

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 built in three parts:

- What the user sees (front end)
 - The logic behind the system (back end)
 - The database that stores information
- This design makes the system easy to grow and maintain.

Presentation Layer Technologies

The front end uses HTML, CSS, JavaScript, and Bootstrap to create pages that look good and work well on any device.

Application Layer Frameworks

The back end uses Python frameworks:

- **Flask** for simple prototypes
- **Django** for larger, more powerful systems

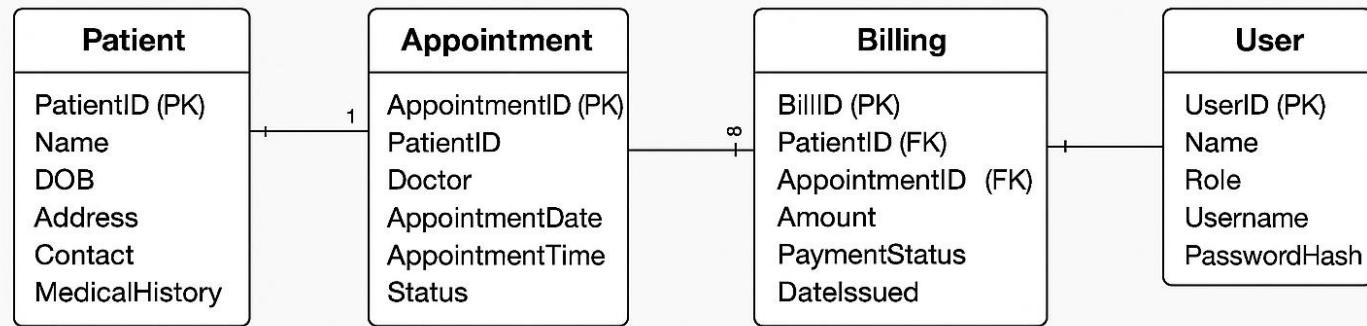
Data Layer and Supporting Tools

The database uses MySQL or PostgreSQL with SQLAlchemy to manage data.

Tools like Nginx, Gunicorn, and Certbot help keep the system fast, secure, and reliable.



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

Local Instance 3306 - Warning - not supported

Administration Schemas SQL File 6* Billing Patient Appointment User

Limit to 1000 rows

Context Help Snippets

Schemas

Filter objects

clinichis

Tables Views Stored Procedures Functions InventoryDB

Object Info Session

No object selected

```
1 • SELECT * FROM clinic_his.Billing;
2 • INSERT INTO Billing (PatientID, AppointmentID, Amount, PaymentStatus, DateIssued)
VALUES (1, 1, 150.00, 'Unpaid', '2025-12-01');
3
4
5 • SELECT * FROM Billing;
-- TEST CASE 5: Update Billing Payment Status to Paid
-- (Change the BillID number if yours is different)
6
7
8
9 • UPDATE Billing
10 SET PaymentStatus = 'Paid'
11 WHERE BillID = 1;
```

Result Grid Filter Rows: Search Export/Import:

BillID	PatientID	AppointmentID	Amount	PaymentStatus	DateIssued
1	1	1	150.00	Paid	2025-11-24
2	1	1	150.00	Unpaid	2025-12-01
3	1	1	150.00	Unpaid	2025-12-01
4	1	1	150.00	Unpaid	2025-12-01
5	1	1	150.00	Unpaid	2025-12-01
HULL	HULL	HULL	HULL	HULL	HULL

Billing 5

Result Grid Form Editor Field Types Query Stats Execution Plan

Action Output

Time	Action	Response	Duration / Fetch Time
16 18:28:10	INSERT INTO Billing (PatientID, AppointmentID, Amount, PaymentStatus, Date... 1 row(s) affected		0.00099 sec

Query Completed

Local instance 3306 - Warning - not supported

Administration Schemas SQL File 6* Billing Patient Appointment User

Limit to 1000 rows

Context Help Snippets

Schemas

Filter objects

clinichis

Tables Views Stored Procedures Functions InventoryDB

Object Info Session

No object selected

```
1 • SELECT * FROM clinic_his.Patient;
2
```

Result Grid Filter Rows: Search Export/Import:

PatientID	Name	DOB	Address	Contact	MedicalHisto...
1	John Doe	1990-05-10	123 Main St	555-1234	None
2	John Doe	1990-05-10	123 Main St	555-1234	None
HULL	HULL	HULL	HULL	HULL	HULL

Patient 1

Result Grid Form Editor Field Types Query Stats Execution Plan

Action Output

Time	Action	Response	Duration / Fetch Time
9 18:27:55	SELECT * FROM clinic_his.Patient LIMIT 0, 1000	2 row(s) returned	0.00087 sec / 0.0000...

