AI MEDICAL REPORT GENERATION

PROJECT REPORT 2025-2026

Submitted By

24500522 MORDEEP DHARSHAN S

24500426 ROHITSHARMA V

24500431 SIBIRAJ S

24500421 MANIKANDAN M

Under the guidance of

Mrs. C.SRIDEVI B.E.,(CSE)

HOD - COMPUTER & IT

Diploma in Information Technology

of the directorate of Technical Education, Government of Tamil Nadu

DEPARTMENT OF INFORMATION TECHNOLOGY

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202.

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202

Department of Information Technology

BONAFIDE CERTIFICATE

This is certified that this project work entitled **AI MEDICAL REPORT GENERATION** has been submitted **MORDEEP DARSHAN S, ROHITSHARMA V, SIBIRAJ S, MANIKANDAN M** in the partial fulfilment of the requirements for the award of Diploma in Information Technology during the academic year 2025-2026, who carried out the project work under our supervision.

Project Guide	**Head of the Department**
Mrs. C.SRIDEVI B.E.,(CSE)	**Mrs. C.SRIDEVI B.E.,(CSE)**
HOD-COMPUTER & IT	**HOD-COMPUTER & IT**

This is to certify that_____ was examined for the project work viva-voce held on_____

INTERNAL EXAMINER EXTERNAL EXAMINER

Acknowledgement

We express our sincere and profound thanks to our chairman **Mr.A.K.T.Mahendiran B.Com.,** for creating this magnificent edifice of learning by providing all necessary facilities to complete diploma engineering course successfully.

We express our gratitude to our principal **Mr.P.K.Kabilar M.E.,MBA.,** for constant encouragement and facilities provided towards the successful completion of our project work.

At the same time we would like to express our heartfelt thanks to our head of the department **Mrs.C.Sridevi.,B.E.,(CSE).,** for guiding and needful advices and time to complete this project.

We would like to show our gratitude to our department staffs for all instructions and guidance given throughout.

Our sincere thanks and affection to our parents and friends who gave hand in all our steps.

CONTENTS

ABSTRACT

The AI Medical Report System is an intelligent healthcare management platform designed to automate the extraction, storage, and analysis of patient discharge summaries using advanced artificial intelligence technologies. By leveraging OpenAI-powered image and text processing, the system accurately extracts critical medical information such as patient details, diagnosis, prescribed medications, doctor information, hospital data, and follow-up instructions from uploaded documents.

The platform features a professional admin dashboard that enables efficient patient management, report tracking, and centralized data storage. It supports automated scheduling and alert mechanisms for follow-up visits, ensuring timely patient care and reducing the risk of missed consultations.

Additionally, the system provides AI-driven recommendations, including personalized food plans, activity guidelines, and medication advice tailored to each patient's condition. This enhances patient recovery and promotes better adherence to post-discharge instructions.

Built using PHP and MySQL, the system ensures scalability, security, and ease of integration into existing healthcare workflows. Overall, the AI Medical Report System improves operational efficiency, minimizes manual data entry, and enhances the quality of patient care through intelligent automation and data-driven insights.

1. INTRODUCTION

The rapid advancement of artificial intelligence has significantly transformed various industries, including healthcare. Managing patient discharge summaries and medical records manually is often time-consuming, error-prone, and inefficient. Healthcare providers frequently face challenges in extracting critical information from unstructured documents, tracking patient follow-ups, and ensuring adherence to post-discharge care instructions.

The AI Medical Report System is designed to address these challenges by providing an intelligent, automated solution for processing and managing medical data. By utilizing OpenAI-powered image and text extraction technologies, the system can accurately capture essential details from discharge summaries, such as patient information, diagnosis, medications, doctor details, and hospital records.

This system integrates a professional dashboard that enables healthcare administrators to efficiently manage patient records, monitor follow-up schedules, and generate alerts for upcoming visits. Furthermore, it enhances patient care by offering AI-driven recommendations, including dietary guidance, activity suggestions, and medication advice tailored to individual health conditions.

Developed using PHP and MySQL, the platform ensures a scalable, secure, and user-friendly environment for healthcare management. By automating repetitive tasks and providing intelligent insights, the AI Medical Report System aims to reduce manual workload, improve data accuracy, and support better clinical decision-making.

2. LITERATURE SURVEY

The healthcare sector has increasingly adopted digital technologies to improve patient data management, reduce manual errors, and enhance the overall quality of care. Traditional methods of handling discharge summaries and medical records are largely paper-based or semi-digital, making data retrieval, analysis, and follow-up management difficult. Several studies and existing hospital information systems highlight the importance of digitizing clinical documents for better accessibility and decision-making.

Optical Character Recognition (OCR) has been widely used in medical document processing to convert scanned records into editable text. However, conventional OCR systems often face limitations when dealing with handwritten notes, complex medical terminology, or poorly formatted reports. Recent advancements in artificial intelligence and natural language processing have significantly improved the ability to extract meaningful medical information from unstructured documents. AI-based document understanding systems can now identify patient details, diagnoses, prescriptions, and follow-up instructions with greater accuracy than rule-based approaches.

Research in medical informatics also emphasizes the role of Electronic Health Record (EHR) systems in improving patient care. EHR platforms provide structured storage of patient history, treatment details, laboratory results, and discharge instructions. Despite these advantages, many existing systems focus mainly on record storage and lack intelligent features such as automated advice generation, follow-up reminders, and personalized patient guidance. This creates a gap between data storage and practical patient support.

Artificial intelligence has recently been applied in clinical decision support systems to assist doctors and healthcare staff by generating recommendations based on patient symptoms, diagnoses, and treatment history. Machine learning and large language models have shown promising results in interpreting medical text, summarizing reports, and producing patient-friendly advice. These technologies are especially useful in post-discharge care, where patients often need clear instructions related to food, medication, activity restrictions, and future consultations.

Several patient management systems currently provide appointment scheduling and reminder functionalities, but they often require manual data entry and do not automatically interpret discharge summaries. Similarly, some healthcare applications offer diet and lifestyle suggestions, but they are not directly integrated with hospital discharge documents. Therefore, there is a growing need for a unified system that can combine medical report extraction, patient record management, follow-up tracking, and AI-driven recommendations within a single platform.

Based on the review of existing methods and technologies, it is evident that integrating OpenAI-powered text extraction and intelligent advisory features into a medical report management system can significantly improve efficiency and patient care. The proposed AI Medical Report System aims to bridge the limitations of existing solutions by automating discharge summary extraction, storing complete medical details in a structured database, and generating useful post-discharge advice through a professional dashboard.

2.1 Problem Statement

In modern healthcare environments, patient discharge summaries contain critical information such as diagnosis, medications, doctor details, hospital information, and follow-up instructions. However, these documents are often stored as unstructured text or scanned images, making it difficult to efficiently extract, organize, and utilize the data. Manual handling of such records is time-consuming, error-prone, and leads to inefficiencies in patient management.

Healthcare providers frequently face challenges in tracking patient history, scheduling follow-up visits, and ensuring that patients adhere to post-discharge instructions. Important details such as next visit dates, dietary restrictions, medication schedules, and activity guidelines are often overlooked or not properly communicated, which can negatively impact patient recovery and increase the risk of readmission.

Existing hospital management and electronic health record systems primarily focus on data storage rather than intelligent processing. They lack automated mechanisms to extract key information from discharge summaries and do not provide personalized, AI-driven recommendations for patients. Additionally, many systems require manual data entry, increasing the workload for healthcare staff and the possibility of human errors.

There is a clear need for an intelligent, automated solution that can accurately extract and structure medical information from discharge summaries, manage patient records efficiently, and provide timely follow-up alerts and personalized care recommendations. The proposed AI Medical Report System aims to address these challenges by integrating OpenAI-powered extraction, centralized data management, and AI-driven advisory features into a single, user-friendly platform.

3. PROPOSED SYSTEM

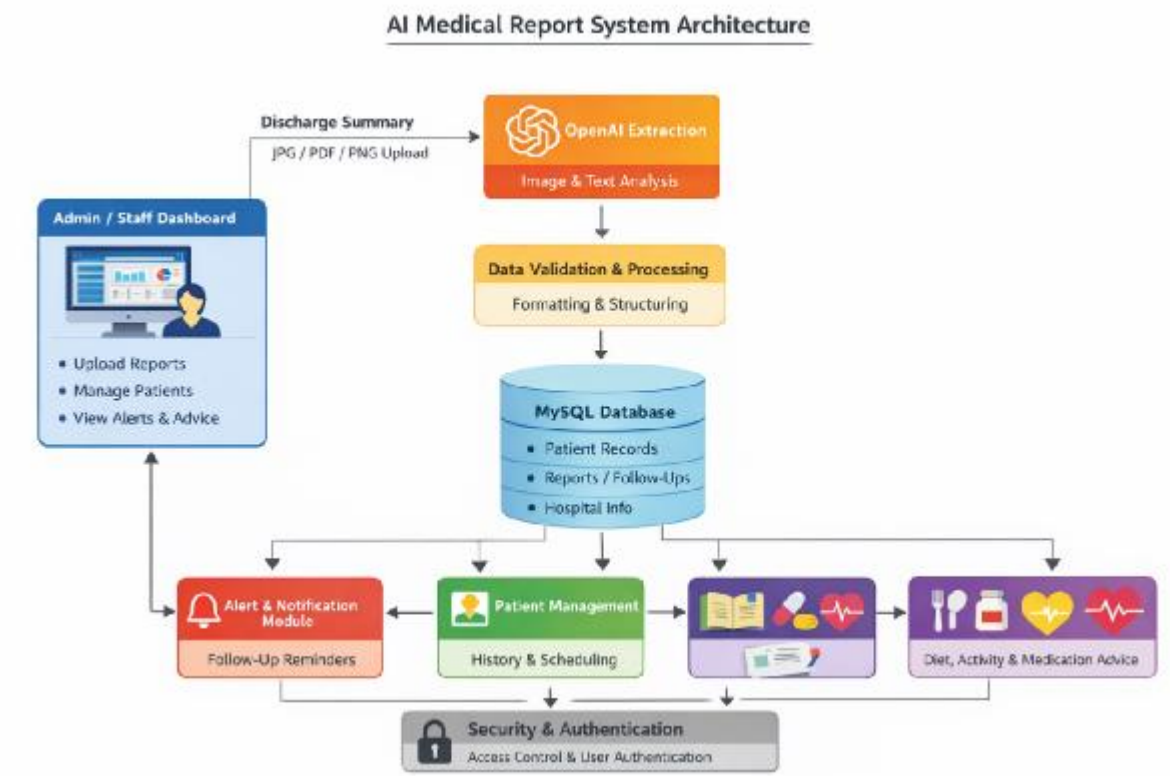
The proposed AI Medical Report System is designed to automate the extraction, storage, and intelligent use of patient discharge summary data. The system accepts discharge summaries in image or document form and processes them using OpenAI-powered extraction to identify important medical details such as patient name, age, gender, diagnosis, doctor name, hospital name, medicines, discharge date, follow-up date, food advice, and activity instructions.

