

**SCTR's Pune Institute of Computer Technology  
(PICT) Pune**

**AN  
INTERNSHIP REPORT  
ON**

# **VitalsIQ - ML Health Analyzer**

**SUBMITTED BY**

Name: Vedant Vinayak Narawadkar

Class: TE - 07

Roll no: 32151

**Under the guidance of**

Dr. M. N. Kakade

(Asst. Prof. PICT, Pune)



**DEPARTMENT OF ELECTRONICS AND  
TELECOMMUNICATION ENGINEERING  
ACADEMIC YEAR 2024-25**



**DEPARTMENT OF ELECTRONICS AND  
TELECOMMUNICATION ENGINEERING**

**SCTR's Pune Institute of Computer Technology (PICT), Pune  
Maharashtra 411043**

**CERTIFICATE**

This is to certify that the internship report titled  
**"VitalsIQ - ML Health Analyzer"**

**Submitted by**  
***Vedant V. Narawadkar***  
***Roll No.: 32151***

has satisfactorily completed the curriculum-based internship under the guidance of *Dr. M. N. Kakade (Asst. Prof. PICT, Pune)* towards the partial fulfillment of third year Electronics and Telecommunication Engineering Semester VI, Academic Year 2024-25 of Savitribai Phule Pune University.

Internship Guide  
Dr. M. N. Kakade  
(Asst. Prof. PICT, Pune)

Internship Coordinator  
Ms. A. K. Patel

Head of the Department  
Dr. Mousami Munot  
(E&TC Engg.)

Place: PICT, Pune

Date: 17/04/25

# **Acknowledgement**

It gives me great pleasure in presenting the internship report on "VitalsIQ ML Health Analyzer." I would like to take this opportunity to thank my internship guide, Dr. M. N. Kakade, for giving me all the help and guidance needed. I am really grateful for their kind support and valuable suggestions that proved to be beneficial in the overall completion of this internship.

I am thankful to our Head of Electronics and Telecommunication Engineering Department, Dr. M. V. Munot, for her indispensable support and suggestions throughout the internship work. I would also genuinely like to express my gratitude to the Department Internship Coordinator, Ms. A. K. Patel, for her constant guidance and support and for the timely resolution of the doubts related to the internship process.

Finally, I would like to thank my mentor, Dr. M. N. Kakade, for their constant help and support during the overall internship process.

# Contents

|    |                                     |    |
|----|-------------------------------------|----|
| 1  | Title                               | 2  |
| 2  | Introduction                        | 2  |
| 3  | Problem Statement                   | 2  |
| 4  | Objectives and scope                | 4  |
| 5  | Methodological details              | 5  |
| 6  | Usage of Modern engineering tools   | 8  |
| 7  | Outcome/ results of internship work | 9  |
| 8  | Achievements                        | 13 |
| 9  | References                          | 14 |
| 10 | Completion Certificate              | 17 |

# 1. Title

## VitalsIQ - ML Health Analyzer

### 2. Introduction

**VitalsIQ** is a ground-breaking, machine learning-powered health analysis platform designed to revolutionize how people engage with their medical data. In an era where medical reports can feel like cryptic puzzles—loaded with jargon and numbers that leave most of us scratching our heads—**VitalsIQ** steps in as your personal health translator. This innovative web application leverages cutting-edge technologies to extract, process, and analyse health metrics, delivering personalized insights that are clear, actionable, and tailored to you. Whether you're monitoring your vitals, decoding a recent lab report, or looking to catch potential health risks early, **VitalsIQ** makes it effortless to stay in the driver's seat of your wellness journey.

The heart of **VitalsIQ** lies in its seamless integration of advanced OCR and AI technologies. Users start by uploading a scanned medical report or image—think blood work, lipid panels, or any standard health document. From there, Marker OCR gets to work, meticulously extracting key health parameters like cholesterol levels, blood sugar, or heart rate with pinpoint accuracy. This raw data is then handed off to the Gemini API 1, which structures and organizes it for deeper analysis. The real magic happens with Gemini API 2, which dives into the numbers, spotting trends, flagging potential risk factors, and generating health recommendations that feel like they're coming from a trusted doctor. All of this is presented through a sleek web interface, built with HTML, CSS, JavaScript, and powered by a robust Python Flask backend, ensuring a smooth and intuitive experience from start to finish.

What sets **VitalsIQ** apart is its mission to democratize health insights. It's not just for doctors or data nerds—it's for everyone. The platform breaks down barriers, making complex medical information accessible to anyone with a smartphone or computer. Imagine uploading your latest lab results and instantly understanding what's going well, what needs attention, and how to take the next steps. **VitalsIQ** doesn't just stop at analysis; it's built to evolve, with plans to expand its capabilities and reach. From supporting more health metrics to creating a mobile-friendly design, the future of **VitalsIQ** is about bringing AI-driven health monitoring to every corner of the globe.

