🧪 Capstone Project: Online Pathology Lab Booking System

(MERN Stack + PostgreSQL + Azure + Python Chatbot)

A patient-facing lab test booking platform with secure appointment handling, test catalogs, and real-time notifications. Built using MERN stack principles, modular microservices, PostgreSQL database, and a Python-based GenAI helpdesk.

🚩 Milestone 1: PostgreSQL Integration and Node for Backend

🎯 Goal:
Implement initial microservices (UserService, TestService), design PostgreSQL schemas, and build a responsive React UI for test selection and appointment booking.

🔧 Tasks & Subtasks

Task	Subtasks
PostgreSQL Schema	Design schemas for users, tests, appointments
Microservice Setup	Create basic UserService and TestService using Node.js
React UI	Pages for registration, login, dashboard, and test booking
Form Handling	Email, phone, password validation in React
Appointment UI	Display slots, test availability calendar
GitHub Versioning	Repo setup, commits, issue tracking, README

📦 Deliverables

- PostgreSQL schema + sample inserts
- Node.js microservices (basic structure for 2 services)
- React UI for booking, login/registration, test listings
- GitHub repo with structured commit history

🧾 Rubric (100 Marks)

Criteria	Marks
PostgreSQL Schema & Query Handling	20
Node Microservices Structure	20
React UI Quality	20
Validation & UX	10
Booking Flow Logic	15

Criteria	Marks
GitHub Versioning & Structure	10
Use Case Completeness	5

🚩 Milestone 2: Backend with Express and API Integration, React Frontend

🎯 Goal:

Build RESTful APIs using Express for all booking workflows, implement JWT-based authentication, and connect frontend forms to backend services.

🔧 Tasks & Subtasks

Task	Subtasks
Express API Development	CRUD endpoints for users, appointments, tests
JWT Authentication	Secure login/session handling
Middleware	Input sanitization, logging, error handling
React Integration	Use Axios to consume APIs for booking and test management
Postman Testing	API testing and request validation
Security Best Practices	Helmet, CORS, bcrypt, etc.

📦 Deliverables

- Express backend source code
- JWT-based auth and middleware
- Postman collection with tested routes
- Frontend integrated with backend
- GitHub repo with documentation

🧾 Rubric (100 Marks)

Criteria	Marks
API Implementation (CRUD)	20
JWT or Session Auth	15
Frontend Integration with API	20
Middleware, Validation & Logging	10
API Testing via Postman	10

Criteria	Marks
Navigation and UI-API Flow	15
Security Setup	10

🚩 Milestone 3: Azure Deployment & GitHub Workflow

🎯 Goal:
Refactor services (AuthService, TestService, BookingService) into containers, deploy using Azure App Services, and manage the DevOps workflow with GitHub Actions and Jira/Trello.

🔧 Tasks & Subtasks

Task	Subtasks
Microservice Refactor	Separate microservices for Auth, Booking, and Tests
Dockerization	Dockerfiles and docker-compose.yml setup
Azure Deployment	Deploy at least one service using Azure App Services or VMs
GitHub Actions	Set up CI/CD pipelines for build/test/deploy
Task Tracking	Manage tasks and sprints using Jira/Trello
Documentation	Architecture diagram, deployment notes, setup guide

📦 Deliverables

- Codebases for 3 microservices
- Azure deployment URL for at least one service
- GitHub Actions workflows
- Architecture diagram (PNG or Markdown)
- Project management screenshot (Jira/Trello)

🧾 Rubric (100 Marks)

Criteria	Marks
Proper Microservice Refactoring	20
Docker Setup	15
Azure Deployment (min 1 service)	20
GitHub Actions CI/CD	15
Task Tracking via Jira/Trello	10

Criteria	Marks
Inter-Service API Design	10
Documentation	10

🚩 Milestone 4: PostgreSQL Migration, Python Chatbot Helpdesk & Final Deployment

🎯 Goal:
Migrate any MongoDB remnants to PostgreSQL, integrate a Python-based chatbot for lab FAQs, implement a notification system, and deploy the full-stack system.

🔧 Tasks & Subtasks

Task	Subtasks
Data Migration	Ensure all data in PostgreSQL
Python Chatbot	Flask + LLM chatbot for test FAQs and instructions
Notifications	Email/SMS reminders for appointments and results (mock or real)
Testing	Unit + integration testing with Jest/Vitest
Final Deployment	All services + chatbot deployed (Azure/Render/etc.)
Presentation	Slides, walkthrough video, and documentation

📦 Deliverables

- PostgreSQL dump + updated schemas
- Python chatbot (Flask-based or React-integrated)
- Working notification system (mock or actual)
- Final hosted application URL
- Test report (coverage + sample cases)
- Final PPT + video

🧾 Rubric (100 Marks)

Criteria	Marks
PostgreSQL Migration Completion	20
Python Chatbot Integration	15
Notification Feature	10
Testing Coverage & Result Accuracy	15

Criteria	Marks
Final App Deployment	15
Presentation (Slides + Video)	15
Innovation / Enhancements	10

✅ Evaluation Metrics

- Seamless booking UX
- Reliability of backend services
- Data consistency across PostgreSQL
- Helpfulness of chatbot
- Overall completeness of the pathology lab workflow