After extraction, the data is validated and stored in a structured MySQL database. This enables fast retrieval of patient records, easy tracking of discharge history, and efficient management of follow-up care. The system provides a professional admin dashboard through which healthcare staff can upload reports, review extracted data, manage patients, monitor follow-up alerts, and access AI-generated recommendations.

The proposed system also includes intelligent advisory modules that generate patient-specific suggestions based on the discharge summary. These include medication reminders, dietary recommendations, activity precautions, and next-visit alerts. By combining document intelligence with patient management features, the system reduces manual work, improves accuracy, and enhances post-discharge healthcare support.

The system is developed using PHP for backend processing, MySQL for database management, HTML, CSS, JavaScript, and Bootstrap for the user interface, and OpenAI API for image and text understanding. It offers a secure, scalable, and user-friendly solution suitable for hospitals, clinics, and healthcare administrators.

3.1 ARCHITECTURE



The architecture of the AI Medical Report System consists of multiple interconnected layers that work together to process discharge summaries and deliver intelligent patient management.

1. User Interface Layer

This layer provides the professional dashboard for administrators and staff. It allows users to log in, upload discharge summaries, manage patient records, view extracted data, monitor alerts, and access AI-generated advice. The interface is designed with a left-side navigation menu and dashboard cards for quick access to key modules.

2. Document Upload Layer

In this layer, the user uploads patient discharge summaries in image or document format. The system securely receives the file and stores it temporarily for processing. Supported files may include JPG, PNG, and PDF formats.

3. AI Extraction Layer

This is the core intelligence layer of the system. Uploaded reports are sent to the OpenAI-powered extraction engine, which analyzes the content and extracts structured medical data. The system identifies patient details, diagnosis, medicines, doctor details, hospital information, discharge instructions, and follow-up dates from unstructured medical text or scanned images.

4. Data Processing and Validation Layer

Once the information is extracted, the system processes and validates the data before storing it. This layer checks for missing fields, formats dates, standardizes medicine names, and organizes extracted text into structured categories.

5. Database Layer

The validated data is stored in a MySQL database. Separate tables maintain information related to patients, medical reports, doctors, hospitals, medicines, follow-up schedules, food advice, activity guidance, and system users. This structured storage allows efficient querying, editing, and reporting.

6. Application Logic Layer

This layer handles the main system operations such as patient management, report history, alert generation, follow-up scheduling, and advice retrieval. It connects the frontend dashboard with the database and AI modules using PHP-based business logic.

7. Alert and Notification Layer

This layer monitors follow-up dates and medication schedules. It generates reminders and alerts for upcoming patient visits, ensuring continuity of care and timely medical attention.

8. AI Recommendation Layer

Based on the extracted summary details, this layer generates personalized food advice, medication guidance, activity recommendations, and general health precautions. These outputs help patients better understand their recovery plan.

9. Security and Authentication Layer

This layer manages login authentication, session control, role-based access, and secure report handling. It ensures that only authorized users can access patient information and system modules.

3.2 MODULES DESCRIPTION

1. User Authentication and Role Management Module

This module ensures that only authorized users can access the system. It provides secure login, logout, session handling, password protection, and role-based access control. Different users such as Admin, Doctor, Reception Staff, and Data Entry Operator can be given different permissions based on their responsibilities. For example, an admin can manage all records and users, while a staff member may only upload reports and view patient details.

This module is very important because medical data is highly sensitive and must be protected from unauthorized access. It also improves accountability, since every action performed in the system can be linked to a logged-in user. Features such as password hashing, session timeout, forgot password, user activity log, and access restrictions can be included to make the application more secure and professional.

2. Professional Dashboard Management Module

The dashboard module acts as the main control center of the entire system. After login, users are directed to a professional dashboard with a left-side navigation menu and summary widgets. The dashboard displays key information such as total patients, total uploaded reports, today's follow-ups, pending alerts, recently extracted summaries, and AI advice generated.

This module helps administrators quickly understand the status of the system without manually checking each section. Graphs, status cards, recent activities, and quick-action buttons can be used to improve usability. A professional dashboard saves time and increases efficiency by giving users direct access to major modules from one screen.

3. Discharge Summary Upload Module

This module allows medical staff or administrators to upload patient discharge summaries into the system. The uploaded files may be in formats such as JPG, PNG,

JPEG, or PDF. The module validates the file type, file size, and file integrity before saving it to the server. It also connects each uploaded file with the respective patient record or creates a new patient entry if needed.

The discharge summary upload module is the starting point of the entire workflow. Once a summary is uploaded, it is prepared for AI extraction. The system may also maintain original file storage for future reference, report auditing, and re-processing. Additional features like drag-and-drop upload, preview before upload, upload progress bar, and duplicate file detection can make this module more effective and user-friendly.

4. OpenAI-Powered Report Extraction Module

This is the core intelligence module of the AI Medical Report System. It uses OpenAI-based image and text understanding to read and interpret the discharge summary. Once a report is uploaded, this module analyzes its contents and extracts key medical details such as patient name, age, gender, UHID, admission date, discharge date, diagnosis, symptoms, doctor name, hospital name, prescribed medicines, follow-up date, food instructions, and activity restrictions.

Unlike traditional manual data entry, this module reduces effort and improves speed by automatically identifying structured data from unstructured medical documents. It can handle complex discharge summaries and produce readable structured output. This module is highly valuable because it transforms raw hospital documents into usable digital data. It directly supports automation, reduces human errors, and forms the foundation for all other modules.

5. Data Validation and Structuring Module

After the AI extraction process is completed, the extracted information may still require validation and formatting before being stored permanently. This module checks whether all essential fields are available, whether the extracted date formats are valid, whether

medicine details are properly separated, and whether follow-up dates are readable and usable.

This module improves the quality of the system by cleaning and standardizing extracted data. For example, it can convert different date styles into one standard database format, remove unwanted characters, identify missing mandatory details, and organize free-text content into structured categories. It also allows manual review and correction when extraction confidence is low. This ensures that the stored information is accurate, clean, and ready for patient management and advisory functions.

6. Patient Registration and Patient Management Module

This module maintains a complete digital profile of every patient in the system. It stores patient information such as name, age, gender, mobile number, address, medical ID, discharge history, diagnosis history, and linked discharge summaries. It allows administrators to add new patients, edit patient details, search patient records, and view all past reports under one profile.

This module is essential for long-term patient care and medical record continuity. Instead of treating each uploaded discharge summary as an isolated file, the system connects all reports to a patient-centered record. It makes it easier to track recovery progress, repeated admissions, previous diagnoses, and recurring treatments. Search filters, patient profile pages, report timeline view, and quick access to follow-up status can be included as advanced features.

7. Doctor and Hospital Information Management Module

This module stores and manages details related to the doctor, department, and hospital mentioned in the discharge summary. It captures data such as doctor name, specialization, department, hospital name, location, hospital contact details, and report

association. This module helps the system maintain proper medical context for each patient report.

The doctor and hospital information module is useful for organizing clinical records and improving administrative tracking. For example, reports can be filtered based on hospital, department, or consulting doctor. This becomes highly useful when the system is used in multi-hospital or multi-doctor environments. It also improves reporting, record verification, and communication with the healthcare provider if needed.

8. Medication Management and Prescription Tracking Module

This module manages all medication details extracted from the discharge summary. It stores the prescribed medicine name, dosage, frequency, duration, route of administration, and special notes. If multiple medicines are prescribed, the system stores each one separately in linked records for better management.

This module helps both administrators and patients maintain proper medication history. It can be used to generate medication advice, reminders, and prescription history reports. It also supports identifying long-duration medicines, repeated medications across different visits, and doctor-specific prescriptions. This module is especially important for post-discharge care because incorrect handling of medicines can affect patient recovery.

9. Follow-Up Alert and Reminder Module

This module is designed to track follow-up dates mentioned in the discharge summary and generate timely reminders. Once the next visit date is extracted and stored, the system monitors upcoming follow-ups and displays alerts on the dashboard. It can show reminders such as follow-up due today, due tomorrow, overdue follow-ups, and upcoming consultations.

This module improves patient continuity of care and reduces the chance of missed follow-up visits. Alerts can be highlighted on the dashboard, listed in a separate reminder page, or even extended to SMS or email notifications in future versions. This feature is highly useful in hospitals and clinics where staff need to monitor multiple patients and ensure that discharge instructions are being followed properly.

10. AI-Driven Food Advice Module

This module generates personalized dietary recommendations based on the patient's diagnosis, discharge instructions, and recovery condition. After reading the discharge summary, the system can provide food guidance such as recommended foods, foods to avoid, hydration suggestions, light diet advice, or disease-specific nutritional precautions.

This module adds intelligence to the system by converting technical discharge instructions into simple, understandable patient guidance. For example, if a patient has diabetes, hypertension, post-surgery recovery, or gastric issues, the system can generate suitable food advice accordingly. This helps patients follow a better recovery plan and reduces confusion after discharge. The generated advice can be displayed in the patient profile, printed in reports, or shown in the dashboard.

11. AI-Driven Activity and Lifestyle Advice Module

This module provides personalized suggestions regarding physical activity, rest, mobility, exercise limitations, and lifestyle precautions based on the discharge summary. It may include advice such as bed rest duration, walking limits, lifting restrictions, wound care precautions, sleep guidance, and recovery-based activity planning.

Many patients do not fully understand the physical restrictions mentioned in medical reports. This module simplifies those instructions and presents them in a patient-

friendly format. For example, after surgery or serious illness, the system can advise limited physical movement, gradual walking, proper rest, or avoidance of strenuous work. This module improves the usefulness of the system by helping patients recover safely and follow the doctor's instructions more clearly.

