# **⚕️ MediLens — AI-Powered Health Symptom & Medical Image Analyzer**

**Submission for: Unthinkable Solutions Company Challenge**

**Project by: Ishaan Sharma (22BCE5122)**

**Institution: Vellore Institute of Technology, Chennai**

## **0. 🔗 Project Links**

* **Deployed Frontend:** [Link to Deployed React App]
* **Deployed Backend:** https://hsc1606.onrender.com
* **YouTube Demo Video:** [Link to YouTube Video]
* **Github Frontend Link:**  [Link to YouTube Video]
* **Github Backend Repository Link:**[Link to YouTube Video]

## **1. 📖 Project Overview**

MediLens is a full-stack, AI-driven med-tech web application designed for rapid and intelligent diagnostic analysis. It provides users with a seamless interface to input health symptoms via text or upload medical images (e.g., skin conditions, X-rays), receiving instant, AI-powered insights. The application is built with a hyper-modern, clinical-grade user interface that prioritizes speed, accuracy, and user experience. It intelligently uses geolocation to provide context-aware results, including nearby medical facilities, making it a powerful tool for preliminary health assessment.

## **2. 🛠️ Tech Stack & Architecture**

The application follows a decoupled frontend-backend architecture, ensuring scalability and maintainability.

### **Frontend (Client-Side)**



| **Component** | **Technology / Library** | **Purpose** |
| --- | --- | --- |
| **Framework** | ⚛️ React 18 + Vite | Blazing fast, modern single-page application (SPA). |
| **Styling** | 🎨 Tailwind CSS & Custom Inline CSS | Responsive design and futuristic neon/glow aesthetics. |
| **Animations** | 🎬 Lottie-React | High-quality animations for splash, auth, loading, etc. |
| **State Management** | 🔄 React Hooks (useState, useEffect, useContext) | Efficient and localized state control. |
| **API Client** | 🌐 Native Fetch API (wrapped in api.js) | Communication with the backend REST API. |
| **Session Storage** | 💾 Browser localStorage | Persisting user sessions and authentication tokens. |
| **Deployment** | 🚀 Render (Static Site) | Continuous deployment and global CDN. |

### **Backend (Server-Side)**

| **Component** | **Technology / Library** | **Purpose** |
| --- | --- | --- |
| **Framework** | 🐍 Python 3.10 + Flask / FastAPI | Robust, scalable, and Python-native REST API development. |
| **AI / ML** | 🧠 TensorFlow / PyTorch | For inference on text (NLP) and image (CV) models. |
| **Database** | 🐘 PostgreSQL / MongoDB | Storing user data, analysis history, and credentials. |
| **Authentication** | 🔐 JWT (JSON Web Tokens) | Secure, stateless authentication for API endpoints. |
| **API Specification** | 📜 OpenAPI (Swagger) | Interactive API documentation and contract definition. |
| **Deployment** | 🐳 Docker + Render (Web Service) | Containerized, scalable, and isolated cloud deployment. |

## **3. 🔌 API Documentation**

The MediLens backend exposes a secure REST API for all client-side operations.

**Base URL:** https://hsc1606.onrender.com

Authentication: All protected endpoints require an Authorization header with a Bearer Token.

Authorization: Bearer <YOUR\_JWT\_TOKEN>

### **Authentication Endpoints**

#### **POST /api/auth/signup**

Registers a new user.

* **Request Body:**

{

"username": "ishaan\_sharma",

"email": "ishaan.sharma@example.com",

"password": "strong\_password\_123"

}

* **Success Response (201 Created):**

{

"message": "User created successfully"

}

* **Error Response (409 Conflict):**

{

"error": "Email or username already exists"

}

#### **POST /api/auth/login**

Authenticates a user and returns a JWT.

* **Request Body:**

{

"email": "ishaan.sharma@example.com",

"password": "strong\_password\_123"

}

* **Success Response (200 OK):**

{

"token": "ey...",

"username": "ishaan\_sharma"

}

* **Error Response (401 Unauthorized):**

{

"error": "Invalid credentials"

}

### **Analysis Endpoints (Protected)**

#### **POST /api/analyze/text**

Analyzes user-provided symptom text.

* **Headers:** Authorization: Bearer <token>
* **Request Body:**

{

"symptoms": "I have a persistent headache, fever, and a sore throat.",

"latitude": 12.8238,

"longitude": 80.2234

}

* **Success Response (200 OK):**

{

"analysisId": "analysis\_12345",

"timestamp": "2025-10-17T12:00:00Z",

"possibleConditions": [

{"name": "Common Cold", "severity": "Mild"},

{"name": "Influenza", "severity": "Moderate"}

],

"recommendations": "Rest, stay hydrated, and consult a doctor if symptoms persist.",

"nearbyHospitals": [

{"name": "Global Health City", "distance": "5.2 km"},

{"name": "Chettinad Health City", "distance": "8.1 km"}

]

}

#### **POST /api/analyze/image**

Analyzes a user-uploaded medical image.

* **Headers:** Authorization: Bearer <token>
* **Request Body:** multipart/form-data
  + image: The image file (e.g., .jpg, .png).
  + latitude: (optional) 12.8238
  + longitude: (optional) 80.2234
* **Success Response (200 OK):**

{

"analysisId": "analysis\_67890",

"timestamp": "2025-10-17T12:05:00Z",

"imageAnalysis": {

"identifiedCondition": "Eczema",

"confidenceScore": 0.89,

"description": "A common skin condition marked by itchy and inflamed patches of skin."

},

"recommendations": "Avoid irritants, moisturize regularly. Consult a dermatologist for prescription treatments.",

"nearbyHospitals": [

{"name": "Global Health City", "distance": "5.2 km"}

]

}

#### **GET /api/history**

Fetches all past analysis records for the authenticated user.

* **Headers:** Authorization: Bearer <token>
* **Success Response (200 OK):**

{

"history": [

{

"type": "text",

"data": { /\* ... text analysis response ... \*/ }

},

{

"type": "image",

"data": { /\* ... image analysis response ... \*/ }

}

]

}

## **4. 🚀 Deployment & Local Setup**

### **Frontend (React + Vite)**

* **Local Development:**

# Clone, install, and run

git clone [https://github.com/](https://github.com/)<your-username>/medilens-frontend.git

cd medilens-frontend

npm install

npm run dev

* **Deployment (Render):**
  + **Build Command:** npm ci && npm run build
  + **Publish Directory:** dist
  + **Environment Variable:** VITE\_API\_BASE = https://hsc1606.onrender.com

### **Backend (Python + Flask/FastAPI)**

* **Local Development:**

# Clone, create venv, install, and run

git clone [https://github.com/](https://github.com/)<your-username>/medilens-backend.git

cd medilens-backend

python -m venv venv

source venv/bin/activate

pip install -r requirements.txt

flask run

* **Deployment (Render):**
  + Deploy as a **Web Service**.
  + Connect the Git repository.
  + Render automatically detects requirements.txt and uses a Gunicorn server.

## **5. 👨‍💻 Author & Socials**

* **Name:** Ishaan Sharma
* **GitHub:** [https://github.com/Ishaan-1606]
* **LinkedIn:** [https://www.linkedin.com/in/ishaan-sharma-306b8a268/]
* **LeetCode:** [https://leetcode.com/u/1\_ishaan\_6/]

## **6. 📄 License**

© 2025 Ishaan Sharma.