**Intelligent Model for detecting disinformation in Fake News**

# Abstract

In the recent years, the topic of fake news has experienced a resurgence of interest in society. The increased attention stems largely from growing concerns around the widespread impact of fake news on public opinion and events Based on the research it is noted that with massive growth of online communication in the future the potential of people to deceive through computer mediated communication has also grown. Such deception can have deleterious impact and far-reaching results, also causing a political upheaval in world politics. The topic of fake news has not only received public attention but also drawn increasing attention from the academic community. We present a robust and efficient solution which has ability for detection of fake news from variegated sources. We focus on developing strong computational model to work diligently, so as to provide efficient reports to the end user. The user gets most accurate information like which sources generates which category of fake news with faster computation for better credulity., We mainly focus on feedback based and content-based approach for fake news detection which includes news articles/social media tweets, from all renowned news websites, social media platform. We provide a solution that outline promising direction for future result.

# Background

The usage and meaning of the term fake news has evolved over time. A Google Trends Analysis of the term reveals a sudden burst in popularity around the time of the US Presidential Election in 2016 by Alcott and Gentzkow 2017 revealed that fake news was widely shared during the three months prior to the election with 30 million total Facebook shares of 115 known pro-Trump fake stories and 7.6 million of 41 known pro-Clinton fake stories .Another example which is major rise in tumult in 2011 which noted the false report of bankrupt of United Airlines s parent company in 2009 caused the stock price to drop by as much as 76% in a matter of minutes .The Fake news term has evolved and become synonymous with the spread of false information. Fake news has generally been defined as “a news article that is intentionally and verifiably false” or “information presented as a news story that is factually incorrect and designed to deceive the consumer into believing it is true".

The intent of deception, and do not capture the broader scope of the term based on its current usage. This definition allows us to capture the different types of fake news identified in which can be differentiated by the means employed to falsify information, such as fabricated content (completely false), misleading content (misleading use of information to frame an issue), imposter content (genuine sources impersonated with false sources), manipulated content (genuine information or imagery manipulated to deceive), false connection (headlines, visuals or captions that do not support the content), and false context (genuine content shared with false contextual information). The definition also allows us to include different types of fake news identified by their motive or intent, such as malicious intent (to hurt or disrepute), profit (for financial gain by increasing views), influence (to manipulate public opinion), sow discord (to create disorder and confusion), passion (to promote ideological biases), amusement (individual entertainment) [126]. We can also subdivide false information by intent as misinformation and disinformation. Misinformation refers to unintentionally spread false information which can be a result of misrepresentation or misunderstanding stemming from cognitive biases or lack of understanding or attention; and disinformation refers to false information created and spread specifically with the intention to deceive [54]. Another type of information that might be closely connected to fake news is satire - satire presents stories as news that might be factually incorrect, but the intent is not to deceive but rather to call out, ridicule, or expose behaviour that is shameful, corrupt, or otherwise “bad". This factors can have a detrimental fallacious impact in all aspect of our life hence requires a robust and scalable approach to vanquish the cause of this issue.

*Flow of Document: (Table with Section Name and Description) (On own)*

|  |  |
| --- | --- |
| Section | Description |
| Section 2 | Software Project Management Plan (SPMS) |
| Section 3 | Software Requirement Specifications (SRS) |
| Section 4 | Software Design Description (SDD) |
| Section 5 | Software Test Document (STD) |

# Sections Common to all Documents

This will contain information about the contents of the common sections present in all the documents.

For instance, Cover Page (Present in Every Document)

So, what all will be present on a cover Page should be mentioned and described in Short.

Other potential Common Sections – References/Terminologies/Appendix/Definitions/Acronyms

# Software Project Management Plan (SPMP)

## Introduction

### Project Overview

Disinformation detection system is a system for detecting disinformation in fake news. It is s a web based application that enables the users from field of information security, information law and also normal users to determine the disinformation in the news articles and provides awareness so as to prevent getting fallacious information about the given topic Disinformation identification System mainly works to identify the fabricated content in the news and provide adaptive based approach to adapt itself and thereby enabling itself to a state as to determine with confidence that the given information generated from this source is of high chance being fake using machine learning .We are following an hybrid based approach which is and ensemble and feedback based approach by analyzing transitive relations and two level ensemble approach so as the domain in determining fake news can work on variety of news as an input .We focus on providing accurate results as of proving scrapping data from various sources so as to provide wider category for classification.

The purpose of this document is to present a detailed description of the Application. It will explain the purpose and features of the system along with its interfaces, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the users and the developers of the system. The main purpose of the application is to gather all information from various news websites or from various social media platforms needed, in background so as to generate reports from our strong and robust machine learning models so as to provide diminutive overhead to the end users by providing the result what they need. The system also provides a NoSQL Database to store the information and add the news sources or articles which are detected fake inside the database so as the user can be presented with an graphical analytical user interface about fake sources or articles detected by the application. This is a single all in one application to the end user, which prevents the user to surf over the internet to guess whether a information is true or false, so that the end user can come to single platform to identify whether a given information is fake or correct by providing most accurate and relevant results .The Application is on web based platform on various OS support and utilize the power of AWS cloud for faster processing using EC2 instances and AWS Sagemaker for faster training of our machine learning models. It can be used by a broad spectrum of people in field of Information Security, Information Law, Normal Users, Business Stakeholders to manage for personal use or even for their organization. SPMP document is created to list the intended audience and provides suggestions for the same. This document is mainly for both developers and project manager to determine the procedure in the creation of further phases of the application and determine a road map or work flow for the same

### Project Deliverables

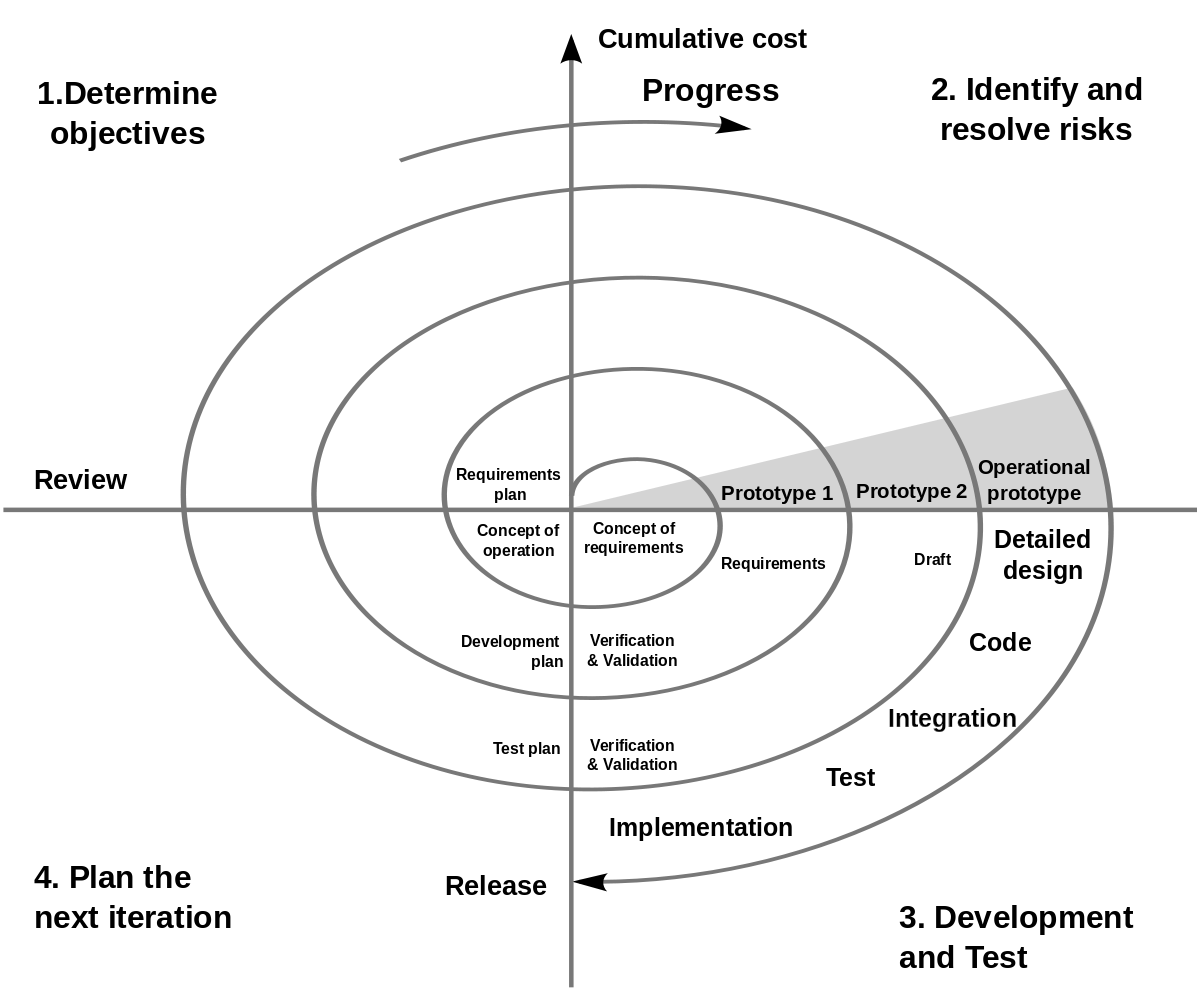
* Analysis Report Management
  + Fake News Report Generation
  + Updating new reports for various new sources
* Main dashboard that shows the details about each source and what category of disinformation detected from the source.
* User’s request message integrity
* Warning feature for the source which is detected fake by our platform

