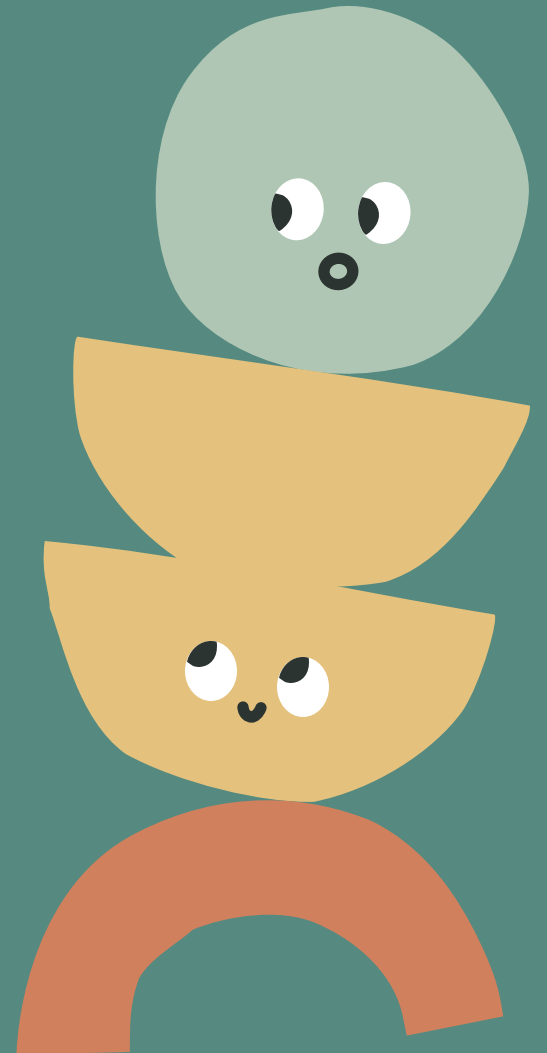


# Healthcare Information System for Small Clinics

A digital system that helps small clinics manage patients, appointments, and billing more easily.





# **Introduction and Vision**

# Project Overview and Vision Statement

## Centralized Healthcare Platform

A secure, web-based platform centralizes patient records, appointments, and billing for small clinics.

## Empowering Small Clinics

The system offers an affordable, user-friendly, HIPAA-compliant solution enhancing patient care and reducing errors.

## Streamlined Workflow and Security

This system improves workflow efficiency, data accuracy, and ensures secure handling of sensitive patient information.

## Scalable and Compliant Vision

The system supports scalable growth, aligns with healthcare regulations, and integrates with future external services.





# **Project Goals and Requirements**

# Goals and Functional Requirements



## Centralized System Goals

Create one main database and website where all clinic information is stored safely.

Users will have different access levels for security.

## Automation and Documentation

Let the system handle simple tasks automatically and provide clear instructions so it's easy to set up and maintain.

## Patient and Appointment Management

Store patient information and schedule appointments without double-booking, helping the clinic stay organized.

## Billing and Reporting Features

Create bills, track payments, and generate reports about visits and clinic revenue.



# Nonfunctional Requirements and Usability

## Usability and Performance

The system should be easy to use, not require much training, handle about 20 users at the same time, and load quickly (under 2 seconds).

## Reliability and Security

The system should work almost all the time (99% uptime), back up data every day, keep data encrypted, and follow HIPAA rules for safety.

## Portability and Maintainability

The system should work on all major web browsers, be built in a way that is easy to update, include clear documentation, and have automated testing to help keep it running smoothly.



# **System Design and Architecture**

# Architecture and Technology Stack

**Three-Tier Architecture:** The system is designed with three main parts:

- **Presentation Layer** – what the user sees on the screen
- **Application Layer** – the system logic that processes information
- **Data Layer** – the database that stores patient, appointment, and billing data

## Presentation Layer Technologies

The user interface that staff would use.

It could be built with **HTML, CSS, JavaScript, and Bootstrap** to create simple and responsive screens.

## Application Layer Frameworks

This part controls how the system works behind the scenes.

It would handle tasks like:

- saving and updating patient records
- scheduling appointments
- processing billing information

## Data Layer and Supporting Tools

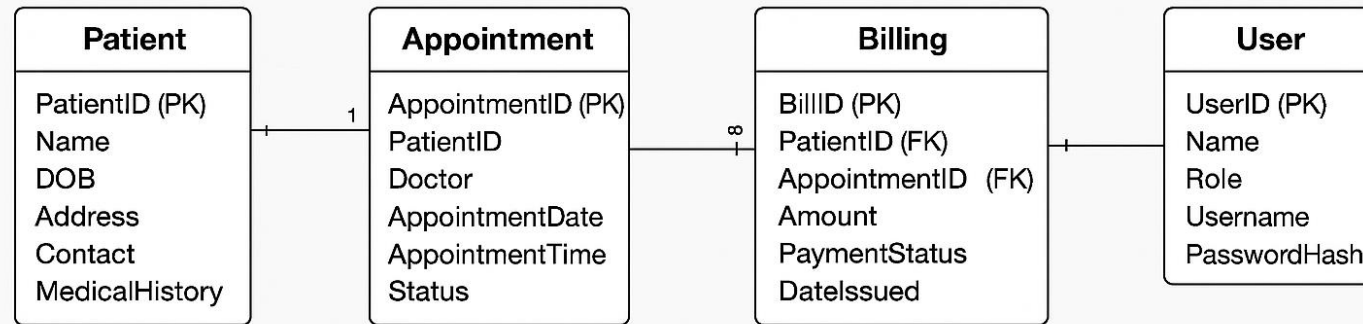
This is the part fully built in the prototype.