12. Reports, History, and AI Advice Management Module

This module manages the complete history of uploaded reports, extracted summaries, follow-up records, and AI-generated recommendations. It allows users to view old discharge summaries, compare current and previous reports, download extracted records, and review advice generated for each patient. It acts as a central archive for all system-generated and uploaded medical content.

This module is useful for auditing, analysis, and future reference. It helps doctors, hospital staff, and administrators understand a patient's medical history over time. It also supports reporting features such as patient-wise report history, doctor-wise case history, month-wise discharge count, and pending follow-up reports. In a professional healthcare system, maintaining historical visibility is very important, and this module provides that support.

4. IMPLEMENTATION

The implementation of the AI Medical Report System is carried out using a combination of web technologies, database management, and artificial intelligence services to create a complete and practical healthcare application. The system is developed using PHP for server-side programming, MySQL for data storage, HTML, CSS, JavaScript, and Bootstrap for designing the user interface, and OpenAI API for extracting and understanding information from discharge summaries.

The implementation begins with the development of a secure login system for administrators and staff users. This includes user authentication, session handling, password encryption, and role-based dashboard access. Once the user logs in successfully, the system displays a professional dashboard with a left-side navigation menu to access modules such as report upload, patient management, follow-up alerts, and AI advice.

The next stage of implementation focuses on the discharge summary upload feature. The user uploads a discharge summary in image or PDF format through the dashboard. The system validates the file type, file size, and format before storing it on the server. After successful upload, the file is sent to the AI extraction module, where OpenAI-powered processing reads the contents of the report and identifies important details such as patient name, age, diagnosis, doctor name, hospital name, medicines, discharge date, and follow-up date.

Once the medical details are extracted, the data validation and structuring module processes the extracted content. This step ensures that the output is clean, formatted, and stored in the correct database fields. Invalid dates, missing values, and improperly extracted medicine details are corrected or flagged for review. After validation, the information is inserted into the MySQL database across related tables such as patients, reports, doctors, hospitals, medications, follow-ups, and advice records.

The patient management module is then implemented to allow administrators to search, edit, and manage patient profiles. Each patient record is linked with uploaded discharge summaries, extracted diagnosis details, prescriptions, and follow-up history. The doctor and hospital module is integrated to store and manage clinical references from the extracted reports.

The follow-up reminder module is implemented to compare stored follow-up dates with the current date and generate alerts for upcoming or overdue patient visits. These alerts are displayed on the dashboard so that staff can take timely action. In future expansion, this module can also be integrated with SMS or email notifications.

The AI advice modules are implemented to generate personalized recommendations based on the extracted diagnosis and discharge instructions. Using OpenAI, the system produces patient-specific food advice, activity guidance, medication-related precautions, and general recovery suggestions. These recommendations are stored in the database and displayed in the patient's detail page.

For frontend implementation, Bootstrap is used to build a responsive and professional user interface. The dashboard includes cards, tables, forms, modals, and navigation menus for easy interaction. JavaScript is used for dynamic behaviors such as live search, alert display, file preview, and form validation.

The backend implementation in PHP handles routing, database interaction, business logic, API communication, session control, and report generation. MySQL tables are properly normalized to reduce redundancy and improve data consistency. The system can also include audit logs and status tracking for uploaded files and generated advice.

Security is a major part of the implementation. Uploaded files are validated and stored securely, user passwords are encrypted, sessions are protected, and database queries are executed using prepared statements to prevent SQL injection. Access to patient data is restricted based on user roles.

Overall, the implementation of the AI Medical Report System transforms manual discharge summary handling into an intelligent digital workflow. It combines secure record management, automated data extraction, patient monitoring, and AI-driven post-discharge guidance into one integrated platform. This implementation improves efficiency, reduces human effort, and enhances the quality of healthcare administration.

4.1 SOFTWARE ENVIRONMENT

Category	Technology	Description
Frontend	HTML	Used to create the structure of web pages such as login, dashboard, and forms
Frontend	CSS3	Used for styling, layout design, colors, and responsive UI
Frontend	Bootstrap	Provides responsive design and ready-made components like sidebar, cards, tables
Frontend	JavaScript	Used for dynamic features like form validation, alerts, and real-time interactions
Backend	Core PHP	Handles server-side logic including authentication, file uploads, and system operations
Database	MySQL	Stores patient data, reports, doctors, medicines, follow-ups, and AI advice
API	OpenAI API	Extracts data from discharge summaries and generates AI-based medical advice
Cloud Hosting	VPS	Hosts the application online with high performance, scalability, and uptime
Server Management	WHM	Used for server-level management such as account creation and resource control
Server Management	cPanel	Provides easy interface for file management, database setup, and deployment
Messaging API	WhatsApp Official API	Sends follow-up reminders, alerts, and patient notifications in real-time
Web Server	Apache	Handles HTTP requests and serves the web application
Development Tool	XAMPP	Local development environment for testing using Apache and MySQL

Security	Session Management	Maintains user login sessions securely
Security	Password Hashing	Encrypts user passwords to ensure data security
Security	Prepared Statements	Prevents SQL injection attacks in database queries

4.2 SYSTEM REQUIREMENTS

Component	Specification	Description
CPU	8 Core Processor	Ensures high performance for handling multiple users and AI processing tasks
RAM	32 GB	Supports smooth execution of backend processes, database operations, and API calls
Storage	300 GB NVMe SSD	Provides fast data access and storage for reports, images, and database records
Network	High-Speed Internet	Required for API communication (OpenAI, WhatsApp) and real-time access
Operating System	Linux (Ubuntu/CentOS)	Stable and secure environment for hosting web applications
Web Server	Apache / Nginx	Handles incoming requests and serves the application
Control Panel	WHM / cPanel	Simplifies server and hosting management

4.3 SAMPLE CODING

Dashboard

```
<?php
require_once __DIR__ . '/../app/bootstrap.php';
require_login();

$repo = new PatientRepository();
$stats = $repo->dashboardStats();
$patients = $repo->recentPatients(8);

$pageTitle = 'Dashboard';
$pageSubtitle = 'Overview of patients, reports, alerts, and follow-ups';

include BASE_PATH . '/app/views/layout/header.php';
include BASE_PATH . '/app/views/layout/sidebar.php';
include BASE_PATH . '/app/views/layout/topbar.php';
?>
<div class="cards">
  <div class="card stat-card"><h3>Total Patients</h3><div class="stat"><?= $stats['patients'] ?></div></div>
  <div class="card stat-card"><h3>Discharge Reports</h3><div class="stat"><?= $stats['reports'] ?></div></div>
  <div class="card stat-card"><h3>Pending Alerts</h3><div class="stat"><?= $stats['alerts'] ?></div></div>
  <div class="card stat-card"><h3>Follow-ups</h3><div class="stat"><?= $stats['followups'] ?></div></div>
</div>

<div class="grid-2">
  <section class="card">
    <div class="card-head">
```

```
        <h3>Recent Patients</h3>
        <a class="btn btn-light" href="<?= base_url('patients.php') ?>">View all</a>
      </div>
      <div class="table-wrap">
        <table>
          <thead>

<tr><th>Name</th><th>Mobile</th><th>Hospital</th><th>Action</th></tr>
          </thead>
          <tbody>
          <?php foreach ($patients as $p): ?>
            <tr>
              <td><?= e($p['patient_name']) ?></td>
              <td><?= e($p['mobile_no']) ?></td>
              <td><?= e($p['hospital_name']) ?></td>
              <td><a  href="<?=  base_url('patient_view.php?id='  .  (int)$p['id'])
?>">Open</a></td>
            </tr>
          <?php endforeach; ?>
          <?php if (!$patients): ?>
            <tr><td colspan="4">No records found.</td></tr>
          <?php endif; ?>
          </tbody>
        </table>
      </div>
    </section>

    <section class="card">
      <h3>Workflow</h3>
      <div class="timeline">
```

```
        <div class="timeline-item"><strong>1.</strong> Upload discharge summary images</div>
        <div class="timeline-item"><strong>2.</strong> OpenAI extracts structured medical details</div>
        <div class="timeline-item"><strong>3.</strong> Save patient, doctor, hospital, medicine data</div>
        <div class="timeline-item"><strong>4.</strong> Generate patient, food, and activity advice</div>
        <div class="timeline-item"><strong>5.</strong> Create follow-up reminders and alerts</div>
      </div>
    </section>
</div>

<div class="cards" style="grid-template-columns:repeat(3,1fr);margin-top:10px">
    <a class="card" href="<?= base_url('upload.php') ?>"><h3>New Upload</h3><p class="subtle">Upload discharge summary images and extract details</p></a>
    <a class="card" href="<?= base_url('patient_edit.php') ?>"><h3>Edit Records</h3><p class="subtle">Correct or update extracted medical information</p></a>
    <a class="card" href="<?= base_url('reminders.php') ?>"><h3>Send Reminders</h3><p class="subtle">Trigger SMS and WhatsApp notifications</p></a>
</div>

<?php include BASE_PATH . '/app/views/layout/footer.php'; ?>
```

Parents

```
<?php
require_once __DIR__ . '/../app/bootstrap.php';
require_login();

$repo = new PatientRepository();
$search = trim($_GET['q'] ?? '');
$rows = $repo->allPatients($search);

$pageTitle = 'Patients';
$pageSubtitle = 'Search and manage patient records';

include BASE_PATH . '/app/views/layout/header.php';
include BASE_PATH . '/app/views/layout/sidebar.php';
include BASE_PATH . '/app/views/layout/topbar.php';
?>
<div class="card">
  <?php if ($msg = flash('success')): ?><div class="alert alert-success"><?= e($msg) ?></div><?php endif; ?>
  <form method="get" class="toolbar">
    <input type="text" name="q" value="<?= e($search) ?>" placeholder="Search by patient name, mobile, ID">
    <button class="btn btn-primary" type="submit">Search</button>
    <a class="btn btn-light" href="<?= base_url('upload.php') ?>">New Upload</a>
  </form>
  <div class="table-wrap">
    <table>
      <thead>
      <tr>
```

```
<th>ID</th><th>Patient</th><th>Age/Gender</th><th>Mobile</th><th>Hospital</th><th>Action</th>
        </tr>
        </thead>
        <tbody>
        <?php foreach ($rows as $row): ?>
        <tr>
           <td><?= e($row['patient_identifier']) ?></td>
           <td><?= e($row['patient_name']) ?></td>
           <td><?= e((string)$row['age']) ?> / <?= e($row['gender']) ?></td>
           <td><?= e($row['mobile_no']) ?></td>
           <td><?= e($row['hospital_name']) ?></td>
           <td><a href="<?= base_url('patient_view.php?id=' . (int)$row['id']) ?>">View</a></td>
        </tr>
        <?php endforeach; ?>
        <?php if (!$rows): ?>
        <tr><td colspan="6">No patients found.</td></tr>
        <?php endif; ?>
        </tbody>
      </table>
   </div>
</div>
<?php include BASE_PATH . '/app/views/layout/footer.php'; ?>
```