Intelligent disinformation detection system will be a cloud-based solution that will be deployed on AWS/Azure server. Additionally, for lower latency and we are making use of elastic load balancer on cloud so as the end-user never gets any delay I request processing and corresponding result display.

## Project Organization

### Software Process Model

Intelligent Model for detecting disinformation in Fake news is a project that demonstrates its need in the current scenario and also provides features which are more streamlined and efficient approach than before. In order to achieve this the application needs to approved by the new kinds of trends in sources and strategies for fake news generation. For the same, their inputs are taken at the highest level of seriousness. We use the Spiral Software Process Model in order to keep a constant sync in process of development along scalability and software maintenance .



The SPIRAL process model used for Disinformation in fake news will consist of mainly 2 to 3 iterations. Each iteration will consist of Research, Planning, Development, Review and Deployment phases. The Research phase will consist of the Literature Review and Requirement survey and analysis. Followed by the Planning phase which will include the designing of System Architecture and Documentation. Then comes the Development phase which include some research on development tools and the actual development of the software. The penultimate phase is the review phase where testing of software is done, and based on the issues the system architecture is re planned or the implementation of the software is improved. After this the cycle is repeated and the Deployment stage is reached when the software is working as expected and there are no issues generated during the testing phase.

### Roles and Responsibilities

* + - Project Manager – Vedang Parasnis
      * Description - Project Manager is responsible for the timely execution and completion of the project. She will work with all the group members and will see that every group member is performing his/her task. She will communicate with the Faculty and inform him/her about the development of the project. She can schedule group meetings to look over the development of the project.
    - Developer – Hritik Jaiswal, Heet Sakaria, Vedang Parasnis
      * Description - Developer will code for applications and programs for backend processing systems to build a working project as proposed by the manager and team. Once the core of the software is developed, the software is passed on to the next Team member that is to the Tester.
    - Designer – Hritik Jaiswal
      * Description - Designer deals with the look and feel of the software. Designer’s task will be to work hand-in-hand with the developer and help him with creative styling ideas, improving frameworks to make the project more user-friendly.
    - Tester – Vedang Parasnis, Heet Sakaria
      * Description - Tester’s role will be to perform checks on the services provided by

the software, to see if they are functioning properly or are bugged for a given

condition.

* + - Analyst – Heet Sakaria, Hritik Jaiswal
      * Description - Project analyst is responsible for managing the development of project through special research, data analysis, and data collection to facilitate strategic decision-making.

### Tools and Techniques

For the development of the product following tools will be used.

* + - The front-end of the web application is developed using HTML / Bootstrap CSS framework.
    - TypeScript for front-end development
    - Express Js using NodeJS runtime and Flask will be used for back-end development and serve as an application server
    - Python Scrapy and Beautifulsoup is used for web scrapping, News Api is used for information gathering
    - Firebase cloud Storage will be used as the database for storage of application data.
    - Visual Studio IDE / Sublime Text, Jupyter Notebook, Google Colab will be used for developing the application and testing of machine learning models.
    - Aws EC2 Instances will be used for faster computation of application server request
    - AWS Sagemaker will be used for faster training of Machine Learning models on the cloud
    - StarUML will be used for creating UML diagrams.

## Project Management Plan

### Tasks

#### Documentation

Requirement Analysis/Writing SRS

1. Description:

To get all the requirements from the client and after proper analysis, forming the final SRS document.

1. Deliverables and Milestones:

* Listing all the requirements.
* Performing proper Requirement Analysis on it.
* Forming the SRS document.
* After some revisions finalizing the SRS.

1. Resources Needed

* MS Word

1. Dependencies and Constraints
   * None
2. Risks and Contingencies

* Only risk here is if any requirement is missed out. Which can be solved by communicating with client on every stage of the development of the project.

Designing System Architecture

1. Description

Designing the system architecture of the project which include on deciding the file structure, database to be used, language to be used for development and communication between all the components.

1. Deliverables and Milestones

* Designing Conceptual Architecture.
* Designing Detailed logical Architecture.
* Verifying internal design of each component and interaction among components.
* Finalizing the System Architecture Design.

1. Resources Needed

* UML Design Software

1. Dependencies and Constraints

* Requirements must be clear before designing the Architecture.

1. Risks and Contingencies
   * Frameworks used should have a continuous long-term support and receive security updates

#### Development

User Access Management

1. Description

Developing the User Access Management which will give users access based on the person’s designation. Users will be divided into user groups, each group with different read/write access. This permission is set and managed by the website admins.

1. Deliverables and Milestones

* Defining user groups.
* Mapping of users to user group.
* Defining access control for each user group

1. Resources Needed

* PostgreSQL
* ReactJS
* Django REST API

1. Dependencies and Constraints

* Getting designation wise what data can be accessed by whom.

1. Risks and Contingencies
   * Users being able to view multiple roles. Risk is mainly post development and will be identified in the testing phase

Hospital resource monitoring setup

1. Description

Developing a monitoring system which keeps record of resources in Hospital.

1. Deliverables and Milestones

* Defining each and every resource item.
* Developing an interface which provides CRUD functionalities for resource management.
* Developing live dashboard for statistical representation of resources which shows the current quantities of resources.

1. Resources Needed

* PostgreSQL
* ReactJS
* Django REST API

1. Dependencies and Constraints

* Data dependency

1. Risks and Contingencies
   * Data Unavailability: This can be mitigated upto a certain amount by taking generic data points from research done. In the further stages this can become a major issue if not rectified
   * Data Cleaning: Depending on the quality of data, the structuration will be possible. Primary/Composite keys are a must and need to present in order to work efficiently.

Patient and Report monitoring

1. Description

Developing a monitoring setup which keeps track of patient data and reports.