Here's why **VitalsIQ** stands out:

- **User-Centric Design:** A web-based interface that's simple to navigate, whether you're tech-savvy or not.
- **AI-Powered Precision:** Combines Gemini APIs for accurate data processing and intelligent health assessments.
- **Scalable Vision:** Built to grow, with plans for advanced AI integrations and broader health parameter support.

### 3. Problem statement

Navigating medical reports is like trying to crack a code without a key—most people are left lost in a maze of technical jargon, confusing numbers, and stats that might as well be in another language. Whether it's a routine blood test or a detailed health panel, understanding key health metrics like cholesterol, glucose levels, or blood pressure is a struggle for the average person. This gap in health literacy means missed chances to catch potential risks early or make informed decisions about wellness. Worse, the tools out there? They're either built for doctors with PhDs, clunky to use, or just spit out raw data without any personalized guidance.

The result? People feel disconnected from their own health data, stuck relying on overbooked doctors or generic Google searches for answers. There's a glaring need for a user-friendly platform that can take complex medical reports, break them down into plain English, and deliver AI-powered insights that actually make sense. VitalsIQ steps up to fill this void, offering a seamless way to extract, analyse, and understand health parameters while empowering users with actionable recommendations to stay ahead of their health game.

## 4. Objectives and scope

### Objectives:

- Develop a user-friendly platform that simplifies the process of understanding medical reports for individuals without medical expertise.
- Leverage Marker OCR to accurately extract key health metrics from uploaded scanned reports or images.
- Utilize Gemini APIs to provide AI-driven insights, including health trends, risk assessments, and personalized recommendations.
- Empower users to take control of their wellness by delivering actionable health data in a clear, digestible format.
- Build a scalable system capable of supporting additional health parameters and advanced AI integrations in the future.
- Enhance health literacy by making complex medical information accessible to a diverse audience through an intuitive web interface.

### Scope

- Create a web-based application using HTML, CSS, JavaScript, and Python Flask for seamless user interaction and backend processing.
- Implement OCR-based extraction to process medical reports in formats like PDFs or images, focusing on standard health metrics (e.g., blood pressure, cholesterol, glucose).
- Integrate Gemini API 1 for structuring extracted data and Gemini API 2 for generating intelligent health assessments.
- Design three core pages: a Landing Page for introduction, an About Us Page for project context, and a Main Analysis Page for report uploads and results display.
- Lay the foundation for future enhancements, such as mobile-friendly design, expanded health parameter support, and deeper AI-driven analytics.
- Exclude real-time health monitoring or direct medical consultations, focusing solely on post-report analysis and insights generation.

## 5. Methodological details

### System Architecture:

VitalsIQ adopts a modular architecture to ensure scalability and maintainability:

- Frontend: Built with HTML, CSS, and JavaScript for a responsive, user-friendly web interface.
- Backend: Powered by Python Flask to handle API requests, OCR processing, and AI integrations.
- AI Pipeline: Integrates Marker OCR for text extraction and Gemini APIs for data structuring and health analysis.

### Backend Modules:

- `app.py`: Core Flask application managing API endpoints, file uploads, and request routing.
- `ocr_processor.py`: Interfaces with Marker OCR to extract health metrics from uploaded medical reports.
- `gemini_data_parser.py`: Uses Gemini API 1 to structure and clean extracted data for analysis.
- `gemini_insights.py`: Leverages Gemini API 2 to generate health insights, risk assessments, and recommendations.
- `utils.py`: Contains helper functions for data validation, file handling, and error management.

### Frontend Modules

- Reusable Components: Includes Header, Footer, and UploadForm for consistent UI design.
- Landing Page: Introduces VitalsIQ with an overview and call-to-action for report uploads.
- About Us Page: Displays project background and team information.
- Main Analysis Page: Manages file uploads, displays extracted metrics, and shows AI-generated insights.
- JavaScript Logic: Handles dynamic rendering, form submissions, and API responses using fetch for seamless interaction.

