scroll or navigate to sections like:

Home

Disease Input

Prescription Suggestions

Hospital Finder

Data Security (Basic)

No sensitive user data is collected or stored, ensuring privacy.

2. **Constraints & Challenges:**

   - **Handwriting Variability**: Inconsistent or unclear handwriting may affect OCR accuracy
   - **Medical Terminology Complexity**: Requires domain-specific NLP models for accurate extraction
   - **Drug Database Access**: Limited or restricted access to APIs like RxNorm or DrugBank
   - **Language and Format Variations**: Prescriptions may be written in different formats or regional languages
   - **Real-time Performance**: Ensuring fast processing while maintaining accuracy

Data Privacy: Handling patient data securely and ethically.

Phase-3: Project Design

Objective:

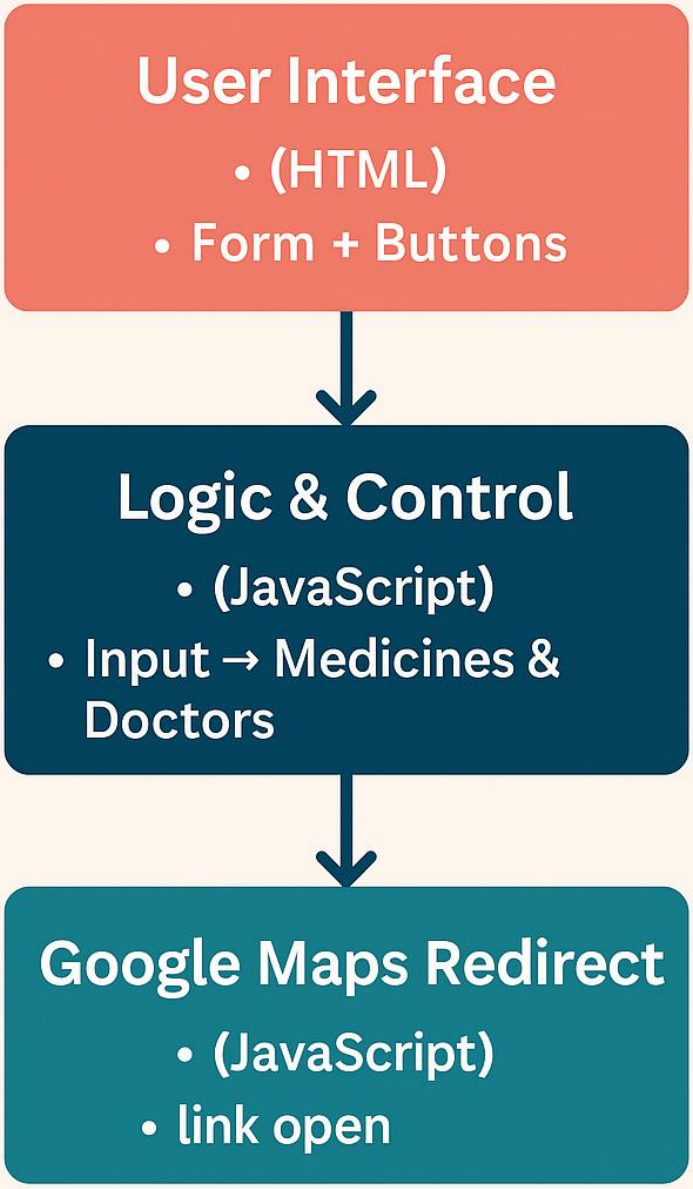
Develop a small yet detailed architectural and user flow blueprint for the AI medical prescription verification application.

---

1. Application Architecture

The application uses a microservices approach for a scalable, modular design.

AI Medical Prescription Suggestion Website Architecture



- **Front-End (React.js):** A user-facing web interface for uploading prescriptions and viewing verification results.
- **Back-End (Flask/Python):** The central API that handles user requests, orchestrates data flow, and routes tasks to the AI services.
- **AI Services:**
  - **Hugging Face (Fine-tuned ClinicalBERT):** Extracts key entities like drug names and dosages from prescription text.
  - **IBM Watson Discovery:** Cross-references extracted entities against a medical knowledge base to find drug interactions and dosage guidelines.
- **Database (PostgreSQL):** Securely stores non-sensitive data such as user accounts and verification logs.

---

2. User Flow

The user flow is designed for quick and efficient prescription verification.