1. Deliverables and Milestones

* Developing an interface which provides CRUD functionalities to interact with patient data or report data.
* Mapping individual patient to respective report data.
* Developing live dashboard for statistical representation of resources which shows the current number of patients with their status which can be admitted, discharged or dead.

1. Resources Needed

* PostgreSQL
* Node js
* React JS
* Django REST API

1. Dependencies and Constraints

* Data dependency

1. Risks and Contingencies
   * Data Unavailability: This can be mitigated upto a certain amount by taking generic data points from research done. In the further stages this can become a major issue if not rectified
   * Data Cleaning: Depending on the quality of data, the structuration will be possible. Primary/Composite keys are a must and need to present in order to work efficiently.

Predictive Model

1. Description

ML based prediction for the status of patient, future inventory required

1. Deliverables and Milestones

* ML model for patient progression
* ML model for inventory requirement

1. Resources Needed
   * Python
   * Azure ML Studio
2. Dependencies and Constraints
   * Azure
3. Risks and Contingencies
   * Data Availability: Attributes that need to be predicted should be well documented and available for the training of the model.

Software Testing

1. Description

Testing the software developed at each stage of development.

1. Deliverables and Milestones

* Performing Django Unit testing on individual component developed.
* Performing Lighthouse testing for checking performance of software.
* Implementing Continuous Integration to maintain that code change does not break testing
* Security testing to ensure that application does not leak data in any manner possible
* Extensive Selenium based tests to keep track of UI/UX actions working correctly even under load
* Creating final test report after final testing of software.

1. Resources Needed
   * TravisCI
   * Lighthouse
   * Sucuri/Pentest-tools
   * Selenium
2. Dependencies and Constraints
   * None
3. Risks and Contingencies
   * None

Software Deployment

1. Description

Deploying the application on an online website/ google chrome store, with a domain name, and made available around the clock, publicly for website and associated rules for the.

1. Deliverables and Milestones

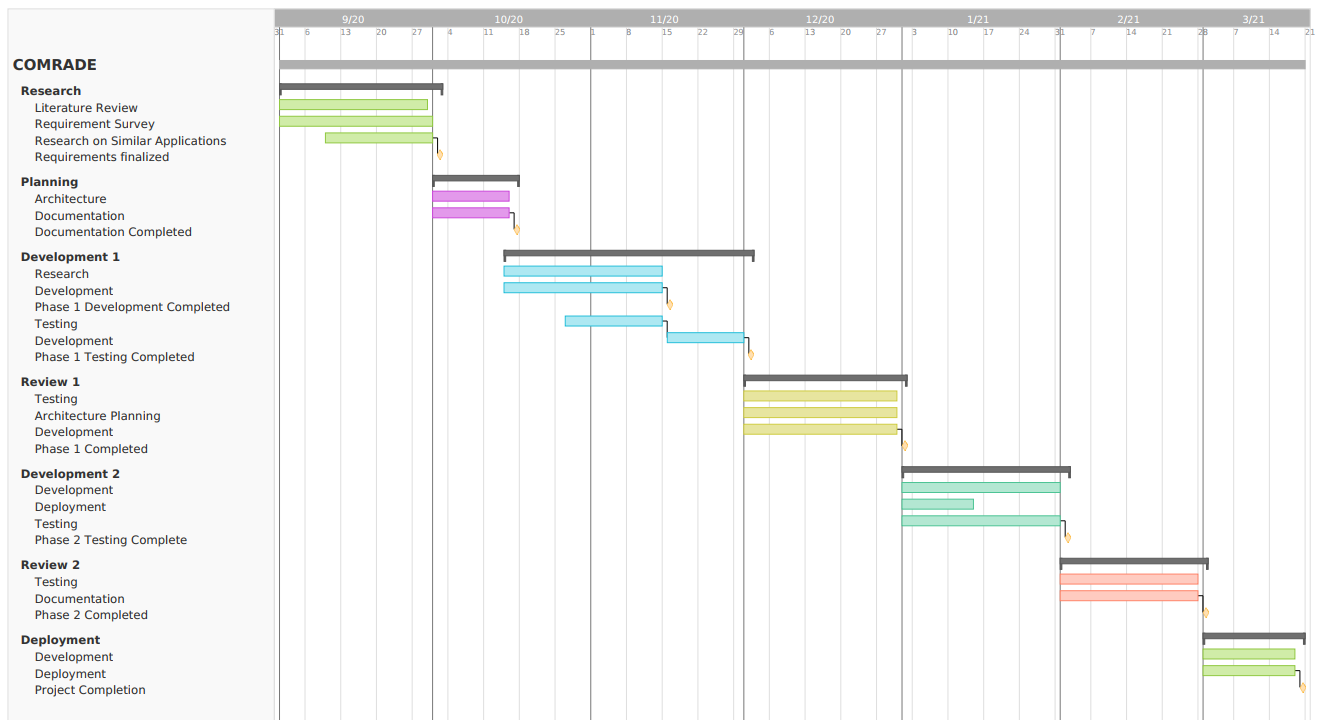
* Online Hosting with necessary measures

1. Resources Needed
   * AWS/Azure server
   * SSL certificate
   * Domain name
2. Dependencies and Constraints
   * AWS/Azure
3. Risks and Contingencies
   * Required domain name isn’t available: Solution isn’t available for this, a new different domain name will have to be sought and used.

### Assignments

|  |  |  |
| --- | --- | --- |
| Sr. No. | Task Name | Member Associated |
| 1 | Requirement Analysis/Writing SRS | Viraj Thakkar, Aditya Vedpathak |
| 2 | Designing System Architecture | Aditya Panchal, Abhishek Patel |
| 3 | User Access Management | Aditya Vedpathak |
| 4 | Hospital resource monitoring setup | Aditya Panchal |
| 5 | Patient and Report monitoring | Aditya Vedpathak |
| 6 | Predictive model | Viraj Thakkar, Aditya Vedpathak |
| 7 | Software Testing | Abhishek Patel, Aditya Panchal |
| 8 | Software Deployment | Viraj Thakkar, Abhishek Patel |

### Timetable



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Task Name** | **Start Date** | **End Date** | **Start on day** | **Duration (Days)** | **Member associated** |
|  |
| **Research** | |  |  |  |  |  |  |
|  | Literature Review | 01/09/2020 | 29/09/2020 | 0 | 29 | Aditya Vedpathak |  |
|  | Requirement Survey | 01/09/2020 | 30/09/2020 | 0 | 30 | Abhishek Patel |  |
|  | Research on Similar Applications | 10/09/2020 | 29/09/2020 | 9 | 20 | Viraj Thakkar, Aditya Panchal |  |
| **Planning** | |  |  |  |  |  |  |
|  | Architecture | 01/10/2020 | 15/10/2020 | 30 | 15 | Aditya Panchal, Aditya Vedpathak |  |
|  | Documentation | 01/10/2020 | 15/10/2020 | 30 | 15 | Viraj Thakkar |  |
| **Development 1** | |  |  |  |  |  |  |
|  | Research | 15/10/2020 | 14/11/2020 | 44 | 31 | Viraj Thakkar |  |
|  | Development | 15/10/2020 | 14/11/2020 | 44 | 31 | Aditya Panchal |  |
|  | Testing | 27/10/2020 | 14/11/2020 | 56 | 19 | Abhishek Patel |  |
|  | Development | 15/11/2020 | 29/11/2020 | 75 | 15 | Aditya Vedpathak |  |
| **Review 1** | |  |  |  |  |  |  |
|  | Testing | 30/11/2020 | 30/12/2020 | 90 | 31 | Viraj Thakkar, Abhishek Patel |  |
|  | Architecture Planning | 30/11/2020 | 30/12/2020 | 90 | 31 | Aditya Panchal, Abhishek Patel |  |
|  | Development | 30/11/2020 | 30/12/2020 | 90 | 31 | Aditya Vedpathak, Aditya Panchal |  |
| **Development 2** | |  |  |  |  |  |  |
|  | Development | 01/01/2021 | 31/01/2021 | 122 | 31 | Aditya Panchal, Aditya Vedpathak |  |
|  | Deployment | 01/01/2021 | 15/01/2021 | 122 | 15 | Viraj Thakkar |  |
|  | Testing | 01/01/2021 | 20/01/2021 | 122 | 20 | Abhishek Patel |  |
| **Review 2** | |  |  |  |  |  |  |
|  | Testing | 01/02/2021 | 28/02/2021 | 153 | 28 | Abhishek Patel, Aditya Panchal |  |
|  | Documentation | 01/02/2021 | 28/02/2021 | 153 | 28 | Viraj Thakkar, Aditya Vedpathak |  |
| **Deployment** | |  |  |  |  |  |  |
|  | Development | 01/03/2021 | 18/03/2021 | 181 | 18 | Aditya Panchal, Aditya Vedpathak |  |
|  | Deployment | 01/03/2021 | 18/03/2021 | 181 | 18 | Viraj Thakkar, Abhishek Patel |  |