The system uses **MySQL** to store all information, including:

- Patient data
- Appointment schedules
- User accounts
- Billing records







# ERD Design



# **Security, Risk Management, and Implementation**

# Security Controls and Risk Mitigation

RISK	LIKELIHOOD	IMPACT	MITIGATION
Data breach	Medium	High	Encryption, RBAC, MFA, audits
Ransomware	Low	High	Offline backups, antimalware
Server downtime	Low	Medium	Redundant hosting, monitoring
Human error	Medium	Medium	Validation, training, recovery
HIPAA noncompliance	Low	High	Policy reviews, audits



# Patient & Billing Prototype Table

[illegible]

# Appointment & User Prototype Table

Local Instance 3306 - Warning - not supported

AdministrationSchemasSQL File 6\*BillingPatientAppointmentUserContext HelpSnippets

SCHEMAS

Filter objects

clinic\_his

TablesViewsStored ProceduresFunctionsInventoryDB

Object InfoSession

No object selected

1INSERT INTO Appointment (PatientID, Doctor, AppointmentDate, AppointmentTime, Status)

2VALUES (1, 'Dr. Smith', '2025-02-01', '11:00:00', 'Scheduled');

3SELECT \* FROM Appointment;

4SELECT \* FROM Appointment

5WHERE AppointmentID = 1;

6-- Test Case 6: Update appointment status to Completed

7UPDATE Appointment

8SET Status = 'Completed'

9WHERE AppointmentID = 1;

10

11-- Test Case 6: Update appointment status to Completed

100%1/4

Result Grid

Filter Rows:Search

Edit:Export/Import:

Appoin...	PatientID	Doctor	AppointmentD...	AppointmentTI...	Status
1	1	Dr. Smith	2025-01-15	10:30:00	Completed
2	1	Dr. Smith	2025-01-15	10:30:00	Scheduled
3	1	Dr. Smith	2025-12-05	09:00:00	Scheduled
4	1	Dr. Smith	2025-12-05	09:00:00	Scheduled
5	1	Dr. Smith	2025-12-05	09:00:00	Scheduled
6	1	Dr. Smith	2025-02-01	11:00:00	Scheduled
7	1	Dr. Smith	2025-12-05	09:00:00	Scheduled
NULL	NULL	NULL	NULL	NULL	NULL

Form EditorField TypesQuery StatsExecution Plan

Appointment 2

ApplyRevert

Action Output

	Time	Action	Response	Duration / Fetch Time
8	18:27:46	SELECT * FROM Appointment LIMIT 0, 1000	7 row(s) returned	0.00053 sec / 0.0000...

Query Completed

Local Instance 3306 - Warning - not supported

AdministrationSchemasSQL File 6\*BillingPatientAppointmentUserContext HelpSnippets

SCHEMAS

Filter objects

clinic\_his

TablesViewsStored ProceduresFunctionsInventoryDB

Object InfoSession

No object selected

1SELECT \* FROM clinic\_his.User;

2-- TEST CASE 7: Show all UNPAID bills with patient name and appointment date

3SELECT

4b.BillID,

5p.Name AS PatientName,

6a.AppointmentDate,

7a.AppointmentTime,

8b.Amount,

9b.PaymentStatus

10FROM clinic\_his.Billing b

11JOIN clinic\_his.Patient p

100%1/2

Result Grid

Filter Rows:Search

Edit:Export/Import:

UserID	Name	Role	Username	PasswordHash
1	Admin User	Admin	admin	admin123
2	Dr Smith	Doctor	drsmith	pass123
4	Sarah Johnson	Receptionist	sarah	pass123
5	Bill Carter	Billing	bill	pay123
NULL	NULL	NULL	NULL	NULL

Form EditorField TypesQuery StatsExecution Plan

User 1

ApplyRevert

Action Output

	Time	Action	Response	Duration / Fetch Time
7	18:27:31	SELECT * FROM clinic_his.User LIMIT 0, 1000	4 row(s) returned	0.0021 sec / 0.00001...

Query Completed





# Implementation Schedule and Testing

## Five-Week Implementation Plan

The project is completed in five weeks, moving through requirements, design, backend work, the user interface, and billing features.

## Focused Development Phases

Each week has its own goals, such as creating the ERD, building the API, designing the UI, and finishing different parts of the system.

## Comprehensive Testing

We test everything, including patient records, appointment scheduling, and billing, to make sure the system works correctly.





# **Data Conversion and User Training**

# Data Migration and Verification



## Data Source Identification

First, find where the old information is stored, such as patient files, spreadsheets, or billing records.

