
Implementation Document

for

E-Vaidya

Version 1.0

Prepared by

Group 12:

Aman	210104
Aniket Borkar	210135
Deven Gangwani	210327
Goutam Das	210394
Kartik Soni	210496
Narendra Singh	210649
Prashant Kumar	210750
Shrey Bansal	210997
Shubham Patel	210709
Swastik Singhal	211090

Group Name: Debuggers

aman21@iitk.ac.in
aniketsb21@iitk.ac.in
devenag21@iitk.ac.in
goutamd21@iitk.ac.in
kartiksoni21@iitk.ac.in
narendras21@iitk.ac.in
kprashant21@iitk.ac.in
shreyb21@iitk.ac.in
devang21@iitk.ac.in
sswastik21@iitk.ac.in

Course: CS253A

Mentor TA: *Anuj Shrivastava*

Date: 20/03/23



CONTENTS	II
REVISIONS	II
1 IMPLEMENTATION DETAILS.....	1
2 CODEBASE	2
3 COMPLETENESS.....	5
APPENDIX A - GROUP LOG	7

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Debuggers	Initiated the document. Added all the necessary details.	20/03/23

1 Implementation Details

- Programming languages used: JavaScript, HTML, CSS
- Runtime JS Environment: NodeJS
- Frameworks used: NodeJS, Express.JS
- Libraries used: React.JS, Bcrypt, MUI, React-Bootstrap, Bootstrap, JWT
- Database systems used: Mongo DB

We have used the MERN stack for implementation of our software. As MERN is JavaScript based, we have used JavaScript as our main programming language. One advantage of MERN stack is it uses JavaScript for both front-end, back-end and database. We found MongoDB to be more useful than other database systems as we had so much unstructured data and we also had to handle downloading of pdfs etc. Express is chosen because allows us to define a flexible route that handle HTTP methods. It also allows to easily add additional functionality to the server. It has a relatively small footprint and can handle many concurrent requests without significant performance degradation. We preferred React.JS for front end development. It provides us with the ease of creating reusable UI components. It allows for efficient state management, making it easier to create complex, dynamic applications. Node.JS in MERN stack allows for the development of server-side applications using JavaScript, which can simplify the development process by allowing you to use a single language for both the front-end and back-end of your app. It uses an event-driven, non-blocking I/O model, making it lightweight, efficient, and well-suited for real-time, data-intensive applications running across distributed devices. We have also fused bootstrap libraries for different CSS components.

2 Codebase

Provide the link to your GitHub repository.

<https://github.com/ShreyB2091/E-Vaidya>

File Structure:

- /Documents
 - Software Design Document.pdf
 - Software Requirement Specifications.pdf
 - Software Implementation Document.pdf
- /client
 - /public
 - favicon.ico
 - index.html
 - /src
 - /components
 - /Nurse
 - InitialNurse.js
 - nurse.css
 - Vitals.js
 - Vitals.css
 - /Pharmacy
 - CompletedPrescriptions.js
 - InitialPharmacy.js
 - Patient.js
 - Prescription.js
 - /Receptionist
 - Appointment.js
 - AppointmentsRequests.js
 - AppointmentStatus.js
 - DoctorAllotment.js
 - DoctorAppointments.js
 - DoctorAppointments.module.css
 - InitialReceptionist.js
 - UploadReports.js
 - ViewAppt.js
 - /Student
 - doctors.css
 - Doctors.js
 - History.css
 - History.js
 - Home.css
 - Home.js
 - request.css
 - Request.js
 - Upcoming.css
 - Upcoming.js
 - /doctor
 - Entry_page.css
 - Entry_page.js
 - Schedule.css

- Schedule.js
 - /View Appointments
 - Appointment.js
 - Conduct_appointment.js
 - DiagnosticTest.js
 - MedHistory.js
 - Medication.js
 - view_appointments.js
 - view_appointments.css
- /homepage
 - homepage.css
 - homepage.js
- /login
 - login.css
 - login.js
- /register
 - register.js
 - register.css
- Icons.js
- Navbar.css
- Navbar.js
- logo.jpeg
- logo.png
- logo_bg.png
- /state
 - /action_creators
 - Index.js
 - /reducers
 - Index.js
 - prescriptionsReducer.js
 - Index.js
 - store.js
- /states
 - /features/Appointments
 - ApptSlice.js
 - store.js
- App.css
- App.js
- Data.js
- Index.js
- Readme.md
- Index.js