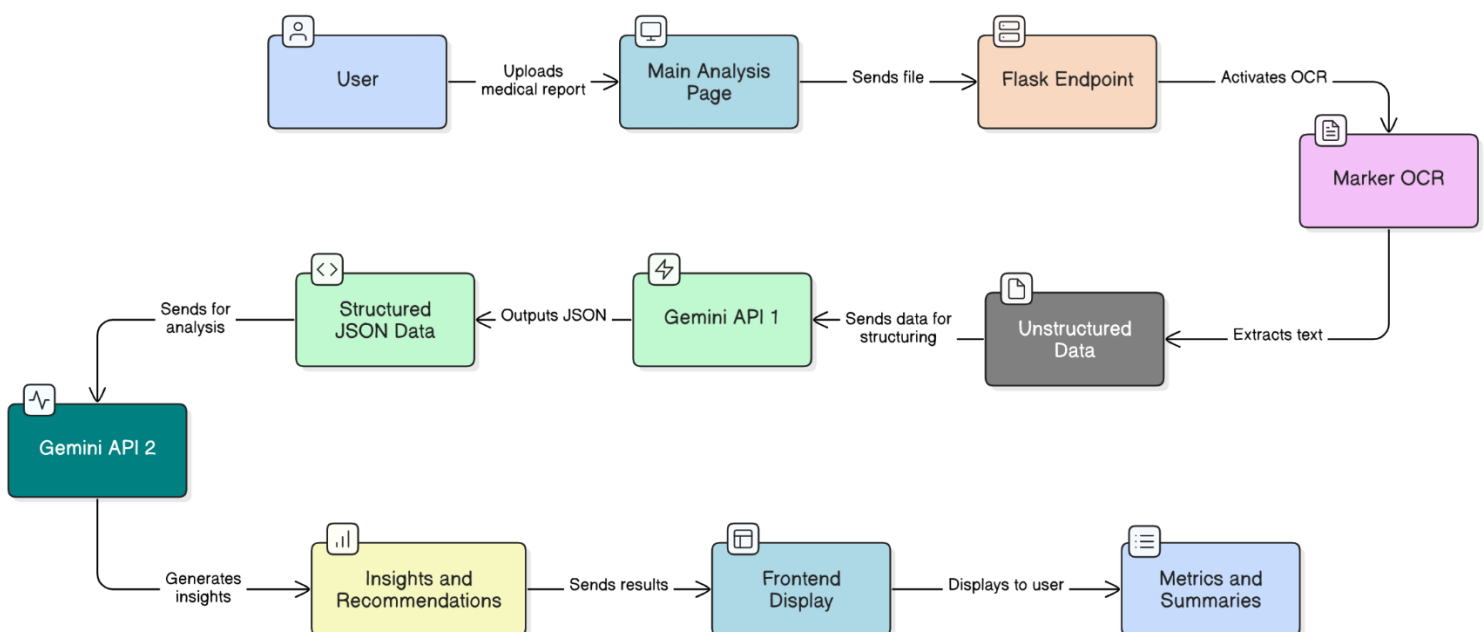


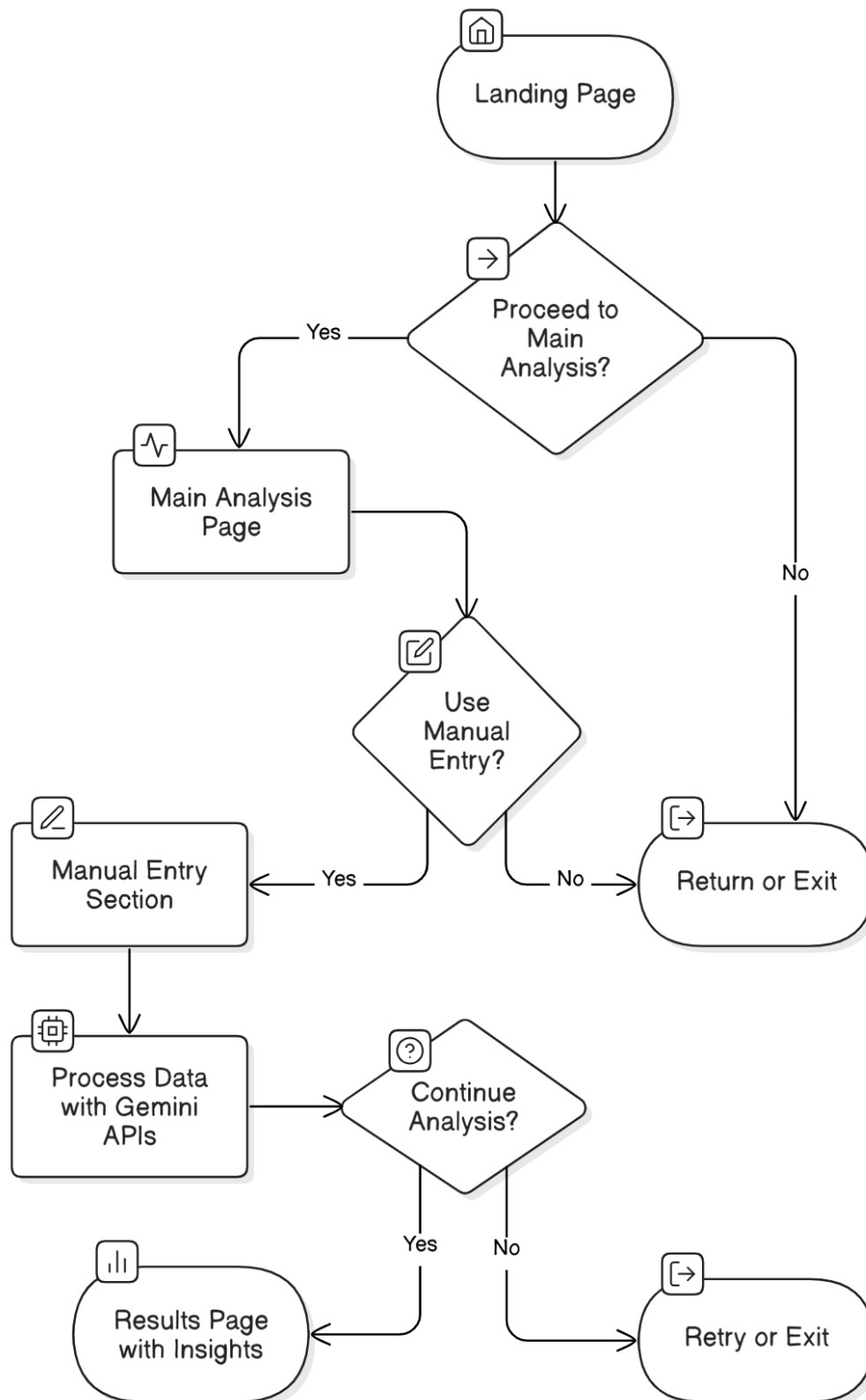
## API Flow

- User uploads a medical report via the Main Analysis Page.
- Flask endpoint receives the file and triggers Marker OCR for text extraction.
- Extracted data is sent to Gemini API 1 for structuring into a usable format.
- Structured data is processed by Gemini API 2 to generate health insights and recommendations.
- Results are returned to the frontend, rendered as visualized metrics and text summaries.

## Data Processing Design:

- Raw Input: Scanned PDFs or images of medical reports containing health metrics (e.g., blood pressure, cholesterol, glucose).
- Extracted Data: Unstructured text output from Marker OCR, including numerical values and labels.
- Structured Data: JSON-like format created by Gemini API 1, organizing metrics into categories (e.g., { "blood pressure": "120/80", "glucose": "90" }).
- Output Insights: Gemini API 2 results, including trend analysis, risk flags (e.g., "elevated glucose"), and health tips (e.g., "consider low-sugar diet").



*Fig 0: User Flowchart*

## 6. Usage of Modern engineering tools

| Tool         | Role  |
|--------------|---|
| Python       | Core programming language for backend logic and AI pipeline processing. |
| Flask        | Lightweight framework to define REST APIs and handle request routing.   |
| Marker OCR   | Extracts text from scanned medical reports for health metric analysis.  |
| Gemini APIs  | Powers AI-driven data structuring and health insights generation.       |
| JavaScript   | Enables dynamic frontend interactions and API communication.            |
| HTML, CSS    | Structures and styles the web interface for a seamless user experience. |
| pip          | Manages Python dependencies for smooth installation and updates.        |
| Postman      | API development and testing tool.                                       |
| Git & GitHub | Version control and source code management.                             |