## Data Mapping and Importing

Next, organize this information and safely move patient, appointment, and billing data into the new system.

## Verification and Validation

Run checks to make sure the data is correct, not duplicated, and properly linked.

## Operational Efficiency

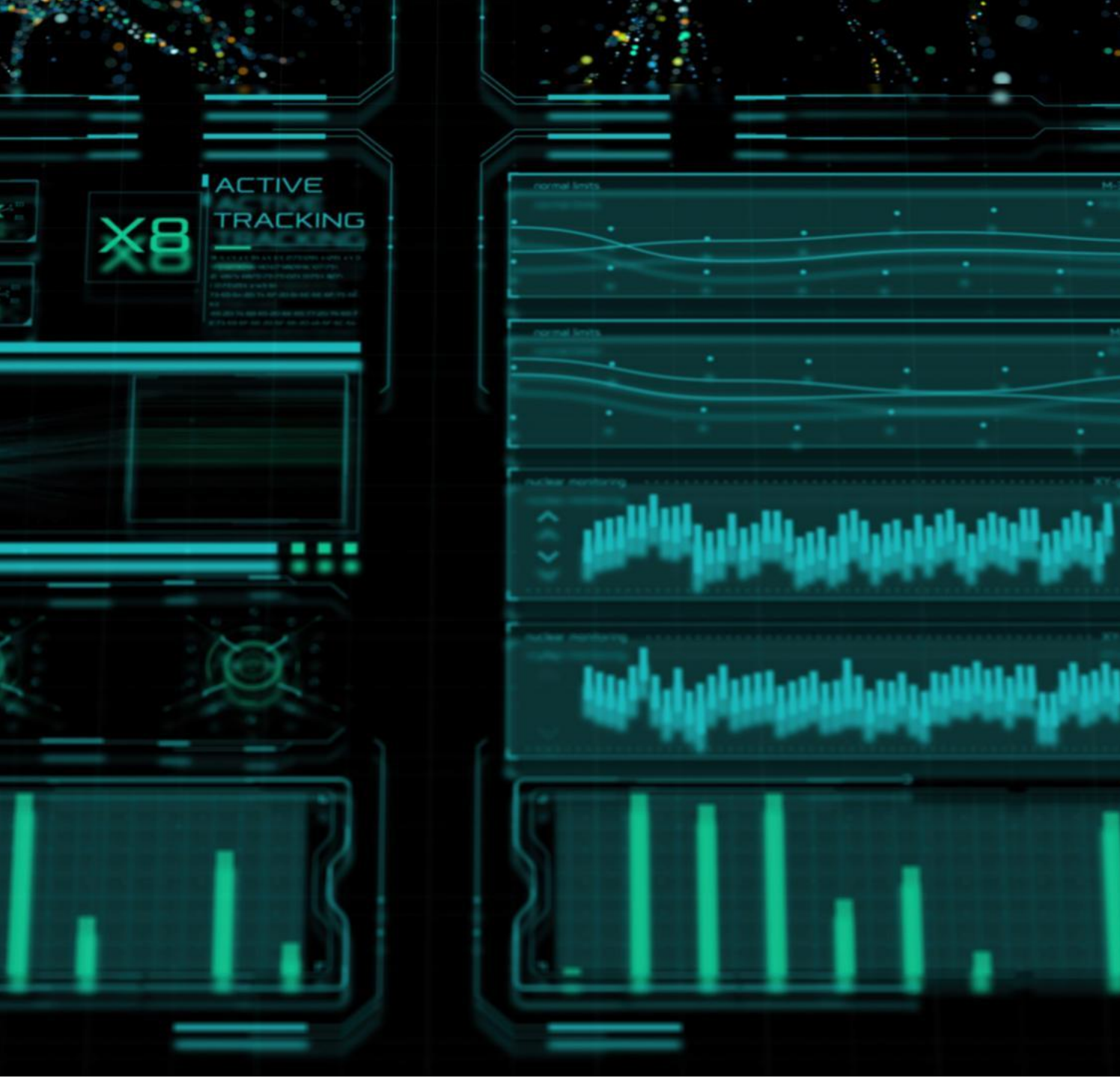
Good data migration helps reduce manual work, prevents mistakes, and makes the clinic run more smoothly.

# Training Objectives and Schedule

DAY	USER ROLE	TRAINING TOPICS
Day 1	Admin & Receptionists	Login, patient registration, appointment scheduling
Day 2	Billing Staff	Invoice creation, payments, financial reports
Day 3	Clinical Staff	Viewing charts, HIPAA guidelines
Ongoing	All Users	Refresher sessions, troubleshooting, Q&A



# **Conclusion and Next Steps**



# Summary and Future Enhancements

## Comprehensive Patient Management

The system makes it easier to manage patients, schedule appointments, and handle billing while keeping information safe and secure.

## Operational Efficiency

The strong system design and clear processes help the clinic work faster and provide better patient care.

## Future System Enhancements

In the future, the system could add new features like insurance integration, more detailed reports, and a mobile app.