Project Title: weCare

Theme: Healthcare management system
Teaching Assistant Name: Aman Sinha
Date: 25-03-2022
Version Number: 1
Group Number: 46
Group members:

Name	ID
Purav Kansara	201901032
Varun Rathod	201901111
Yash Khanpara	201901252
Aditya Srivastava	201901265
Nilay Patel	201901281
Shubham Kevadiya	201901284

System Description:

weCare health management system is a hospital web-application which features patient and doctor portal, booking lab test, scheduling an appointment with doctors, doctor profile, patient's health profile, various health-departments, and machinery information for surgeries and critical care, Career portal with department wise job openings.

Development Technology:

MERN stack - Mongo DB, Express js, React js, Node js, CSS, HTML

Objectives:

- Increasing the level of collaboration between the doctor and the patient.
- Clinical data is successfully processed by the billing and insurance processing modules, resulting in an efficient payout for health visits.
- To facilitate the management of digital information related to the health status of the patients, health infrastructure, and health services.
- To facilitate transparency and maintenance of standards in Counselling for medical education

- To provide the best medical facilities and infrastructure at an affordable price.
- To meet and exceed the requirements and expectations of the community by providing coordinated, compassionate and high-quality health care services at affordable cost.
- Patients can schedule appointments with doctors online or in person at the hospital.
- It should be able to handle the test reports of patients conducted in the pathology lab of the hospital.

Functionalities:

- One can register his/herself by ‘login -> sign up’ to the website.

  - **Non-registered users:** Non-registered users will be able to see the homepage of the website, which provides basic information like
    - Different Articles
    - Infrastructures and their services
    - Different department-wise doctors,
    - Emergency contacts,
    - Details related to website functionalities.
    - Career portal with department-wise job openings.
    - Buy medicines online from drug inventory.
    - The user can book an RT-PCR Lab Test.
    - The user can also apply for the Covid vaccine dose and the booster dose.
    - Users can see current Daily Covid cases, city-wise, state wise.
    - Users can also see vaccination drive coverage.

  - **Registered users:** Any registered user can sign in from the main page, based on the credentials they will be able to see the following:

  - **Doctor page**:
    - Appointment scheduler for Doctors who see patients on a daily basis.
    - It enables doctors to plan and browse their schedules. It is also possible to share such information with patients in order to schedule an appointment.
    - Doctors can recommend different doctors according to the patient’s case and symptoms.
    - List of patients appointed to the doctor.
    - Details of patients like medical history and drug prescription.
    - Lab reports of patients like blood tests, urine test, CT scan, MRI, etc.
    - Doctors can add any new medicines prescribed to a particular patient.
    - Portal for doctors to enter a patient's medical records.

  - **Patient page**:
    - Consult a doctor according to the symptoms.

        - Patients can book a lab test.
        - The patients can see medical details pertaining to only him/her.
        - Patients can access health and wellness pages for health tips and diet.
        - Patients will be notified of their appointments and periodic medical checkups.
        - Graphical representation of patient’s reports like blood pressure, pulse, heart rate, and sugar test.
        - Patients can view the generated bill and can make the payment online through net banking, Credit-Debit Card, etc.
        - Patients can consult a doctor online according to/her symptoms.

    - **Pharmacy page**:
        - The pharmacy page can be accessed only by the chemist. Chemists need to be logged in to manage medical inventory.
        - The chemist will maintain medicine that is about to sustain selling for about two weeks.
        - The chemist can add new medicine or can delete medicine that is expired.
        - The Chemist can query medicine based on the name, brand, weight.
        - The chemist can order medicine that is below the threshold value from the stack.

- **Non-functional Requirements:**
    - Security: Any user that uses the system must have a Login ID and password (i.e has a verified account). Otherwise, users will be able to see the website's first page, which contains basic information such as different articles and their emergency contacts, and information on website features.
    - Privacy:
        - Doctors: Doctors can only consult with patients who have been assigned to them. Doctors are restricted from sharing details of one patient with another.
        - Patients: Patients are restricted to view other patients’ details.
    - Durability: In the event that the system fails, the data should be unaffected.
    - Capacity: The capacity of the site should serve multiple requests from the patient and doctors.
    - Scalability: Adding extra doctors/patients to the records should not involve any complicated procedures.
    - Compatibility: This website should run on operating systems like windows, android, macOS platforms.
    - Performance: The system should still fulfill the minimum performance criteria under increased demand scenarios.
    - Availability: The system should be operational 24 hours a day, seven days a week.
    - Reliability: The system should work flawlessly every time.

Milestone 1:

- Formulation of the problem
- i. Reading relevant background information
- ii. Understanding and documenting the requirements
- Designing a solution, documentation
- DFD Modelling and UML diagrams

Milestone 2:

- Coding and unit testing
- Documentation

Milestone 3:

- Testing
- Reviews
- Re-work and de-bugging

Estimated Total Time:

Activities	**Estimated Time**
a) Formulation of the problem	7 Hrs (total of next 3 sections)
a.1) Reading relevant background information	2 Hrs
a.2) Understanding and documenting the requirements	4 Hrs
a.3) Discussions	1 Hrs
b) Designing a solution, documentation	7 Hrs
c) Relevant learning	3 Hrs
d) Coding and unit testing	50 Hrs
e) Documentation	3 Hrs
f) Testing	4 Hrs
g) Reviews	5 Hrs
h) Re-work and debugging	11 Hrs

Hardware requirements

- Server Specifications (Estimated for 10,000 users)

	RECOMMENDED	MINIMUM
OPERATING SYSTEM	Windows Server 2012	Windows 10
PROCESSOR	Intel Xeon Processor – or – Higher	Intel i7 Processor – or – Higher
MEMORY	8 Gb (or more)	4 Gb
SCREEN	1280 x 1024 resolution (or higher)	1024 x 768 resolution
NETWORK	1000mbit (Gigabit) wired network	100mbit wired network

Software requirements

Web Browser Specifications:

Browser	Self-Hosted Technical Requirement	Cloud Technical Requirement
Chrome	v89+	v89+
Firefox	v91+	v91+
Safari	v12+	v14.1+
Edge	v44+	v44+

Technology / Architecture (attach a separate document, if necessary)

- HTML
- CSS
- Javascript
- Reactjs
- Expressjs
- Nodejs
- Mongodb
- ReqView
- Tiaga

Standard to be followed:

We are going to follow agile Development Model.
The segmentation of the entire project into smaller pieces helps in reducing project risk and total project delivery time requirements. Each iteration comprises a team going through the whole software development life cycle, including planning, requirements analysis, design, coding, and testing, before presenting a final functional product.

Agile Testing Methodology: Scrum
Agile Management Tool: Tiaga