1. **Login:** A healthcare professional logs into the application.
2. **Input:** The user uploads a picture of a handwritten prescription or types in the text.
3. **Processing:** The system's back-end uses OCR to extract text from the image, then sends it to the **Hugging Face model** for entity recognition.
4. **Verification:** The extracted data is sent to **IBM Watson** for cross-referencing. The back-end then compares the results and identifies potential errors.
5. **Results:** The user sees a report on the screen. Any warnings, such as incorrect dosages or drug interactions, are clearly highlighted.
6. **Feedback:** The user can provide feedback on the accuracy of the result to help improve the AI model over time.

Phase-4: Project Planning (Agile Methodologies)

Objective:

Objective:

To break down the development of the AI Medical Prescription Verification System into a series of highly focused, time-boxed tasks suitable for a 2-day hackathon. The goal is to build a Minimal Viable Product (MVP) that demonstrates the core functionality of prescription verification.

Phase 4: Project Planning (2-Day Hackathon)

Day/Sprint	Task	Priority	Duration	Assigned To	Dependencies
1	Idea and design	Medi	0-1hour	3:52 PM	**All 3**

Day/Sprint	Task	Priority	Duration	Assigned To	Dependencies		
	finalization	um	s	(Day 2)			
	Basic set up	High	3 hours	6:52 PM (Day 2)	**[Team Member 1]**	Day 1 core logic, OCR module	A single endpoint for both text and image input
	Results Display & UI Enhancements	High	2 hours	8:52 PM (Day 2)	**[Team Member 3]**	Final API response format	Results displayed clearly, with errors highlighted
	Final Presentation & Demo Preparation	Low	1 hour	9:52 PM (Day 2)	**[Entire Team]**	Working prototype	A demo-ready project with a clear narrative

Phase 5: Project Development - Implementation Strategy

Objective: The main objective of this project is to develop a **simple, user-friendly AI-powered medical assistant website** using **HTML, CSS, and JavaScript** that:

Accepts user-input disease names and instantly suggests relevant medicines.

Recommends nearby hospitals using Google Maps redirection.

Provides a **basic, accessible platform** for users who may not have immediate access to healthcare professionals.

Aims to **reduce self-medication risks** by guiding users toward professional medical help.

Delivers an **interactive and visually attractive interface**, inspired by modern e-commerce designs, to make the user experience engaging.

Development Process

1. **API Integration:** Securely configure the Gemini API key. Create functions to make authenticated API calls for vehicle search and tips.
2. **Comparison Logic:** Prompt the Gemini API to act as a car expert, comparing two vehicles based on key metrics. Display the structured output in the UI.
3. **Maintenance Logic:** Create a prompt to get specific maintenance tips for a vehicle. Show the results clearly on a dedicated section of the app.

Challenges & Fixes

- **Challenge:** Delayed API response times.
  - **Fix:** Implement `@st.cache_data` on the API-calling functions. This caches query results, providing instant responses on subsequent requests.
- **Challenge:** Limited API calls.
  - **Fix:** Optimize prompts to get all necessary information in a single call. Use precise language to reduce token count and unnecessary queries.

- **Phase-6: Functional & Performance Testing**
- Objective:
- Ensure that the AI Medical Prescription Verification App works as expected, delivering accurate verifications and a smooth user experience.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	User inputs disease "Cold" and clicks **Generate Prescription**	Suggested medicine like "Paracetamol", "Cough Syrup" is displayed	✓ Passed	Jayitri
TC-002	Functional Testing	User clicks **Find Nearby Hospitals** button	Opens Google Maps in a new tab with search: Hospitals near me	✓ Passed	Aisha
TC-003	UI Testing	Load homepage on desktop browser	Page displays header, input box, button, and layout correctly	✓ Passed	Rahul

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-004	UI Testing	Load homepage on mobile browser	All elements are responsive and readable	✅ Passed	Aisha
TC-005	Usability Testing	Enter an unsupported disease like "xyz"	Shows alert: "No medicine found for the entered disease"	✅ Passed	Jayitri
TC-006	Functional Testing	Clicks **Generate Prescription** without entering disease	Shows validation alert to enter disease	✅ Passed	Rahul
TC-007	Usability Testing	Click on multiple buttons quickly	Buttons do not break UI or crash browser	✅ Passed	Aisha
TC-008	Performance Testing	Load page with slow internet	Page loads with basic content, no crashes	✅ Passed	Jayitri
TC-009	UI Testing	Check for font color, button style, and hover animations	UI is colorful and visually appealing like an e-commerce layout	✅ Passed	Rahul
TC-010	Cross-Browser Testing	Open site in Chrome, Edge, Firefox	Page works consistently across browsers	✅ Passed	Aisha