Doctors

```
<?php
declare(strict_types=1);

require_once __DIR__ . '/../app/bootstrap.php';
require_login();

$pdo = Database::connection();

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $stmt = $pdo->prepare("INSERT INTO doctors (doctor_name, specialization) VALUES (?, ?)");
    $stmt->execute([
        $_POST['name'] ?? '',
        $_POST['spec'] ?? ''
    ]);
}

$rows = $pdo->query("SELECT * FROM doctors")->fetchAll();

include BASE_PATH . '/app/views/layout/header.php';
include BASE_PATH . '/app/views/layout/sidebar.php';
include BASE_PATH . '/app/views/layout/topbar.php';
?>

<div class="card">
    <h3>Add Doctor</h3>

    <form method="post">
        <input name="name" placeholder="Doctor Name">
        <input name="spec" placeholder="Specialization">
```

```
        <button class="btn btn-primary">Save</button>
    </form>

    <table>
        <tr>
            <th>Name</th>
            <th>Specialization</th>
        </tr>
        <?php foreach ($rows as $r): ?>
            <tr>
                <td><?= htmlspecialchars($r['doctor_name']) ?></td>
                <td><?= htmlspecialchars($r['specialization']) ?></td>
            </tr>
        <?php endforeach; ?>
    </table>
</div>

<?php include BASE_PATH . '/app/views/layout/footer.php'; ?>
```

Hospitels

```
<?php
require_once __DIR__ . '/../app/bootstrap.php';
require_login();

$pdo = Database::connection();

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $stmt = $pdo->prepare("INSERT INTO hospitals (name, address) VALUES (?, ?)");
    $stmt->execute([
        $_POST['name'] ?? '',
        $_POST['address'] ?? ''
    ]);
}

$rows = $pdo->query("SELECT * FROM hospitals")->fetchAll();

include BASE_PATH . '/app/views/layout/header.php';
include BASE_PATH . '/app/views/layout/sidebar.php';
include BASE_PATH . '/app/views/layout/topbar.php';
?>

<div class="card">
    <h3>Add Hospital</h3>

    <form method="post">
        <input name="name" placeholder="Hospital Name">
        <input name="address" placeholder="Address">
        <button class="btn btn-primary">Save</button>
```

```
    </form>

    <table>
        <tr>
            <th>Name</th>
            <th>Address</th>
        </tr>
        <?php foreach ($rows as $r): ?>
            <tr>
                <td><?= htmlspecialchars($r['name']) ?></td>
                <td><?= htmlspecialchars($r['address']) ?></td>
            </tr>
        <?php endforeach; ?>
    </table>
</div>

<?php include BASE_PATH . '/app/views/layout/footer.php'; ?>
```

CSS

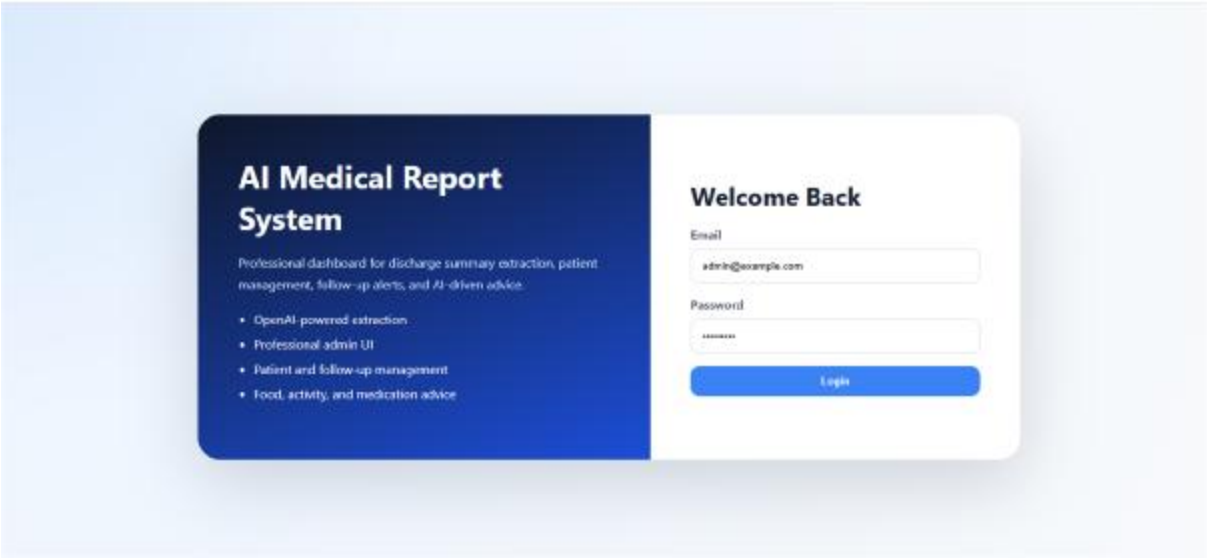
```
:root{
  --bg:#f4f7fb;
  --card:#ffffff;
  --text:#172033;
  --muted:#708099;
  --line:#e5ebf3;
  --primary:#3b82f6;
  --dark:#111827;
  --success:#14b86a;
  --danger:#dc3545;
  --shadow:0 16px 40px rgba(16,24,40,.08);
  --radius:18px;
}
*{box-sizing:border-box}
body{margin:0;font-family:Inter,Segoe UI,Arial,sans-serif;background:var(--bg);color:var(--text)}
a{text-decoration:none;color:var(--primary)}
.app-shell{display:flex;min-height:100vh}
.sidebar{width:270px;background:linear-gradient(180deg,#0f172a,#111827);color:#fff;padding:24px;position:sticky;top:0;height:100vh}
.brand{display:flex;gap:14px;align-items:center;margin-bottom:26px}
.brand-logo{width:50px;height:50px;border-radius:14px;background:linear-gradient(135deg,#60a5fa,#2563eb);display:grid;place-items:center;font-weight:700}
.brand h2{margin:0;font-size:22px}
.brand p{margin:4px 0 0;color:#a7b3c7;font-size:13px}
.menu{display:flex;flex-direction:column;gap:8px}
.menu a{padding:12px 14px;border-radius:12px;color:#d9e2f1;font-weight:500}
.menu a:hover{background:rgba(255,255,255,.08);color:#fff}
.main-content{flex:1;padding:28px}
```

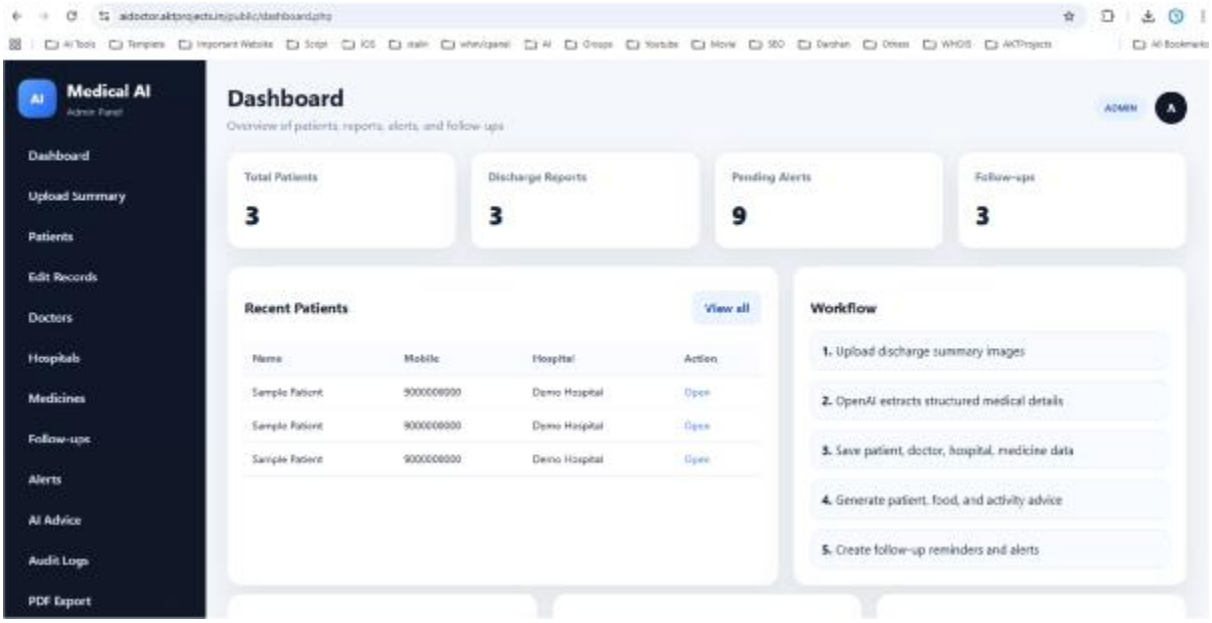
```
.topbar{display:flex;justify-content:space-between;align-items:center;margin-bottom:24px}
.topbar h1{margin:0;font-size:30px}
.subtle{color:var(--muted);margin:6px 0 0}
.topbar-user{display:flex;align-items:center;gap:12px}
.badge{background:#e8f1ff;color:#2456b8;padding:8px 12px;border-radius:999px;font-size:12px;font-weight:700;text-transform:uppercase}
.avatar{width:42px;height:42px;border-radius:50%;display:grid;place-items:center;background:var(--dark);color:#fff;font-weight:700}
.cards{display:grid;grid-template-columns:repeat(4,1fr);gap:18px;margin-bottom:22px}
.stack-cards{grid-template-columns:repeat(2,1fr)}
.card{background:var(--card);padding:22px;border-radius:var(--radius);box-shadow:var(--shadow);border:1px solid var(--line)}
.stat-card h3{margin:0 0 16px;font-size:15px;color:var(--muted)}
.stat{font-size:34px;font-weight:800}
.grid-2{display:grid;grid-template-columns:1.4fr 1fr;gap:18px}
.card-head{display:flex;justify-content:space-between;align-items:center;margin-bottom:14px}
.table-wrap{overflow:auto}
table{width:100%;border-collapse:collapse}
th,td{padding:12px 10px;border-bottom:1px solid var(--line);text-align:left;font-size:14px}
th{color:#5c6b83;font-weight:700;background:#fafcff}
.form-grid{display:grid;gap:16px}
label{display:grid;gap:8px;font-weight:600;color:#314159}
input[type="text"],input[type="email"],input[type="password"],input[type="file"]{
  width:100%;padding:14px 14px;border-radius:12px;border:1px solid #d6e0ee;background:#fff;outline:none
}
.toolbar{display:flex;gap:12px;align-items:center;margin-bottom:16px}
```