# Software Requirements Specifications (SRS)

## INTRODUCTION

### Product Overview

COMRADE is a software which will be used by to visualize the patient data, manage and add new patients and their reports along with predictions about inventory utilization based on disease progression.

The purpose of COMRADE is to help hospitals to better manage patient data as well as for Resource management. Also develop a prediction model based upon patient’s condition to predict the further requirement of resources. And also maintain a live dashboard for displaying statistical representation. Primarily targeted towards hospitals dedicated to COVID patients, COMRADE is geared for better and faster data management. So that hospitals require less man power in doing administrative work, which would help for better handling of patients.

The application is to be developed in the format of a progressive-web-application (PWA). This will enable for use across all device sizes, be it a mobile, tablet, laptop and PC. Making it as user friendly as possible will be another area of focus in order for hospital staff to adopt it with little to no learning curve involved.

## SPECIFIC REQUIREMENTS

### External Interface Requirements

#### Hardware Interfaces

* 1 AWS/Azure server to host the system
  + 1+ Core
  + 1GB+ RAM
  + 8GB+ SSD/HDD Storage
* Ports open:
  + 443 for HTTPS,
  + 5432 for Postgres,
  + 22 for SSH

#### Communication Protocols

* HTTPS Protocol for serving website
* TCP/IP for PostgreSQL
* SSH for accessing the web server

### Software Product Features

1. Visualise patient data, manage and add new patients and their reports

* Description

Adding CRUD functionalities for patient data and report data.

* Input

Patient and Report data.

* Output

CRUD operations on single clicks

* Functions

Create, Read, Update and Delete operations.

1. Inference derivation and correlation based upon patient data
   * Description

Analysing patient data and forming proper insights from it.

* + Input

Patient and Report Data

* + Output

Insights: relationship among patient data attributes.

* + Functions

The data will be analysed in the search of insights.

1. Digitisation of Records from current on-paper format
   * Description

Converting image of a document into a digital document.

* + Input

Image of documents.

* + Output

Digital Document.

* + Functions

Using image processing the image would be digitalized.

1. Inventory prediction based upon progression of COVID progression in patient

* Description

Based on the hospital inventory data and the usage of resources by the COVID patients, it will predict what will be the state of inventory in any time near future.

* Input

Patient and Inventory data.

* Output

State of inventory in near future.

* Functions

With the use of Machine Learning, Regression would be performed on the patient and inventory data for further prediction.

1. A patient trend spotter where every data about every patient in the hospital can be analysed and shown in interactive charts that will also show the progression of a disease in a patient.
   * Description

Analysing individual patient data to recognize trend in patient’s health conditions which would be further shown in the form of interactive charts.

* + Input

Patient data.

* + Output

Interactive chart of the trend spotted in progression of patient data.

* + Functions

With the use of Machine Learning, Regression would be performed on the patient data to spot trend in progression of patient’s health condition.

1. It will keep track of the hospital inventory (medical and equipment) along with the option to corroborate with other hospitals in cases of emergency requirements.
   * Description

An inventory tracker which will keep track of the hospital inventory and in case of emergency, it will also know which nearby hospital can help it with their requirements.

* + Input

Inventory data of all hospitals involved in project.

* + Output

An interface which will show the current status of hospital inventory. And a list of hospitals from where the deficit inventory item can be obtained in case of emergency.

* + Functions

A tracking system to keep track of inventory of hospitals.

1. A dashboard that will allow the hospital to view their performance, trends and statistics in one glance.
   * Description

A live dashboard which will show the statistics based on the hospital data.

* + Input

Hospital data.

* + Output

Graphical representation of hospital performance and trends spotted in that data.

* + Functions

With the use of Machine learning trends are spotted in Hospital data and show the data using graphical representations.

### Software System Attributes

Explain how our Software satisfies these attributes

1. Reliability

Since we are using SQL database, it follows ACID properties namely Atomicity, Consistency, Isolation and Durability which makes the data reliable.

1. Availability

Since the project will be deployed on the AWS or Azure Server, the web application will be live all the time.

1. Security

Since the deployment will be done after doing the security testing, so the security vulnerabilities are well handled. Also because of the use of well-defined access control there is no compromise on the data by an unauthorised access.

1. Maintainability

Since the web application is a dynamic PWA, therefore the maintenance cost is not increased with the increase of data. Therefore, maintaining it is very easy.

1. Portability

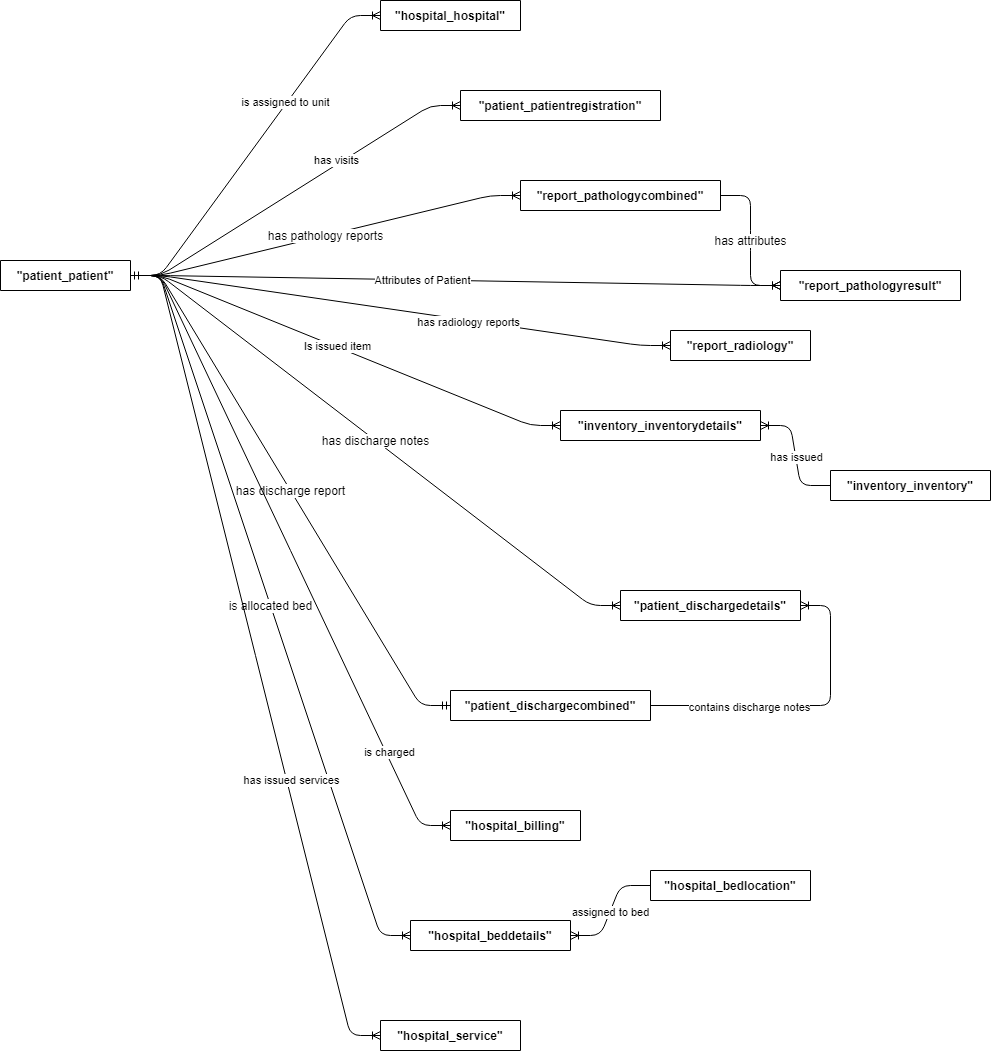
Since it is a PWA, therefore it can be accessed from any device.