## 7. Outcome/ results of internship work

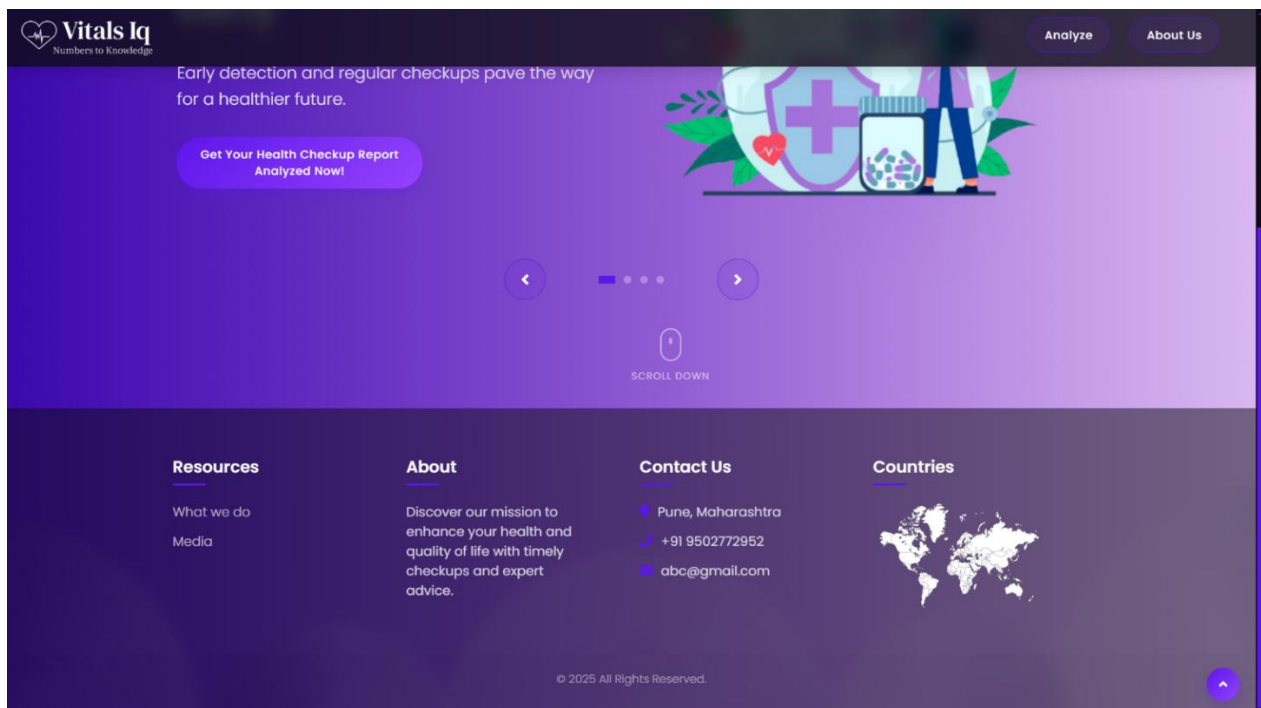
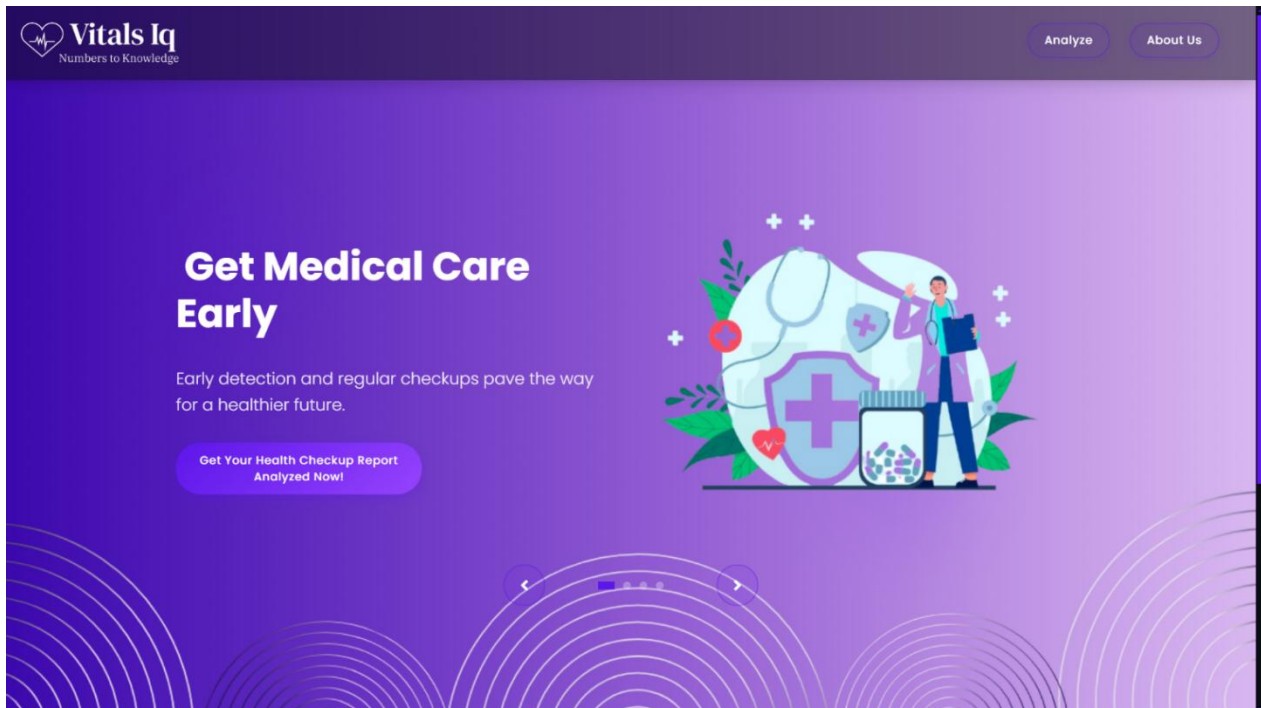