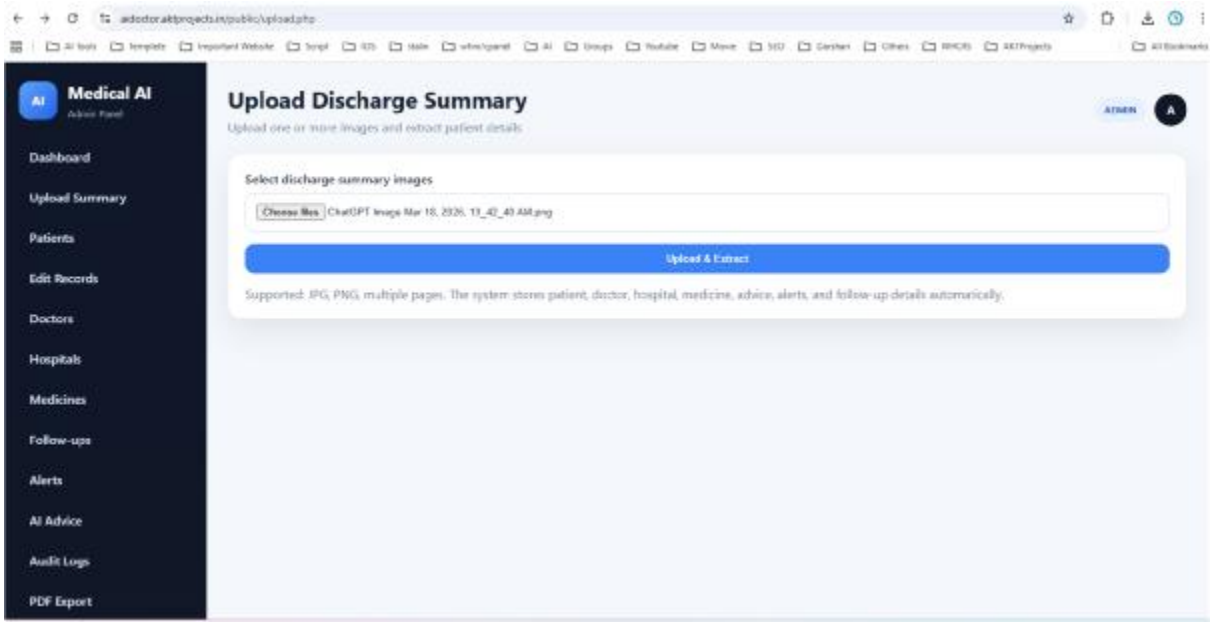
```
.toolbar input{max-width:340px}
.btn{display:inline-flex;align-items:center;justify-content:center;padding:12px
16px;border-radius:12px;border:none;cursor:pointer;font-weight:700}
.btn-primary{background:var(--primary);color:#fff}
.btn-light{background:#eef4ff;color:#2456b8}
.alert{padding:14px 16px;border-radius:12px;margin-bottom:16px;font-weight:600}
.alert-danger{background:#fff1f2;color:#a61b29;border:1px solid #ffd5dc}
.alert-success{background:#edfdf4;color:#067647;border:1px solid #c7f0d6}
.timeline{display:grid;gap:14px}
.timeline-item{padding:14px;border-radius:14px;background:#f8fbff;border:1px  solid
#e4eefb}
.list-item{padding:10px 0;border-bottom:1px dashed var(--line)}
.mt-16{margin-top:16px}
.settings-grid{display:grid;grid-template-columns:repeat(2,1fr);gap:18px}

.login-body{
  min-height:100vh;display:grid;place-items:center;
  background:radial-gradient(circle at top left,#dbeafe 0,#eff6ff 30%,#f8fafc 100%);
}
.login-card{
  width:min(1060px,92vw);background:#fff;border-
radius:28px;overflow:hidden;display:grid;
  grid-template-columns:1.1fr .9fr;box-shadow:0 30px 90px rgba(15,23,42,.16)
}
.login-left{
  padding:52px;background:linear-gradient(160deg,#0f172a,#1d4ed8);color:#fff
}
.login-left h1{font-size:40px;margin:0 0 18px}
.login-left p{font-size:16px;line-height:1.7;color:#dbe6ff}
.feature-list{padding-left:20px;line-height:2}
.login-right{padding:52px;display:flex;flex-direction:column;justify-content:center}
```

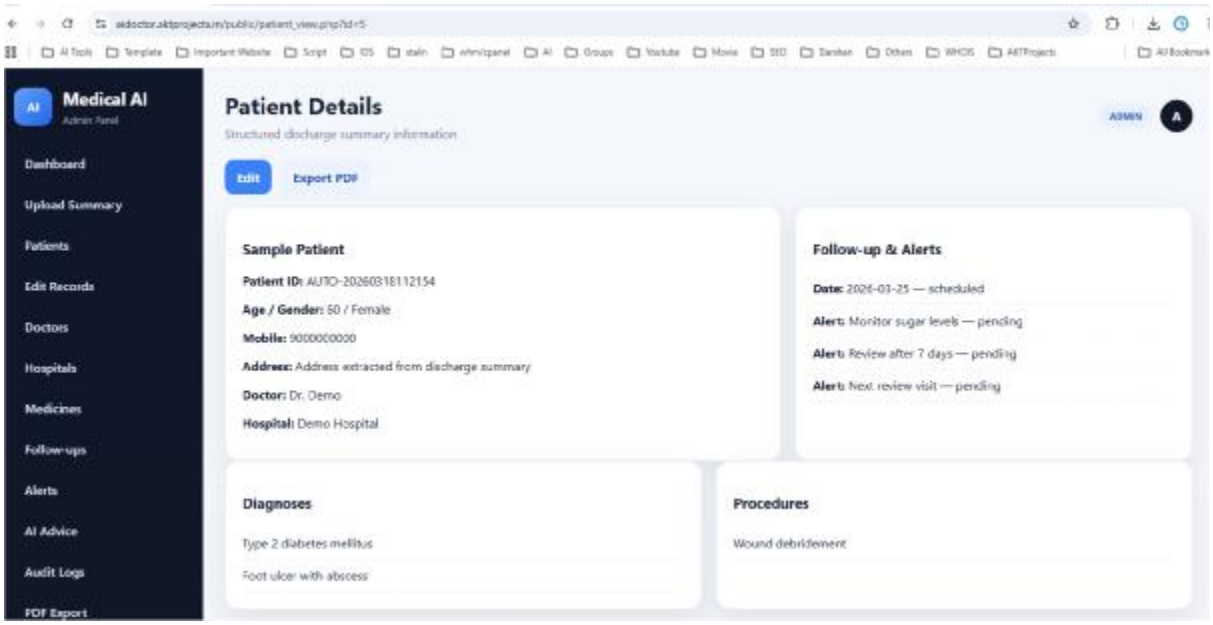
```
.login-right h2{margin:0 0 18px;font-size:32px}

@media (max-width: 1100px){
  .cards{grid-template-columns:repeat(2,1fr)}
  .grid-2,.stack-cards{grid-template-columns:1fr}
}
@media (max-width: 860px){
  .app-shell{flex-direction:column}
  .sidebar{width:100%;height:auto;position:relative}
  .toolbar{flex-wrap:wrap}
  .cards{grid-template-columns:1fr}
  .login-card{grid-template-columns:1fr}
}
```