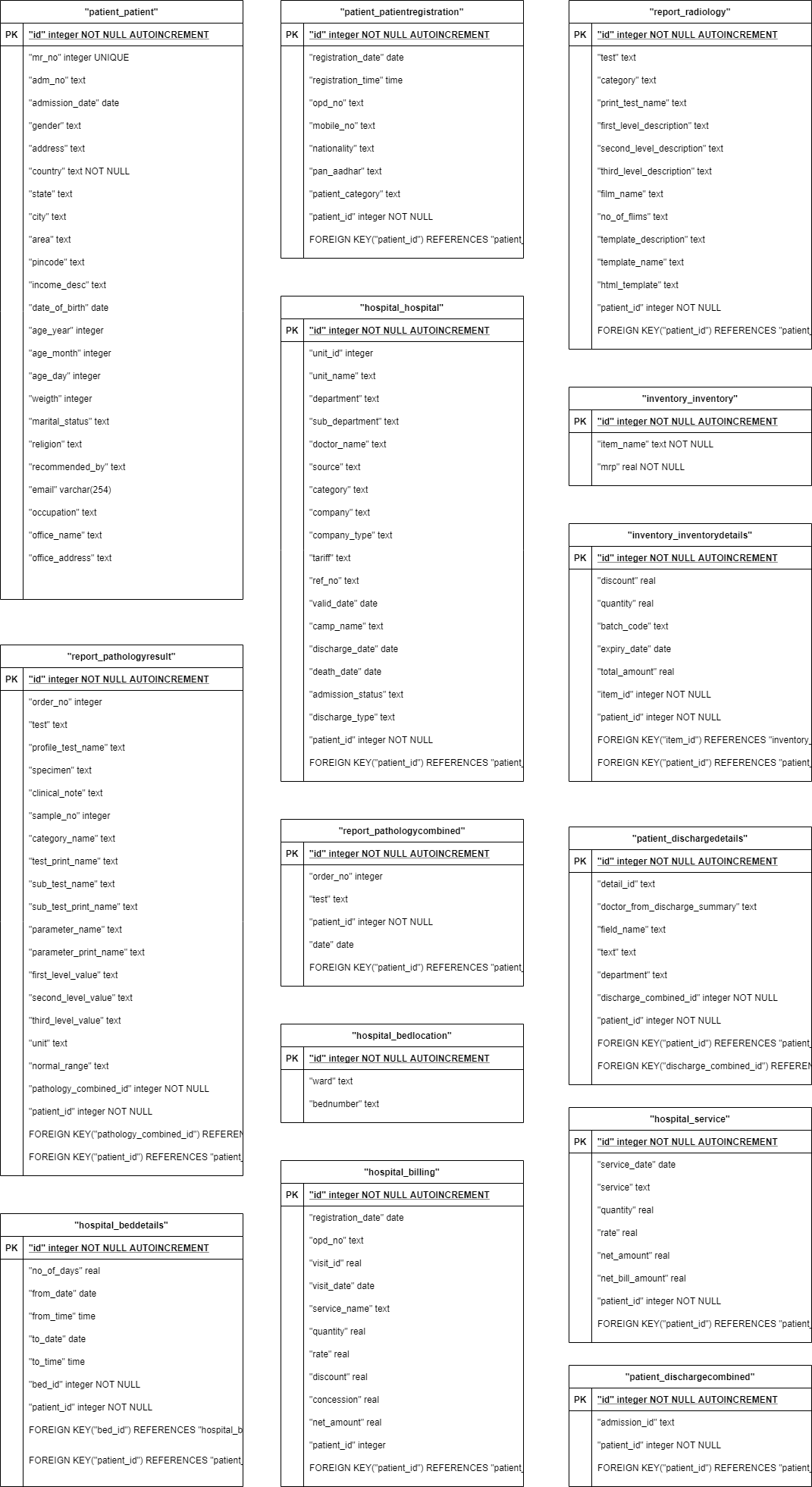
1. Performance

As it’s a PWA, the data is loaded dynamically therefore the increase in data won’t affect the performance of the application.

### Database Requirements

Database is required for storing patient and hospital data. DBMS chosen is PostgreSQL.





# Software Design Description (SDD)

## INTRODUCTION

### Design Overview

COMRADE is implemented using the REST API architecture. This architecture uses different type of request method types to differentiate among the operations to be performed on a particular model based upon the current state of the system. The operations to be performed mainly consists of the CRUD functions.

### Requirements Traceability Matrix

// To Confirm with ma’am

*Provide a matrix showing where each feature identified in the SRS is supported by the design components. (Way too complex for us, but we will break it down into multiple fronts, one for CRUD, one for prediction, that kind of stuff)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Component 1 | Component 2 | Component 3 | Component 4 |
| Requirement 1 |  | X |  | X |
| Requirement 2 | X | X | X | X |
| Requirement 3 |  |  | X |  |
| Requirement 4 | X |  |  | X |

## SYSTEM ARCHITECTURAL DESIGN

### Chosen System Architecture



The architecture chosen is the Django REST API architecture along with React JS as web framework. Django REST API makes getting data from models as simple as dealing with third party API’s. So it is used to make customized API service based on the project.

React is used for writing the fronend whereas Django REST API acts as a third party API which handles the server side and provides data to the Front end.



The above diagram shows the general work flow of the architecture. The request made to the web application first goes to the Django’s URL Patters based on which the view function is called. The view function has the responsibility of showing data on the screen. The view function makes a request to django REST API to get the data to be displayed. Now the Serializer based on the request made to it runs the ORM on the models to get the required data. So the data is obtained from the models and passed to the view function. The view function pushes the data on the HTML template to be displayed based on the URL pattern and so the response is created.