Fig. 1, 2: Landing Page

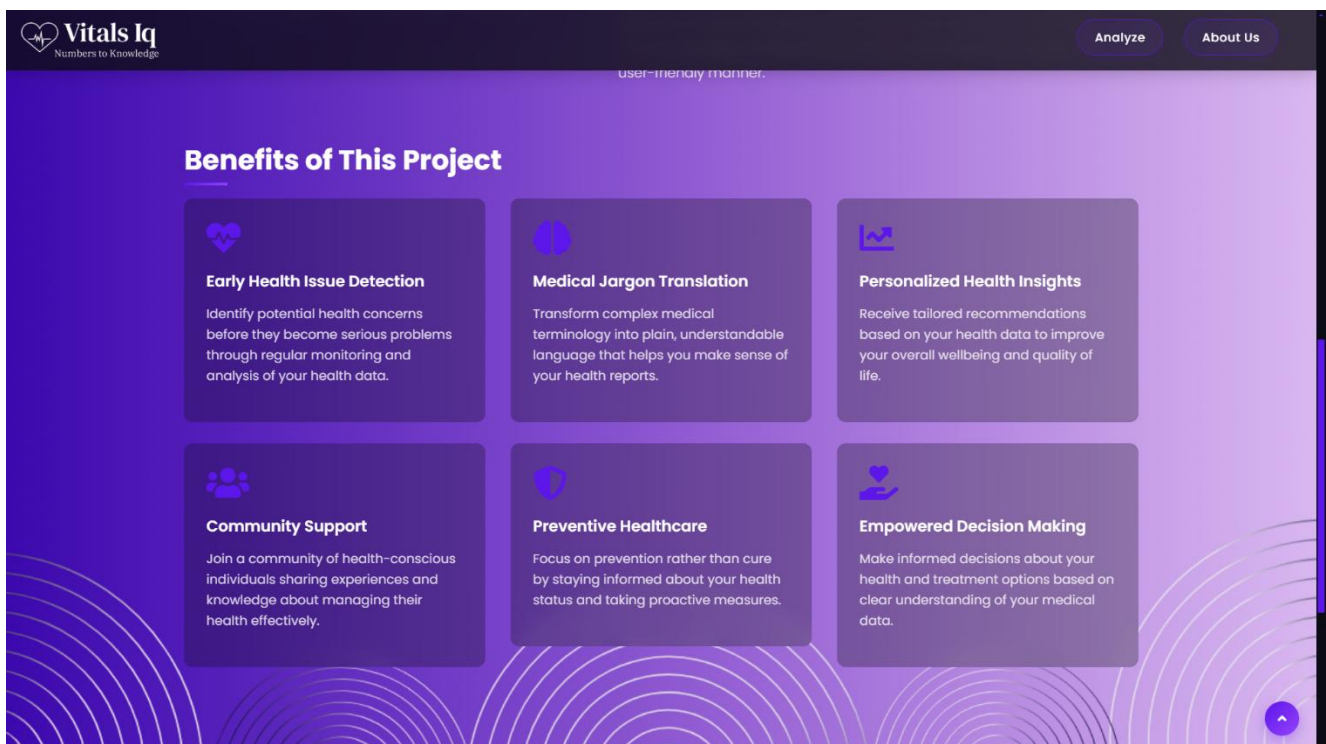
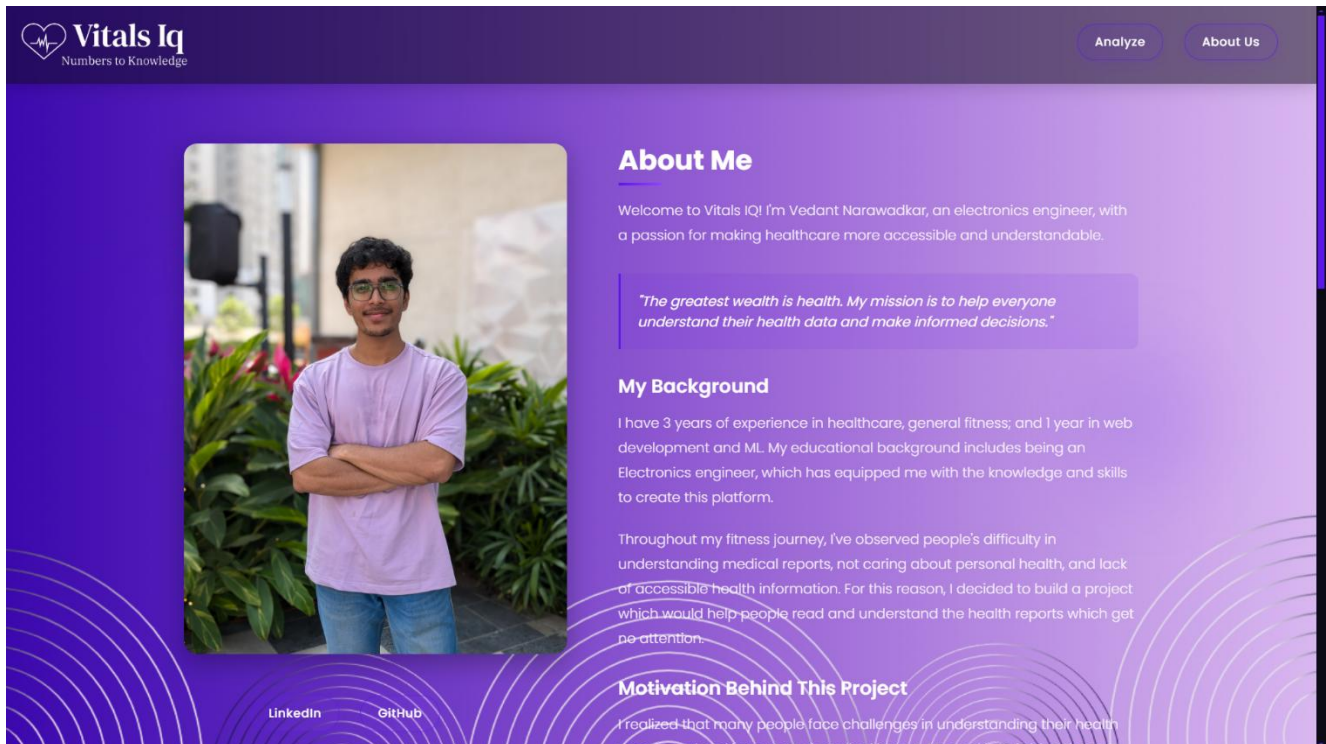