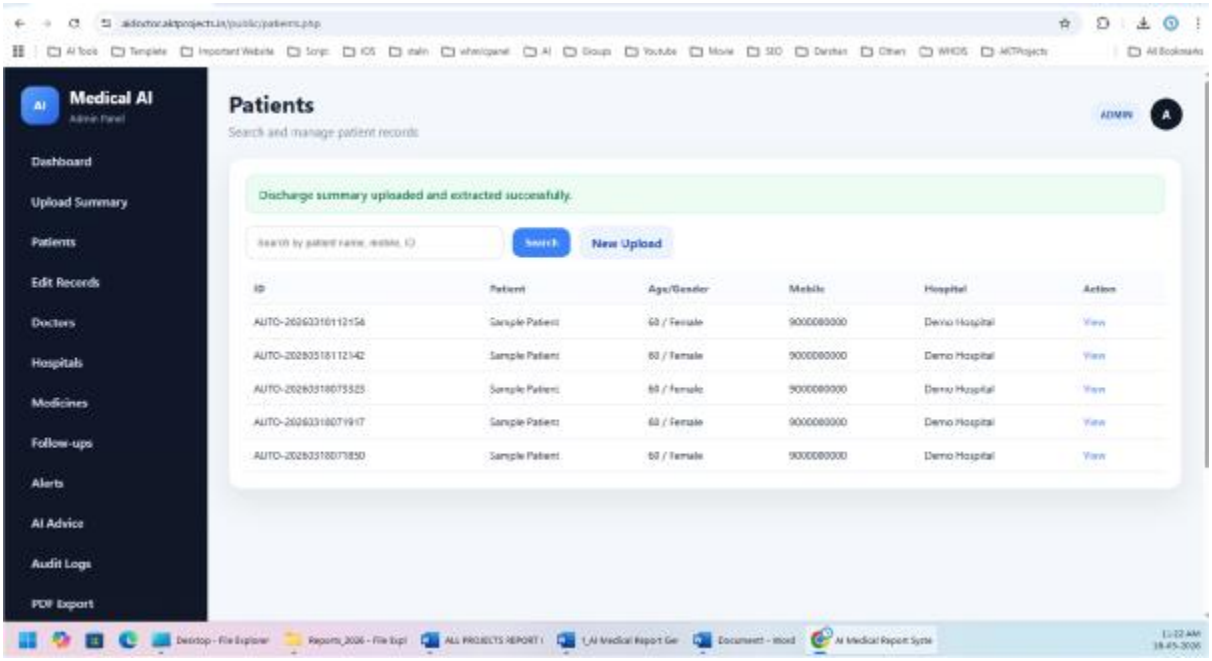
4.4 SCREENSHOT

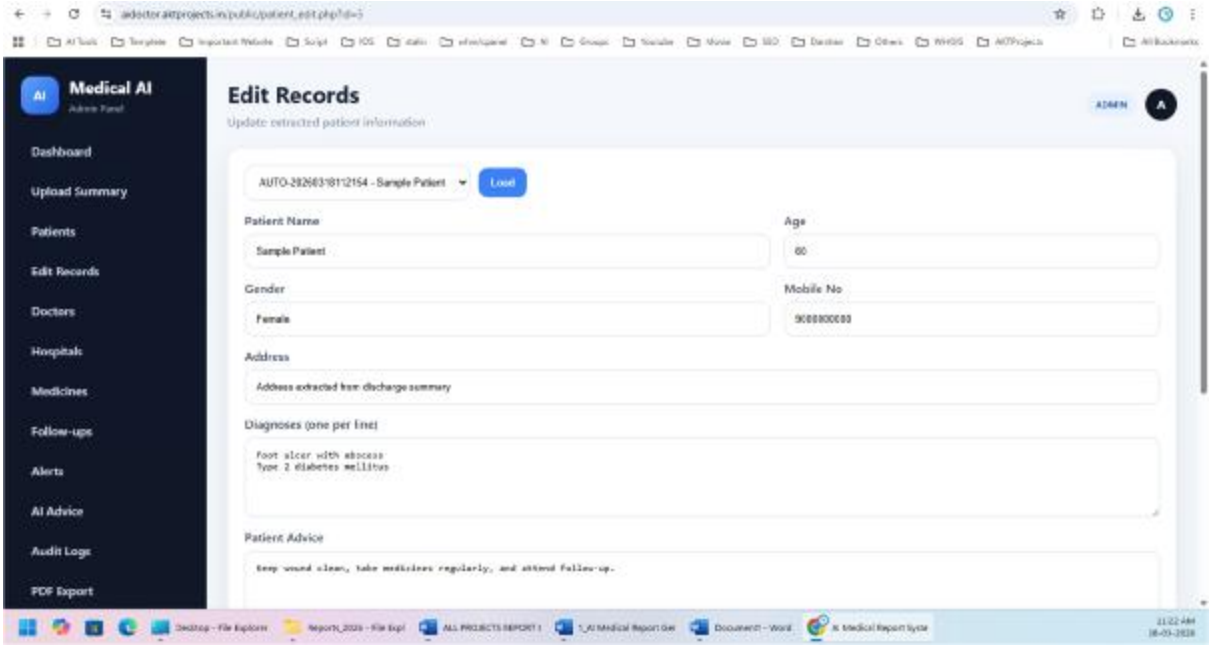


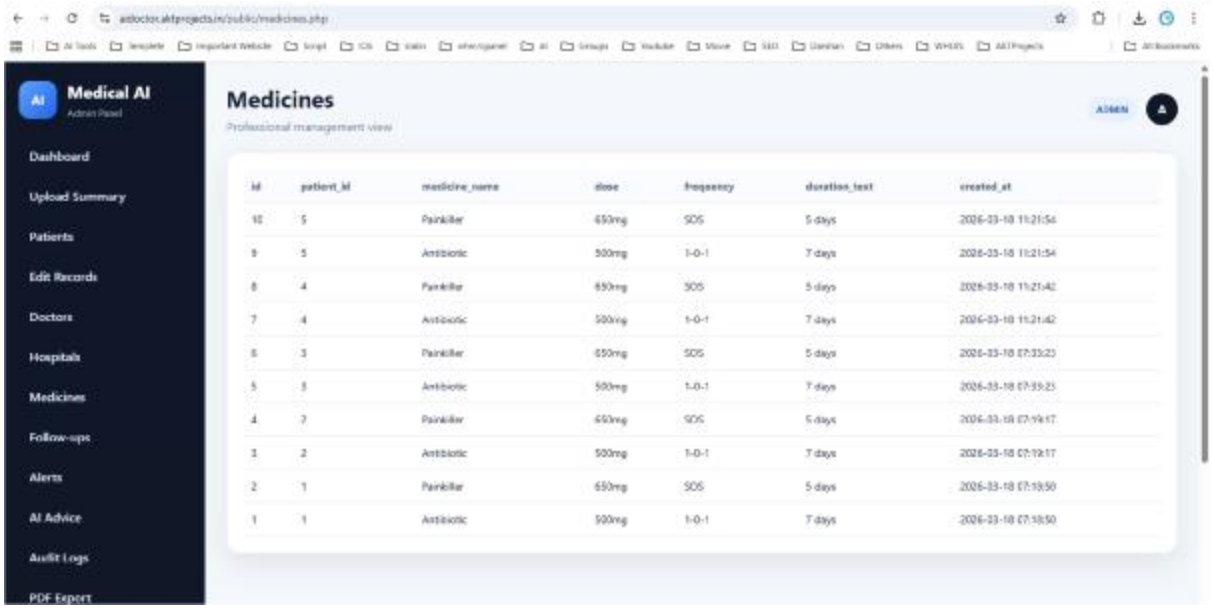


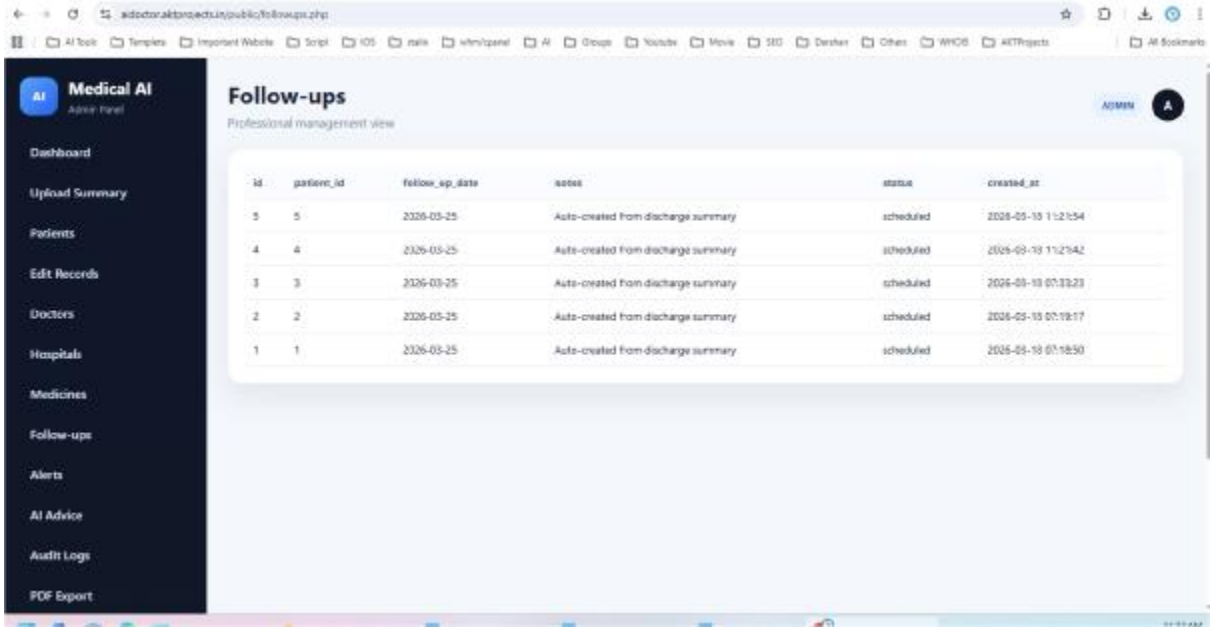

Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Upload Discharge Summary
Upload one or more images and extract patient details
ADMIN
A
Select discharge summary images
Choose files
ChatGPT Image Mar 18, 2026, 11_42_43 AM.png
Upload & Extract
Supported: JPG, PNG, multiple pages. The system stores patient, doctor, hospital, medicine, advice, alerts, and follow-up details automatically.

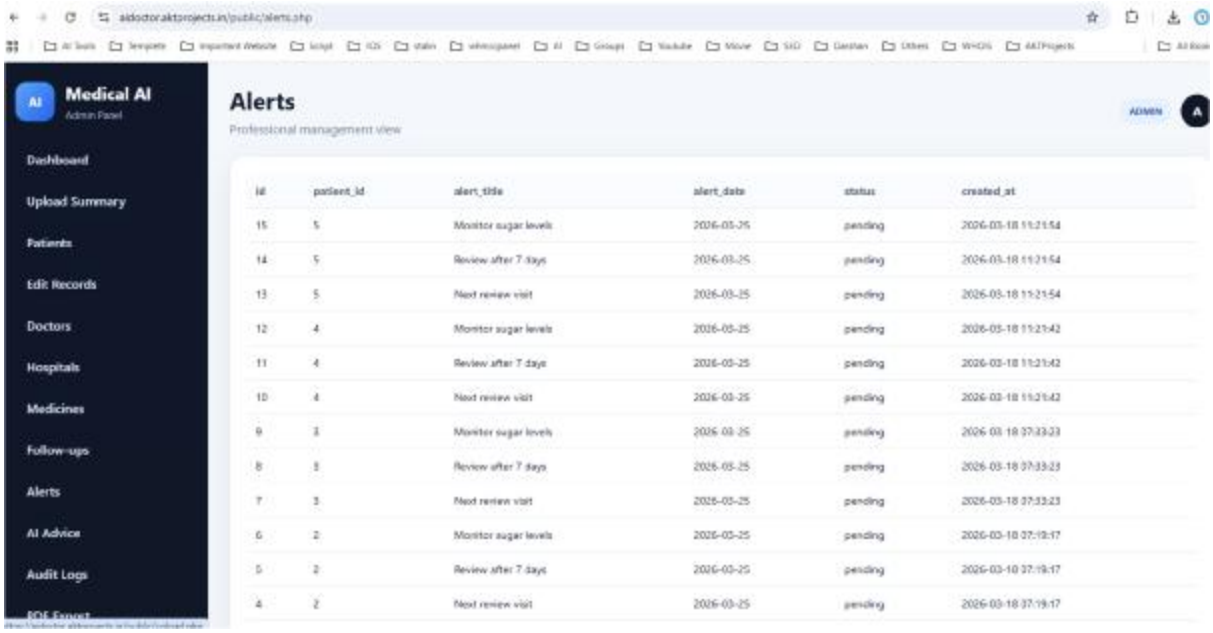

Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Patient Details
Structured discharge summary information
ADMIN
A
Edit
Export PDF
Sample Patient
Patient ID: AUTO-20260318112154
Age / Gender: 60 / Female
Mobile: 9000000000
Address: Address extracted from discharge summary
Doctor: Dr. Demo
Hospital: Demo Hospital
Follow-up & Alerts
Date: 2026-03-25 — scheduled
Alert: Monitor sugar levels — pending
Alert: Review after 7 days — pending
Alert: Next review visit — pending
Diagnoses
Type 2 diabetes mellitus
Foot ulcer with abscess
Procedures
Wound debridement