Query Completed

Appointment & User Prototype Table

The screenshot shows the SSMS interface with the following details:

- Toolbar:** Standard SSMS toolbar with icons for file operations, schema browser, and search.
- Schemas:** Schemas pane showing the current schema is "Administration".
- Query Editor:** A query window containing the following T-SQL code:

```
1 • SELECT * FROM clinic_his.User;
2 | -- TEST CASE 7: Show all UNPAID bills with patient name and appointment date
3 • SELECT
4     b.BILLID,
5     p.Name AS PatientName,
6     a.AppointmentDate,
7     a.AppointmentTime,
8     b.Amount,
9     b.PaymentStatus
10    FROM clinic_his.Billing b
11   JOIN clinic_his.Patient p
```
- Result Grid:** The results of the query are displayed in a grid format. The columns are UserID, Name, Role, Username, and PasswordHash. The data is as follows:

	UserID	Name	Role	Username	PasswordHash
1	1	Admin User	Admin	admin	admin123
2	2	Dr Smith	Doctor	drsmith	pass123
3	3	Sarah Johnson	Receptionist	sarah	pass123
4	4	Bill Carter	Billing	bill	pay123
		(null)	(null)	(null)	(null)
- Context Help:** A floating panel on the right side of the screen providing help information about context help.
- Result Grid Panel:** A vertical panel on the right containing links to Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan.
- Action Output:** A table showing the last action taken, which was a query execution.



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.

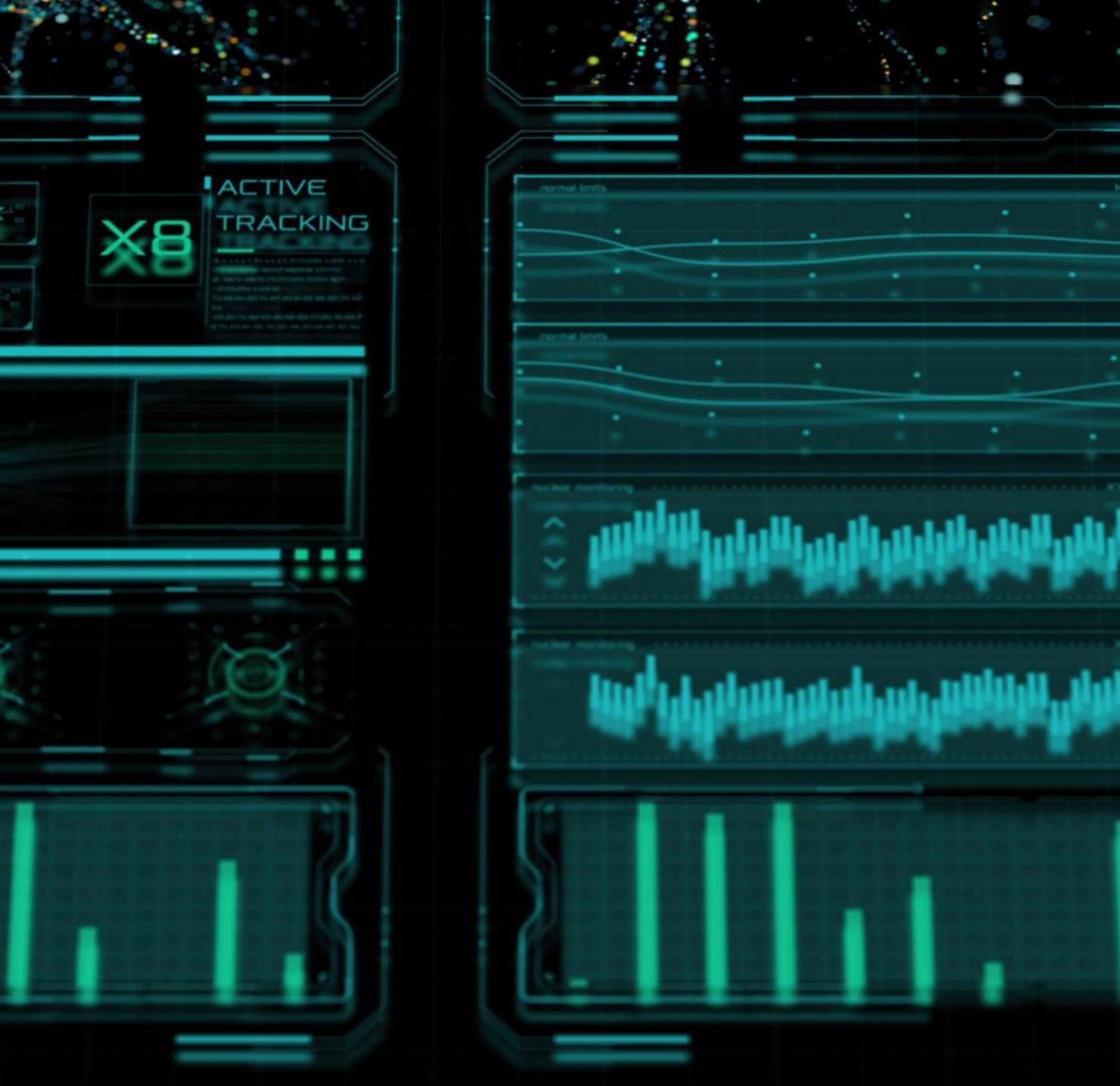
Operational Efficiency

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.