# Project Charter Document

## CureBuddy – Online Doctor Appointment System

### Purpose

The purpose of the CureBuddy project is to develop a secure, user-friendly online appointment scheduling system for healthcare providers. The platform will enable patients to book, manage, and cancel appointments, allow doctors to manage their availability and appointments, and provide administrators with tools to verify doctor accounts and monitor system usage. This system aims to streamline healthcare interactions and improve patient access to care.

### Objective

The primary objectives of the CureBuddy project include:

1. Real-time doctor availability and appointment booking.
2. Role-based dashboards for patients, doctors, and admins.
3. Secure and efficient appointment management workflows.
4. User-friendly interfaces and intuitive navigation.
5. Scalability to support future features like telemedicine and health record management.

### Opportunity

The increasing demand for digital healthcare solutions underscores the need for robust online appointment systems. Studies highlight that patients prefer digital booking for convenience, while doctors seek tools to manage their schedules efficiently. Administrators require centralized platforms to maintain service quality and compliance.

CureBuddy addresses this gap by providing a secure, reliable system to streamline patient-doctor interactions and support future growth in telemedicine and digital health.

### Business Requirements

- **User Authentication:** Secure logins for patients, doctors, and admins
- **Appointment Management:** Book, cancel, and view appointment history
- **Doctor Profile & Availability:** Doctors manage profiles and time slots
- **Admin Dashboard:** Admins verify doctors, monitor system metrics
- **Search & Filtering:** Patients search for doctors by name or location
- **Role-based Dashboards:** Customized interfaces for patients, doctors, and admins

### Technical Requirements

1. **User Authentication**

- OTP or JWT-based login and registration
- Secure password hashing and storage

2. **Appointment Management**

- CRUD APIs for booking, managing, and canceling appointments
- Real-time updates for doctor availability

3. **Role-based Dashboards**

- Patients: view, book, and manage appointments
- Doctors: manage profiles, availability, and appointments
- Admins: verify doctor accounts, monitor system usage

4. **Search Functionality**

- Search by doctor name, hospital name and location

5. **Database**

- MongoDB Atlas for secure and scalable data storage

6. **Deployment**

- Frontend hosted on Vercel
- Backend hosted on Render
- Cloud-hosted database on MongoDB Atlas

## Technology Stack

- **Frontend:** React.js, HTML, CSS, JavaScript
- **Backend:** Node.js + Express.js
- **Database:** MongoDB Atlas
- **Authentication:** OTP/JWT
- **Deployment:**
  - Frontend → Vercel
  - Backend → Render
  - Database → MongoDB Atlas

## PESTEL Analysis

- **Political:** Compliance with healthcare data privacy and security regulations.
- **Economic:** Cost-effective development using the MERN stack.

- **Social:** Growing demand for accessible and digital healthcare solutions.
- **Technological:** Modern, secure, and cloud-based infrastructure.
- **Environmental:** Cloud hosting reduces resource consumption and local infrastructure needs.

## Risk Analysis & Mitigation

| Risk | Mitigation Strategy |
|---|---|
| **Data Privacy Concerns** | Use HTTPS, secure authentication (OTP/JWT), and data encryption. |
| **Performance Bottlenecks** | Optimize queries, use caching, and conduct load testing. |
| **Over Budget Costs** | Utilize free-tier services where possible and monitor usage. |
| **Compatibility Issues** | Regular dependency updates and rigorous testing. |