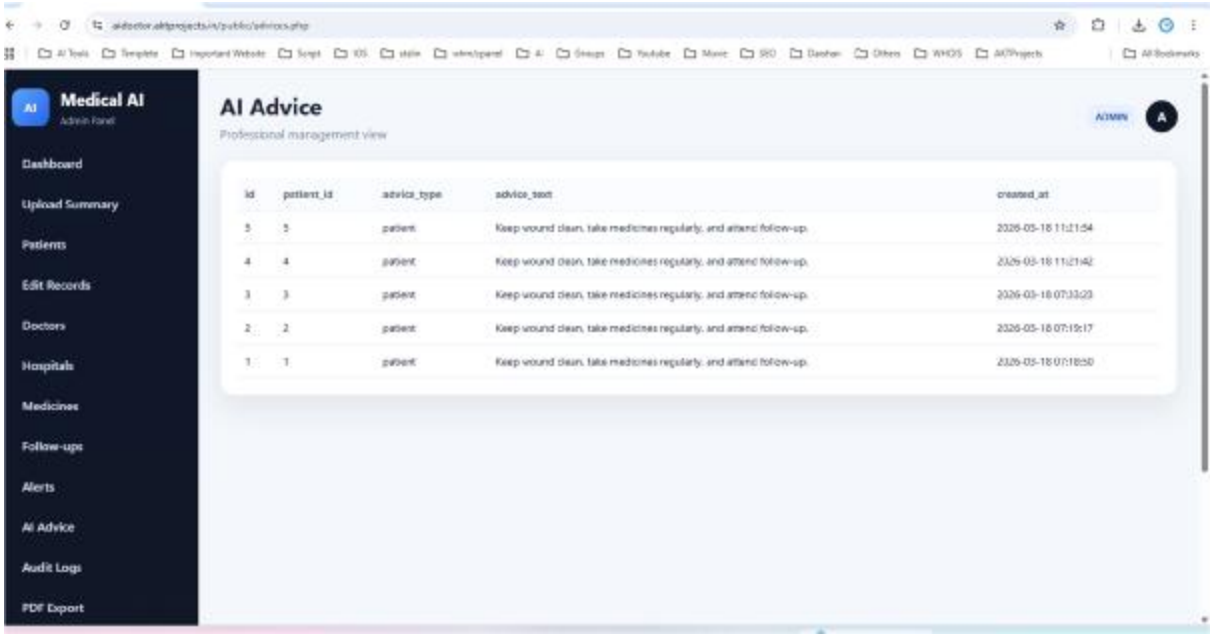

Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Patients
Search and manage patient records
ADMIN
Discharge summary uploaded and extracted successfully.
Search
New Upload
ID
Patient
Age/Gender
Mobile
Hospital
Action
AUTO-20260318112156
Sample Patient
60 / Female
9000000000
Demo Hospital
View
AUTO-20260318112142
Sample Patient
60 / Female
9000000000
Demo Hospital
View
AUTO-20260318073525
Sample Patient
60 / Female
9000000000
Demo Hospital
View
AUTO-20260318071917
Sample Patient
60 / Female
9000000000
Demo Hospital
View
AUTO-20260318071850
Sample Patient
60 / Female
9000000000
Demo Hospital
View

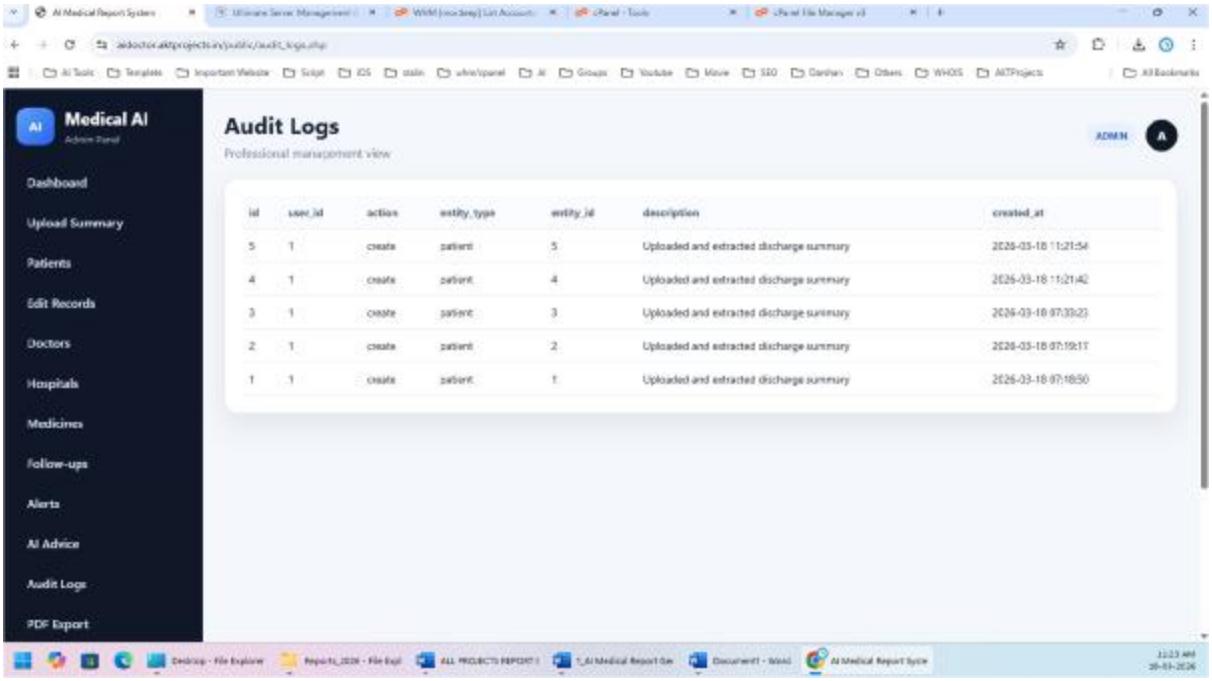

Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Edit Records
Update extracted patient information
ADMIN
AUTO-20260318112154 - Sample Patient
Load
Patient Name
Sample Patient
Age
60
Gender
Female
Mobile No
9000000000
Address
Address extracted from discharge summary
Diagnoses (one per line)
Foot ulcer with abscess
Type 2 diabetes mellitus
Patient Advice
Keep wound clean, take medicines regularly, and attend follow-up.

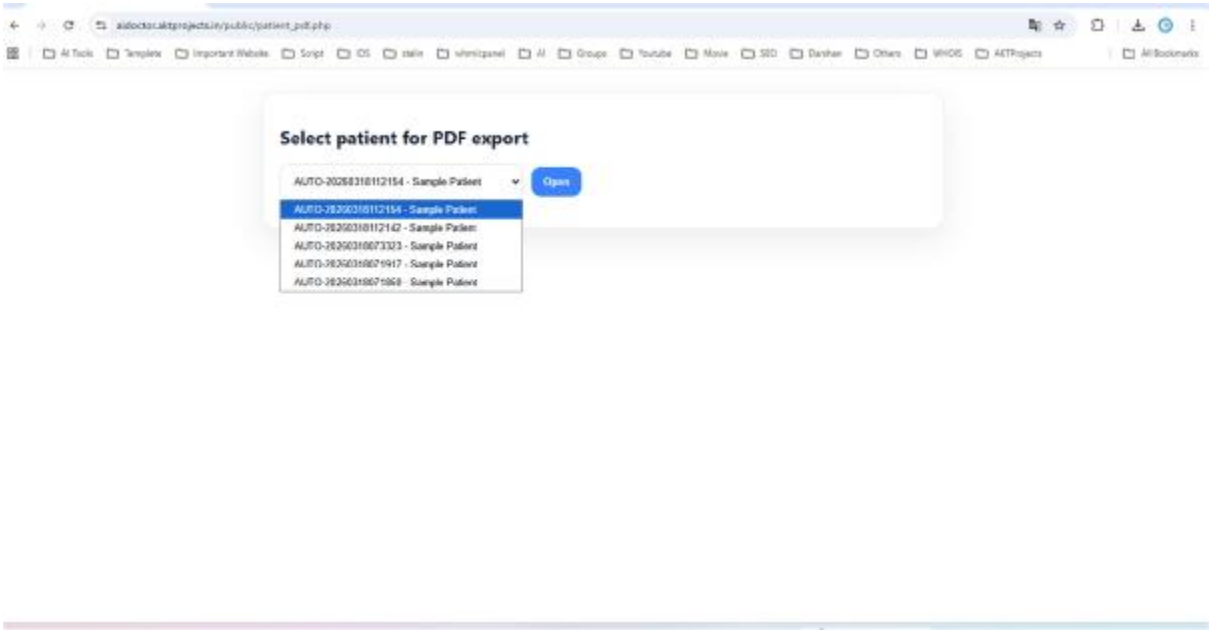

Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Medicines
Professional management view
ADMIN
id
patient_id
medicine_name
dose
frequency
duration_text
created_at
10 5 Painkiller 650mg SOS 5 days 2026-03-18 11:21:56
9 5 Antibiotic 500mg 1-0-1 7 days 2026-03-18 11:21:54
8 4 Painkiller 650mg SOS 5 days 2026-03-18 11:21:42
7 4 Antibiotic 500mg 1-0-1 7 days 2026-03-18 11:21:42
6 3 Painkiller 650mg SOS 5 days 2026-03-18 07:33:23
5 3 Antibiotic 500mg 1-0-1 7 days 2026-03-18 07:33:23
4 2 Painkiller 650mg SOS 5 days 2026-03-18 07:19:17
3 2 Antibiotic 500mg 1-0-1 7 days 2026-03-18 07:19:17
2 1 Painkiller 650mg SOS 5 days 2026-03-18 07:18:50
1 1 Antibiotic 500mg 1-0-1 7 days 2026-03-18 07:18:50