A close up of a sign

Description automatically generated

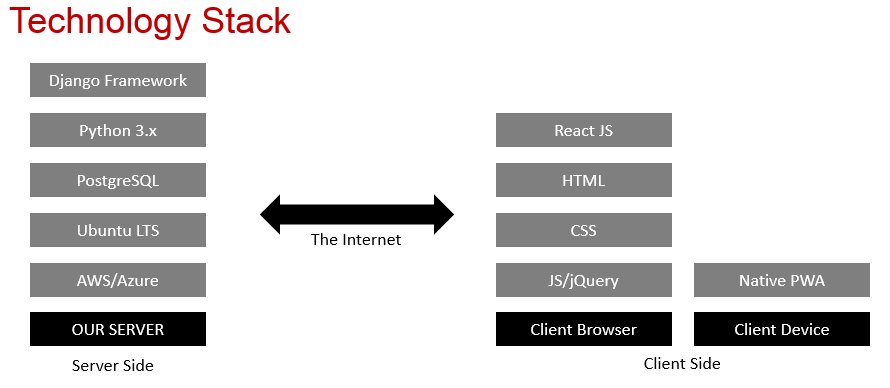
### Discussion of Alternative Designs

NON-REST Frameworks

The alternate option instead of using RESTful approach was using RESTless approach which is also known as the SOAP (Simple Object Access Protocol) approach. RESTful web service support carious data formats like HTML, JSON, text etc. whereas the RESTless web services support XML format. One other difference between RESTful and RESTless Web Service is that the RESTful services use URL to expose business logic while the RESTless services use the service interface to expose business logic. Moreover, RESTful services are easier and flexible than RESTless services. Resources and bandwidth is also an important difference between RESTful and RESTless Web Service. RESTful web services consume less bandwidth and resource while RESTless web services consume more bandwidth and resources.

### System Interface Description

*Describe the system interfaces in detail: O/S interface, files, networking, libraries, graphics libraries etc. (Describe the user interface in section 4 – PPT in LY Project)*



## DETAILED DESCRIPTION OF COMPONENTS

### Component-n

*For each component, the following items should be described here as appropriate: responsibilities, constraints, composition, interactions, and resources. Use appropriate diagrams or other notation to describe your design (Not much explanation is required here)*

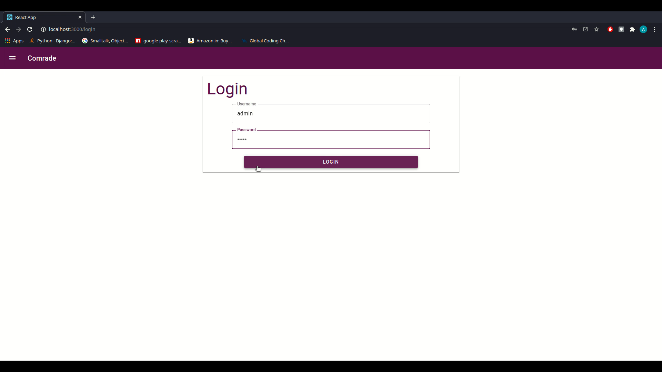
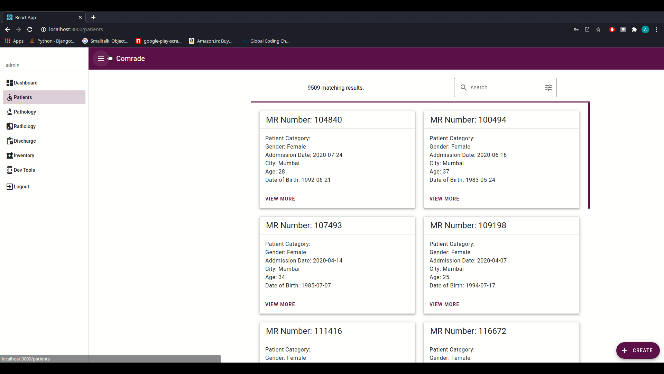
## USER INTERFACE DESIGN

The user interface design will consist of dashboard which will display all the statistical data. There will be patients and reports tab which will display their respective data along with CRUD functionalities.

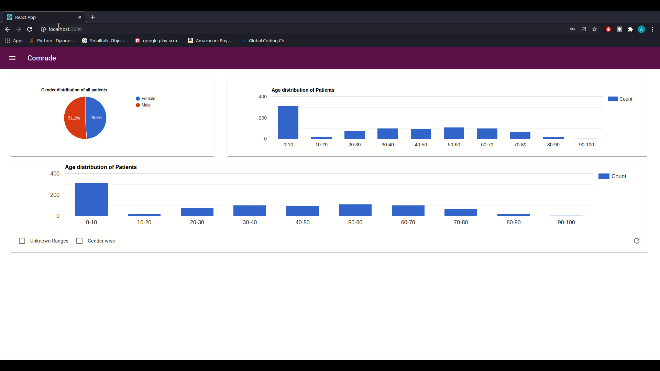
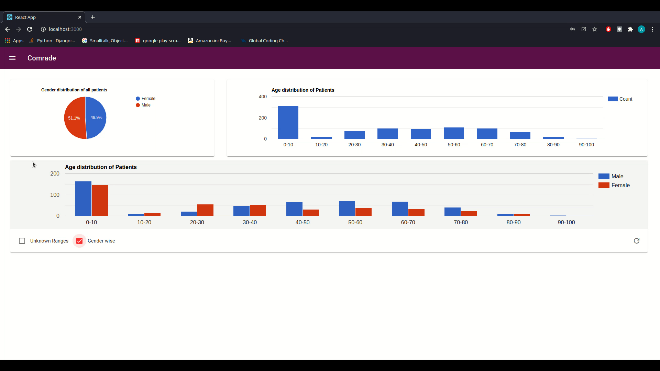
### Description of the User Interface

#### Screen Images

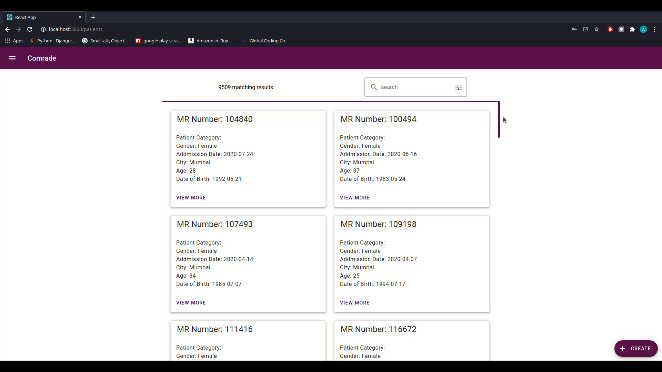
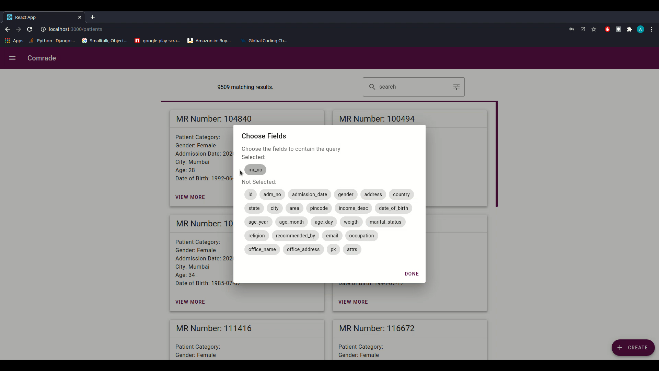
Login screen / Side Navigation