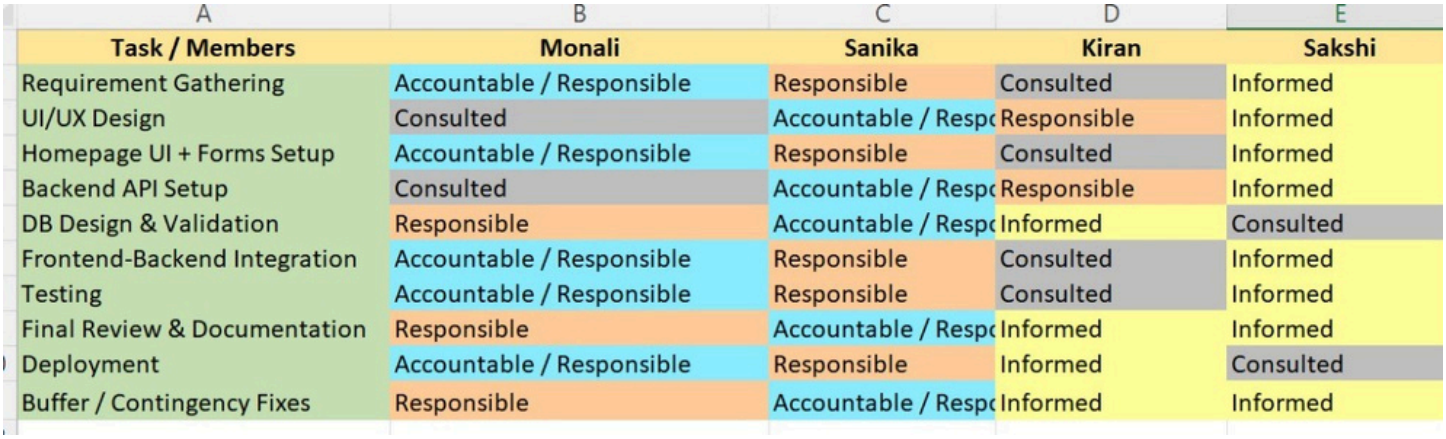
## Timeline & Milestones

| Main Phase | Subtask | Date | Actual Start | Actual End | (Actual)Start Date | (Actual)End Date | (Expected)Start Date | (Expected)End Date | Expected)Duratio | (Actual)Duration |
|---|---|---|---|---|---|---|---|---|---|---|
| Documentation | Requirement gathering & use case writing | Jun-01 | 01-Jun | 03-Jun | 01-06-2025 | 05-06-2025 | 01-06-2025 | 03-06-2025 | 3 | 5 |
| | Database design, system architecture sketch | Jun-02 | 04-Jun | 04-Jun | | | | | | |
| | Project plan & test case document creation | Jun-03 | 05-Jun | 05-Jun | | | | | | |
| UI/UX Design | Wireframe & page structure in Figma | Jun-04 | 06-Jun | 07-Jun | 06-06-2025 | 08-06-2025 | 04-06-2025 | 05-06-2025 | 2 | 3 |
| | Color theme, fonts, UX flow validation | Jun-05 | 08-Jun | 08-Jun | | | | | | |
| Frontend Development | Setup structure, homepage UI | Jun-06 | 09-Jun | 12-Jun | 09-06-2025 | 15-06-2025 | 06-06-2025 | 09-06-2025 | 4 | 7 |
| | Forms (login, register), validation | Jun-07 | 13-Jun | 13-Jun | | | | | | |
| | Navigation logic and error states | Jun-08 | 14-Jun | 14-Jun | | | | | | |
| | UI testing and debugging | Jun-09 | 15-Jun | 15-Jun | | | | | | |
| Backend Development | API setup (register/login, appointments) | Jun-10 | 16-Jun | 18-Jun | 16-06-2025 | 20-06-2025 | 10-06-2025 | 12-06-2025 | 3 | 4 |
| | DB connection, data validation | Jun-11 | 19-Jun | 19-Jun | | | | | | |
| | API testing with Postman, debug | Jun-12 | 20-Jun | 20-Jun | | | | | | |
| Integration | Connect frontend to backend | Jun-13 | 21-Jun | 22-Jun | 21-06-2023 | 23-06-2025 | 13-06-2025 | 14-06-2025 | 2 | 2 |
| | Test full workflows | Jun-14 | 23-Jun | 23-Jun | | | | | | |
| Testing & Debugging | Functional testing, all user roles | Jun-15 | 24-Jun | 26-Jun | 24-06-2025 | 28-06-2025 | 15-06-2025 | 17-06-2025 | 3 | 3 |
| | Bug fixes, security testing | Jun-16 | 27-Jun | 27-Jun | | | | | | |
| | Responsiveness, browser compatibility | Jun-17 | 28-Jun | 28-Jun | | | | | | |
| | Final check, docs review | Jun-18 | 29-Jun | 30-Jun | | | | | | |
| Final Review/Deploy | Deployment on Render or Vercel | Jun-19 | 01-Jul | 01-Jul | 29-06-2025 | 01-07-2025 | 18-06-2025 | 19-06-2025 | 2 | 2 |
| Buffer/Contingency | Reserved for unexpected bugs or delays | Jun-20 | 02-Jul | 02-Jul | 02-07-2025 | 02-07-2025 | 20-06-2025 | 20-06-2025 | 1 | 1 |

## Human Resource Table

| Role | Count | Name |
|---|---|---|
| **UI/UX Designer** | 1 | Monali Babde |
| **Frontend Developer** | 2 | 1. Sanika Kundekar<br>2. Monali Babde |
| **Backend Developer** | 2 | 1. Kiran Shinde<br>2. Sakshi Dube |
| **DB Designer** | 1 | Kiran Shinde |
| **Project Management** | 2 | 1. Sanika Kundekar<br>2. Kiran Shinde |
| **Documentation** | 2 | Creator: Sanika Kundekar<br>Reviewer: Monali Babde |
| **Tester** | 4 | 1.Sanika Kundekar<br>2.Monali Babde<br>3.Kiran Shinde<br>4.Sakshi Dube |

## RACI Chart (First Review)



| | A | B | C | D | E |
|---|---|---|---|---|---|
| | **Task / Members** | **Monali** | **Sanika** | **Kiran** | **Sakshi** |
| | Requirement Gathering | Accountable / Responsible | Responsible | Consulted | Informed |
| | UI/UX Design | Consulted | Accountable / Respo | Responsible | Informed |
| | Homepage UI + Forms Setup | Accountable / Responsible | Responsible | Consulted | Informed |
| | Backend API Setup | Consulted | Accountable / Respo | Responsible | Informed |
| | DB Design & Validation | Responsible | Accountable / Respo | Informed | Consulted |
| | Frontend-Backend Integration | Accountable / Responsible | Responsible | Consulted | Informed |
| | Testing | Accountable / Responsible | Responsible | Consulted | Informed |
| | Final Review & Documentation | Responsible | Accountable / Respo | Informed | Informed |
| | Deployment | Accountable / Responsible | Responsible | Informed | Consulted |
| | Buffer / Contingency Fixes | Responsible | Accountable / Respo | Informed | Informed |

## Resources Needed

- Documentation on React, Express, MongoDB
- Cloud accounts (MongoDB Atlas, Vercel, Render)
- Development tools (VS Code, Postman, Git/GitHub)

## Success Criteria

- Patients can book, manage, and view appointments seamlessly
- Doctors can manage availability and appointments effectively
- Admins can approve doctors and monitor system usage
- Secure, role-based dashboards and smooth performance
- Responsive across devices

## Conclusion

CureBuddy is a robust, secure, and user-friendly platform that bridges the gap between patients and healthcare providers by streamlining appointment scheduling and management. With its modern architecture and role-based access control, CureBuddy enhances the overall healthcare experience for patients, doctors, and administrators while laying a strong foundation for future expansions such as telemedicine integration.