Fig. 3, 4: About Page






AnalyzeAbout Us

## Upload Your Health Report PDF


Easily upload your health report in PDF format to get a detailed analysis. Simply drag and drop your file into the box below or click to select a file from your device. Once uploaded, click "Upload & Analyze" to view your health insights on the next page.

### Upload Your PDF

  
Drag & Drop your PDF here or click to upload

Choose File

Upload & Analyze



AnalyzeAbout Us

## Enter Your Health Parameters Manually

If you don't have a PDF report, you can manually enter your health parameters below. Fill in the fields with your latest health metrics (e.g., blood pressure, glucose levels). Once you've entered your data, click "Analyze Manual Entry" to see your health analysis on the next page.

### Manual Entry Form

|   |  |
|---|--|
| Blood Pressure (mmHg):<br>e.g., 120/80  | HDL Cholesterol (mg/dL):<br>e.g., 50                   |
| Blood Glucose (mg/dL):<br>e.g., 90      | LDL Cholesterol (mg/dL):<br>e.g., 100                  |
| Hemoglobin (g/dL):<br>e.g., 13.5        | White Blood Cell Count (cells/ $\mu$ L):<br>e.g., 7000 |
| Total Cholesterol (mg/dL):<br>e.g., 180 | Platelets (cells/ $\mu$ L):<br>e.g., 250000            |

Analyze Manual Entry

Fig. 5, 6: Report Page

**Vitals Iq**  
Numbers to Knowledge

Analyze About Us

|                         |                       |
|-------------------------|-----------------------|
| Blood Pressure:         | 130/70 mmHg           |
| Blood Glucose:          | 121 mg/dL             |
| Hemoglobin:             | 11.1 g/dL             |
| Total Cholesterol:      | 180 mg/dL             |
| HDL Cholesterol:        | 70 mg/dL              |
| LDL Cholesterol:        | 110 mg/dL             |
| White Blood Cell Count: | 7800 cells/ $\mu$ L   |
| Platelets:              | 300700 cells/ $\mu$ L |

### Health Analysis

#### Overall Assessment

The provided parameters suggest some areas of concern, particularly regarding blood glucose and hemoglobin levels. While blood pressure and cholesterol levels are within a generally acceptable range, the elevated blood glucose suggests potential prediabetes. The low hemoglobin points towards mild anemia. Further investigation is recommended to confirm these findings and rule out any underlying health conditions.

#### Potential Supplementation Requirements

- **Iron Supplement:** If diagnosed with iron deficiency anemia, an iron supplement (e.g., ferrous sulfate, 325 mg daily) can help restore iron levels. Consult a doctor before starting any iron supplement to determine the appropriate dosage and duration.
- **Magnesium:** Magnesium plays a role in glucose metabolism and insulin sensitivity. A daily supplement of 200-400 mg may be beneficial for blood sugar control. Choose magnesium citrate or glycinate for better absorption.
- **High-Quality Multivitamin:** A multivitamin can help ensure adequate intake of essential micronutrients, supporting overall health and addressing potential deficiencies. Look for a multivitamin tailored to your age and gender.

#### Additional Recommendations

- **Regular Monitoring:** Schedule regular check-ups with your doctor to monitor blood glucose, hemoglobin, and cholesterol levels. This allows for early detection and management of any potential health issues.
- **HbA1c Test:** Request an HbA1c test, which measures average blood glucose levels over the past 2-3 months, providing a more comprehensive picture of glucose control.
- **Further Anemia Evaluation:** Consult a doctor to determine the underlying cause of the low hemoglobin. Further blood tests may be necessary to assess iron stores, vitamin B12 levels, and other potential factors.