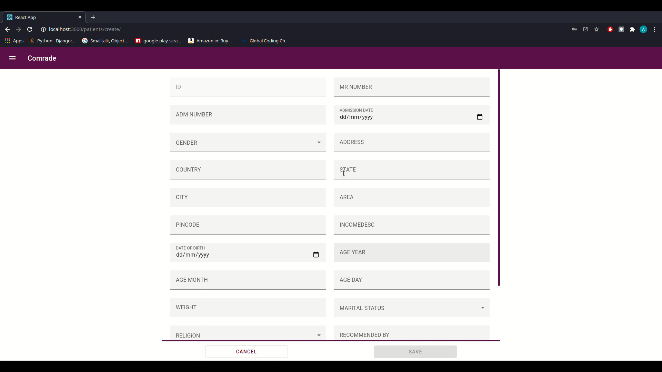
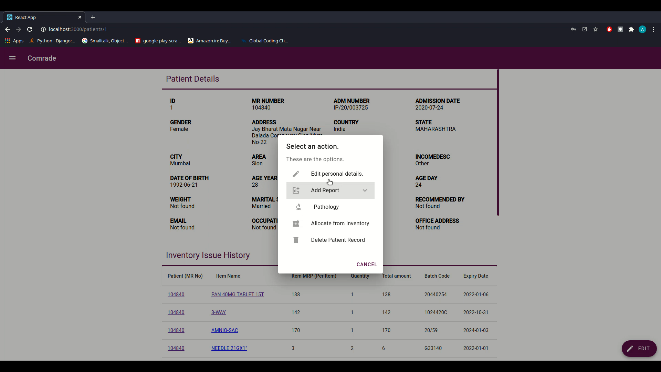
Dashboard

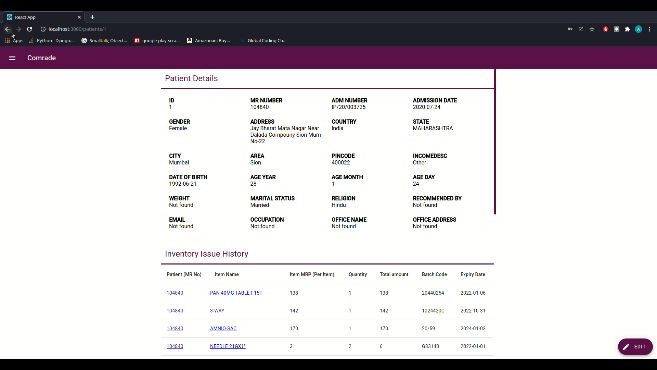
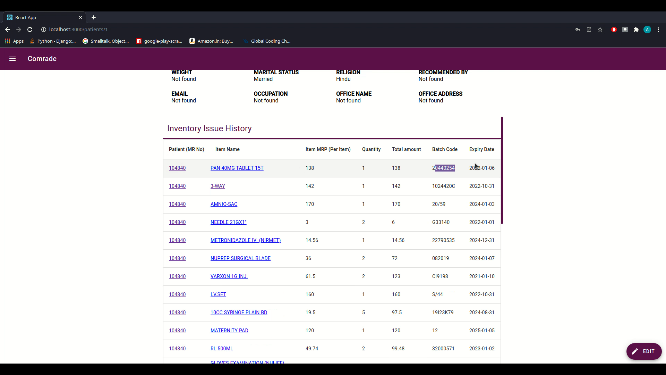
Patient Page

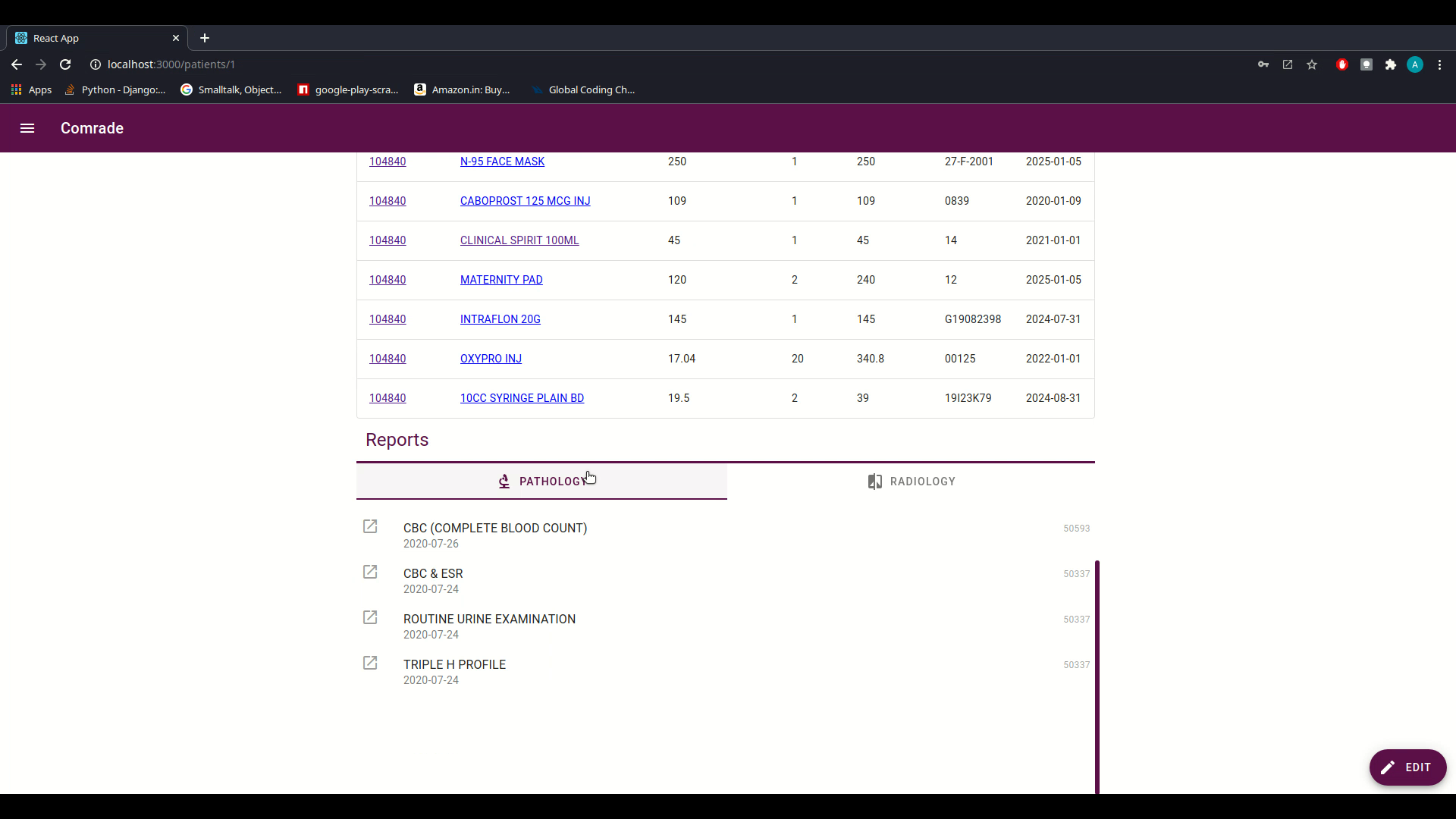
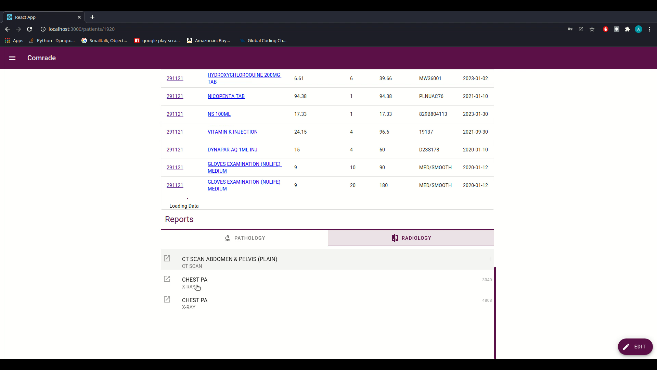
 