Brief Explanation

- **/Documents**
 - Contains the Software Documents
- **/client**
 - **/public**
 - Contains **index.html** which is the skeleton for the app
 - **/src**
 - Contains **app.js** which provides the framework for various components
 - **/components**
 - Contains folders of various components
 - Each Folder contains files for the corresponding pages
 - These files include JavaScript files and CSS files

3 Completeness

1) Login Page:

This page is the entry page of the website for all the users. New users will register themselves initially and they can login to access E-Vaidya afterwards

(Completely done)

2) Home Page:

User-can:

- Request-an-Appointment- There will be a card for requesting an appointment that will redirect user to a new page where student can select appointment details and book appointment for HC.
- View-All-Appointment - There will be a card for requesting an appointment that will redirect user to a new page where student can see the forthcoming appointments and their details.
- View Medical History - It will also be present as card that will redirect user to a page which in a tabular form contain all previous appointment details.
- Doctor Schedule - It will also be present as a card that will open a dialog box containing a table about doctors available for a particular slot on a particular date and the current token number of the patient they are seeing.

(Completely Done)

3) Doctor's Page:

Doctor can:

- Doctor-Schedule - It will land to a page from which doctor can view his/her schedule.
- View-Appointment - This will open a new page having all the appointments of the students as clickable cards, upon choosing a card a new page will open where a doctor can view patients' medical history as well as he can prescribe the patient medical tests and medicines

(Completely done)

4) Pharmacy-Page:

- This will contain the medication details of appointments that the doctor prescribed having a detail of the prescription as clickable cards which upon clicking will open a tabular dialog box containing the info of medicine and amount to be given as well as a checkbox for viability of medicine.
- Completed Prescription Page-It will show the list of medicines given to all the patients.

(Completely done)

5) Receptionist-Page

- The page will separate clickable cards for all the function ability for the following-
- View Appointment request - It will open a new page - The page will have the request patient made online with patients' credentials and description as clickable cards which upon clicking will open doctor allotment page

- **Doctor Allotment Page** - It will contain credentials of patient of student, details of preferred doctor as well as their description and preferred slot for prescription in which the receptionist will be able to add a doctor's appointment.
- **Upload Medical records**- It will open a new page in which a receptionist can add a medical report in a pdf form to a particular patient profile.

(Completely done)

Future Development Plans:

- **Update UI:** Make changes in the UI to make the system more user friendly and fluid by providing intuitive animations and a uniform colour code to represent specific information without discrepancies.
- **Useful Links:** Provide a page with useful links related to HC. Provide a page with just info on Offline Reimbursement Plan.
- **Change Password Option:** Allows the user to change the password by providing the current password, and then asking for the new password and its retype. This changes the user password in the database.
- **Notification by Mail (whenever Status Change or Feedback received):** Whenever the reimbursement status changes, the patient user who filled the form is automatically notified through their mail.
- **Manage Notification:** Provides the option to toggle the notifications ON or OFF, or even selectively enable the notifications.
- **Provide a multiple feedback system:** A feedback system which allows users with any roles to discuss by multiple comments on a transaction.
- **Create a Draft Form:** Allows the user to temporarily save the form details without having to discard the already written data.

Appendix A - Group Log

Sr. No.	Date	Activity	Participants
1	26-2-2023	Distributed work among the team and planned the implementation process	Swastik, Kartik, Aniket, Shubham, Goutam, Shrey, Deven, Narendra, Prashant, Aman
2	01-03-2023	Discussed on development of back-end and brief details about the design of application	Swastik, Kartik, Aniket, Shubham, Goutam, Shrey, Deven, Narendra, Prashant, Aman
3	09-03-2023	Everyone reported about the progress of the work they were assigned and further work distribution was done.	Swastik, Kartik, Aniket, Shubham, Goutam, Shrey, Deven, Narendra, Prashant, Aman
4	13-03-2023	Added basic CSS of webpages, Implemented remaining React.js for pages	Swastik, Kartik, Aniket, Shubham, Goutam, Shrey, Prashant, Aman
5	15-03-2023	Discussed regarding implementation of database and started the implementation	Kartik, Aniket, Goutam, Aman, Shubham, Prashant Shrey
6	18-03-2023	Discussed and resolved problems faced in integration of backend with frontend started resolving them and added minor changes to CSS	Swastik, Kartik, Aniket, Shubham, Goutam, Shrey, Prashant, Deven, Narendra, Aman
7	20-03-2023	Finished the Software Implementation Document and resolved queries in application	Swastik, Kartik, Aniket, Shubham, Goutam, Shrey, Deven, Prashant, Aman, Narendra