Regenerate

Copy Results Download PDF

Fig. 7, 8: Report &amp; Analysis Page

## 8. Achievements

The development of VitalsIQ showcased the successful application of modern AI and web development techniques, blending OCR technology, Gemini APIs, and a Flask-based backend to create a health analysis platform. Throughout the project, the following technical and learning milestones were achieved:

### 1. AI-Powered Health Analysis System

- Designed an end-to-end system to process medical reports and deliver health insights.
- Integrated Gemini APIs for intelligent data structuring and risk assessment.
- Enabled users to understand complex health metrics without medical expertise.

### 2. Robust OCR Integration

- Implemented Marker OCR to extract health parameters from scanned reports.
- Handled diverse report formats, ensuring accurate text recognition.
- Optimized extraction pipeline for reliability across varying image qualities.

### 3. Seamless Web Application Development

- Built a user-friendly interface using HTML, CSS, and JavaScript.
- Developed a Flask backend to manage file uploads and API workflows.
- Created three distinct pages for landing, about, and analysis functionalities.

### 4. Efficient API Workflow

- Designed RESTful endpoints for report uploads and results delivery.
- Connected frontend with backend for smooth data flow and rendering.
- Tested APIs using Postman to ensure consistent performance and error handling.

### 5. Modular Code Structure

- Organized backend into separate modules for OCR, AI processing, and utilities.
- Structured frontend with reusable components for maintainability.
- Ensured clean code practices for easier debugging and future updates.



## 6. User-Centric Health Insights

- Delivered personalized recommendations based on analyzed health data.
- Presented results in a clear, visual format for non-technical users.
- Focused on accessibility to empower diverse audiences with health knowledge.

## 7. Scalable System Foundation

- Laid groundwork for adding more health metrics in future iterations.
- Built with extensibility to support advanced AI models and integrations.
- Prepared codebase for potential mobile-friendly enhancements.

## 8. Enhanced Problem-Solving Skills

- Tackled challenges like inconsistent OCR outputs and API rate limits.
- Debugged data parsing issues to ensure accurate health metric extraction.
- Developed proactive solutions using logging and testing tools.

## 9. Version Control Mastery

- Utilized Git and GitHub for efficient code versioning and collaboration.
- Maintained clear commit messages to track project progress.
- Managed repository to support iterative development and updates.

# 9. References

[1] Python Official Documentation

<https://docs.python.org/3/>

Used for understanding Python syntax, file handling, and backend logic implementation for VitalsIQ.

[2] Flask Documentation

<https://flask.palletsprojects.com/en/stable/>

Referred for setting up RESTful APIs, routing, and handling file uploads in the backend.

[3] Marker OCR Documentation

<https://marker.inc/docs/>

Used to integrate and configure Marker OCR for extracting health metrics from medical reports.

[4] Google Gemini API Documentation

[https://ai.google.dev/docs/gemini\\_api](https://ai.google.dev/docs/gemini_api)

Referred for implementing Gemini APIs for data structuring and generating AI-driven health insights.

[5] JavaScript (MDN Web Docs)

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Used for learning JavaScript fundamentals, DOM manipulation, and frontend interactivity.

[6] HTML5 and CSS3 (MDN Web Docs)

<https://developer.mozilla.org/en-US/docs/Web/HTML>

<https://developer.mozilla.org/en-US/docs/Web/CSS>

Referred for building and styling the web interface, including the Landing, About Us, and Analysis pages.

[7] Git and GitHub Documentation

<https://git-scm.com/doc>

<https://docs.github.com/en>

Used for version control, repository management, and collaborative development.

[8] pip Documentation

<https://pip.pypa.io/en/stable/>

Referred for managing Python dependencies and ensuring smooth installation of project requirements.

[9] Postman API Platform

<https://www.postman.com/product/api-client/>

Used for testing and validating Flask API endpoints during backend development.

[10] Visual Studio Code Documentation

<https://code.visualstudio.com/docs>

Served as the primary code editor for writing Python, JavaScript, HTML, and CSS code.

[11] W3Schools Web Tutorials

<https://www.w3schools.com/>

Supplementary resource for HTML, CSS, and JavaScript syntax and frontend design practices.

[12] Stack Overflow

<https://stackoverflow.com/>

Used to troubleshoot issues like OCR parsing errors, API integration challenges, and Flask debugging.

[13] YouTube Tutorials and Guides

Channels such as:

- a. freeCodeCamp.org
- b. Corey Schafer
- c. Tech With Tim

Provided practical tutorials on Flask development, OCR integration, and AI API usage.

## 10. Completion Certificate