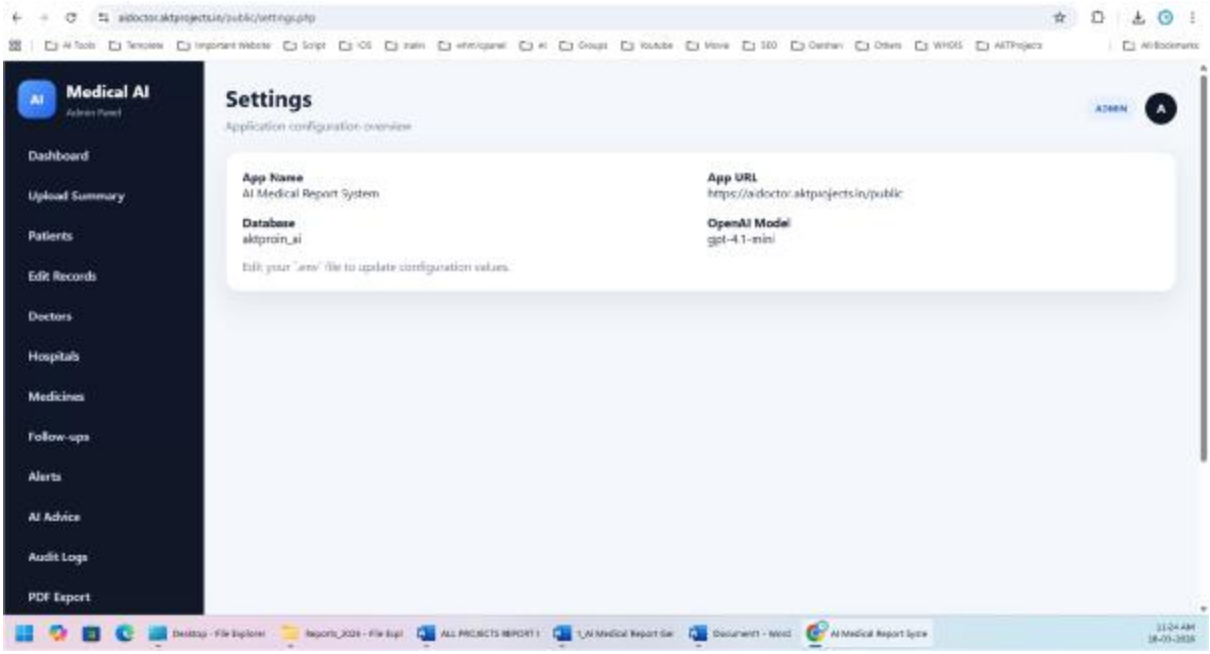

Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Follow-ups
Professional management view
ADMIN
id
patient_id
follow_up_date
notes
status
created_at
5 5 2026-03-25 Auto-created from discharge summary scheduled 2026-03-18 11:21:54
4 4 2026-03-25 Auto-created from discharge summary scheduled 2026-03-18 11:21:42
3 3 2026-03-25 Auto-created from discharge summary scheduled 2026-03-18 07:33:23
2 2 2026-03-25 Auto-created from discharge summary scheduled 2026-03-18 07:19:17
1 1 2026-03-25 Auto-created from discharge summary scheduled 2026-03-18 07:18:50


Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Alerts
Professional management view
ADMIN
id
patient_id
alert_title
alert_date
status
created_at
15 5 Monitor sugar levels 2026-03-25 pending 2026-03-18 11:21:54
14 5 Review after 7 days 2026-03-25 pending 2026-03-18 11:21:54
13 5 Next review visit 2026-03-25 pending 2026-03-18 11:21:54
12 4 Monitor sugar levels 2026-03-25 pending 2026-03-18 11:21:42
11 4 Review after 7 days 2026-03-25 pending 2026-03-18 11:21:42
10 4 Next review visit 2026-03-25 pending 2026-03-18 11:21:42
9 3 Monitor sugar levels 2026-03-25 pending 2026-03-18 07:33:23
8 3 Review after 7 days 2026-03-25 pending 2026-03-18 07:33:23
7 3 Next review visit 2026-03-25 pending 2026-03-18 07:33:23
6 2 Monitor sugar levels 2026-03-25 pending 2026-03-18 07:15:17
5 2 Review after 7 days 2026-03-25 pending 2026-03-18 07:15:17
4 2 Next review visit 2026-03-25 pending 2026-03-18 07:15:17


Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
AI Advice
Professional management view
ADMIN
id
patient_id
advice_type
advice_text
created_at
5 5 patient Keep wound clean, take medicines regularly, and attend follow-up. 2026-03-18 11:21:54
4 4 patient Keep wound clean, take medicines regularly, and attend follow-up. 2026-03-18 11:21:42
3 3 patient Keep wound clean, take medicines regularly, and attend follow-up. 2026-03-18 07:33:23
2 2 patient Keep wound clean, take medicines regularly, and attend follow-up. 2026-03-18 07:15:17
1 1 patient Keep wound clean, take medicines regularly, and attend follow-up. 2026-03-18 07:18:50


Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Audit Logs
Professional management view
ADMIN
A
id
user_id
action
entity_type
entity_id
description
created_at
5 1 create patient 5 Uploaded and extracted discharge summary 2026-03-18 11:21:54
4 1 create patient 4 Uploaded and extracted discharge summary 2026-03-18 11:21:42
3 1 create patient 3 Uploaded and extracted discharge summary 2026-03-18 07:33:23
2 1 create patient 2 Uploaded and extracted discharge summary 2026-03-18 07:19:17
1 1 create patient 1 Uploaded and extracted discharge summary 2026-03-18 07:18:50


Select patient for PDF export
AUTO-20260318112154 - Sample Patient
Open
AUTO-20260318112154 - Sample Patient
AUTO-20260318112142 - Sample Patient
AUTO-20260318073323 - Sample Patient
AUTO-20260318071917 - Sample Patient
AUTO-20260318071858 - Sample Patient


Medical AI
Admin Panel
Dashboard
Upload Summary
Patients
Edit Records
Doctors
Hospitals
Medicines
Follow-ups
Alerts
AI Advice
Audit Logs
PDF Export
Settings
Application configuration overview
ADMIN
A
App Name
AI Medical Report System
App URL
https://aidoctor.aktprojects.in/public
Database
aktproin_ai
OpenAI Model
gpt-4.1-mini
Edit your `.env` file to update configuration values.

5. CONCLUSION

The AI Medical Report System successfully addresses the challenges associated with manual handling of discharge summaries and patient records by introducing an intelligent, automated, and scalable solution. By integrating OpenAI-powered extraction, the system is capable of converting unstructured medical documents into structured and meaningful data with high accuracy, significantly reducing manual effort and human errors.

The system provides a professional dashboard that enables efficient management of patients, reports, and follow-up schedules. Features such as automated follow-up alerts, centralized data storage, and real-time access improve operational efficiency in hospitals and clinics. Additionally, the inclusion of AI-driven modules for food advice, activity guidance, and medication recommendations enhances post-discharge patient care and ensures better adherence to medical instructions.

Deployment on a Cloud VPS with WHM/cPanel ensures reliability, scalability, and secure access, while integration with the WhatsApp Official API enables effective communication through real-time notifications and reminders. The use of modern technologies such as PHP, MySQL, and Bootstrap ensures that the system is user-friendly, maintainable, and suitable for real-world healthcare environments.

In conclusion, the AI Medical Report System not only digitizes medical records but also adds intelligence and automation to healthcare management. It improves accuracy, saves time, enhances patient engagement, and supports better clinical outcomes, making it a valuable solution for modern healthcare institutions.

Future Enhancements

- Integration with Electronic Health Record (EHR) systems for seamless data sharing
- Multi-language support for better accessibility across regions
- Voice input and speech-to-text for faster report entry
- Mobile application (Android/iOS) for doctors and patients
- Advanced AI diagnosis support and risk prediction
- SMS and Email notifications along with WhatsApp alerts
- Cloud storage integration for scalable and secure data handling
- Role-based analytics dashboard with reports and insights
- Barcode/QR code integration for quick patient identification
- Enhanced data security with encryption and audit logs

REFERENCES

1. OpenAI, OpenAI API Documentation, https://platform.openai.com/docs
2. World Health Organization (WHO), Digital Health and Patient Records Guidelines, https://www.who.int
3. Health Level Seven International (HL7), Electronic Health Records Standards, https://www.hl7.org
4. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016
5. Daniel Jurafsky, James H. Martin, Speech and Language Processing, Pearson, 3rd Edition
6. Raghupathi W., Raghupathi V., Big Data Analytics in Healthcare: Promise and Potential, Health Information Science
7. PHP Official Documentation, https://www.php.net/docs.php
8. MySQL Documentation, MySQL Reference Manual, https://dev.mysql.com/doc
9. Bootstrap Documentation, https://getbootstrap.com/docs
10. Apache HTTP Server Documentation, https://httpd.apache.org/docs
11. Meta (Facebook), WhatsApp Business API Documentation, https://developers.facebook.com/docs/whatsapp
12. Amazon Web Services (AWS), Cloud VPS and Hosting Services, https://aws.amazon.com
13. DigitalOcean, Virtual Private Server (VPS) Documentation, https://www.digitalocean.com/docs
14. Pressman R.S., Software Engineering: A Practitioner's Approach, McGraw-Hill
15. Sommerville I., Software Engineering, Pearson Education