Add/Edit Patient

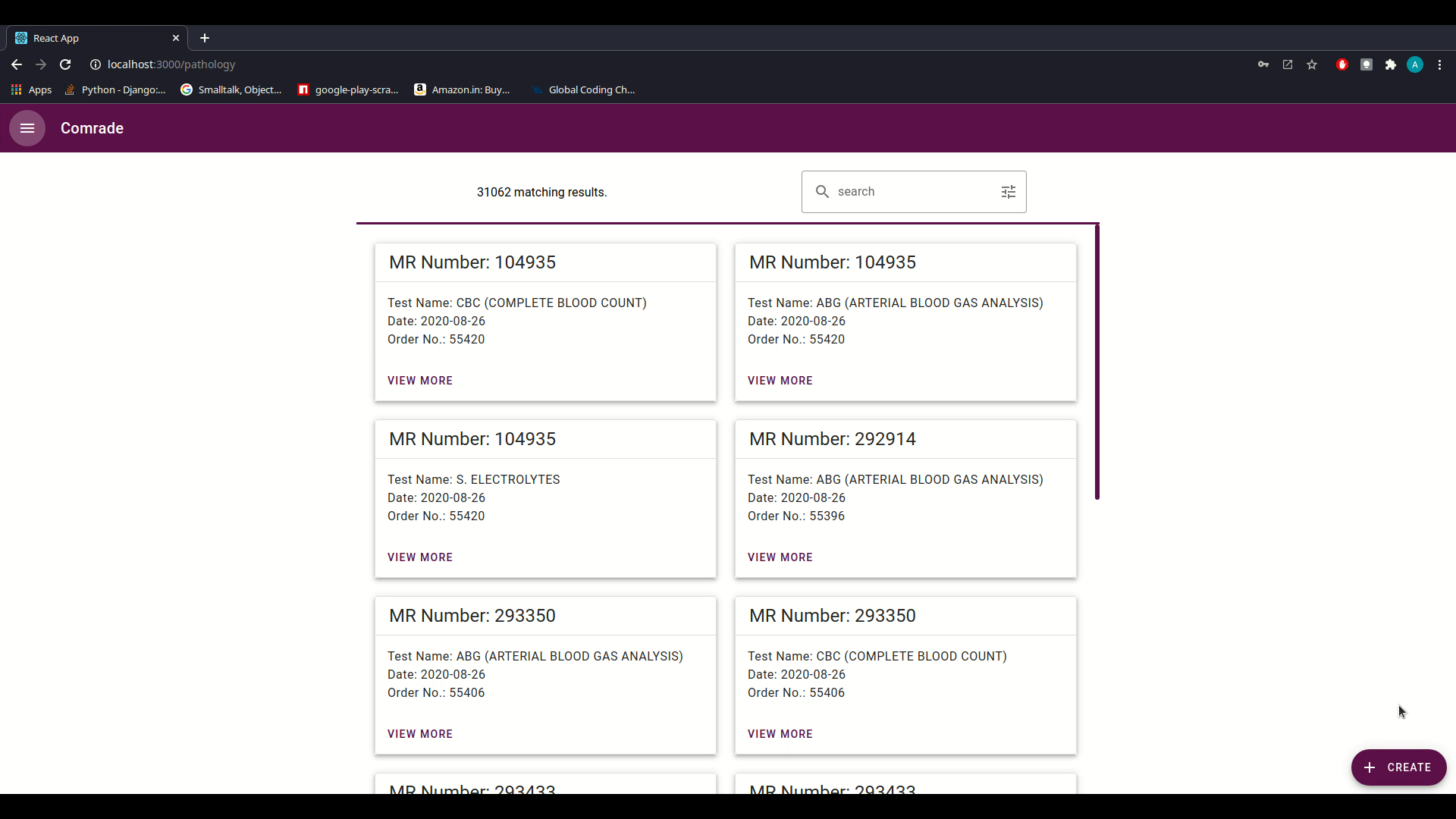
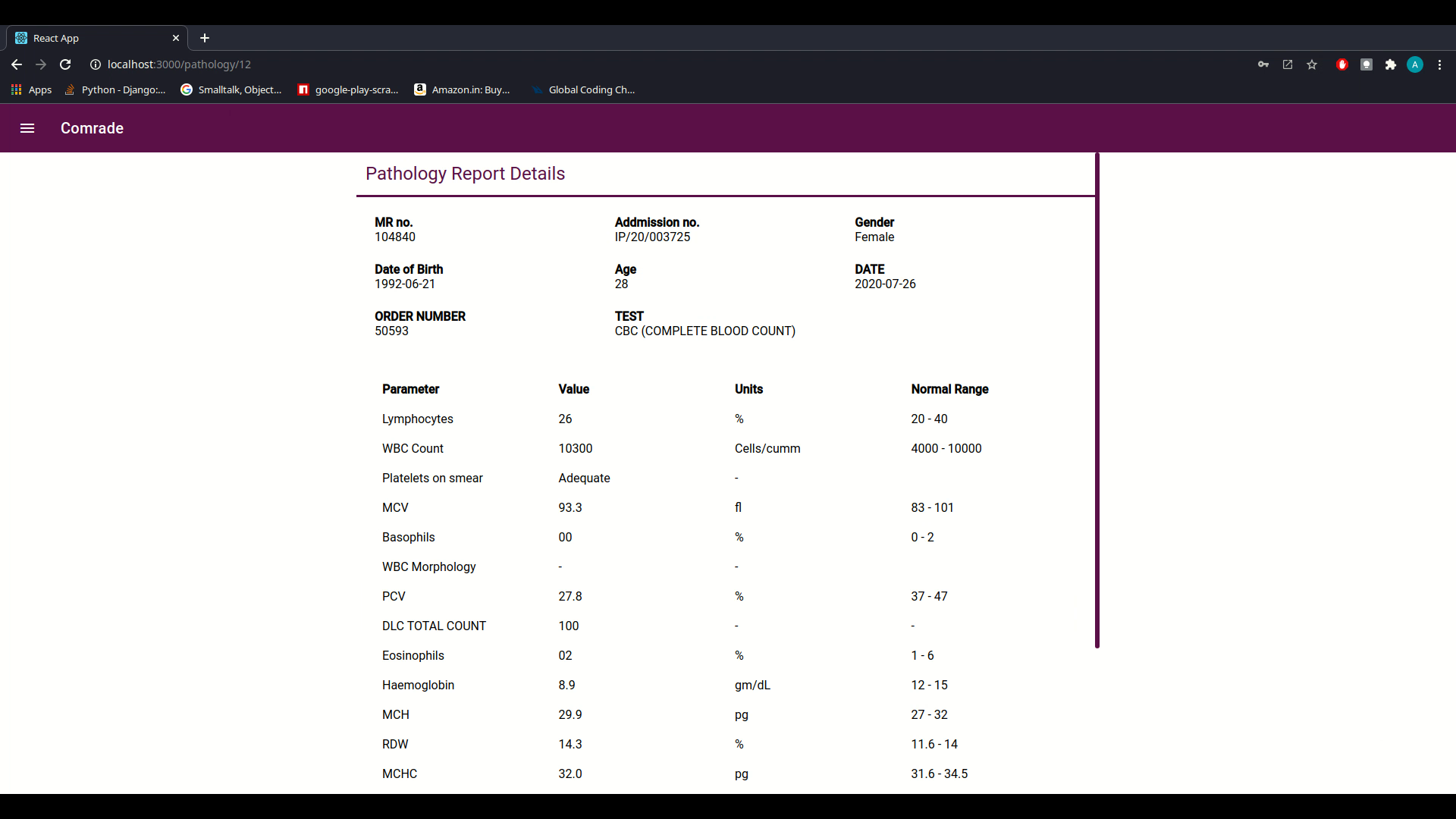
 

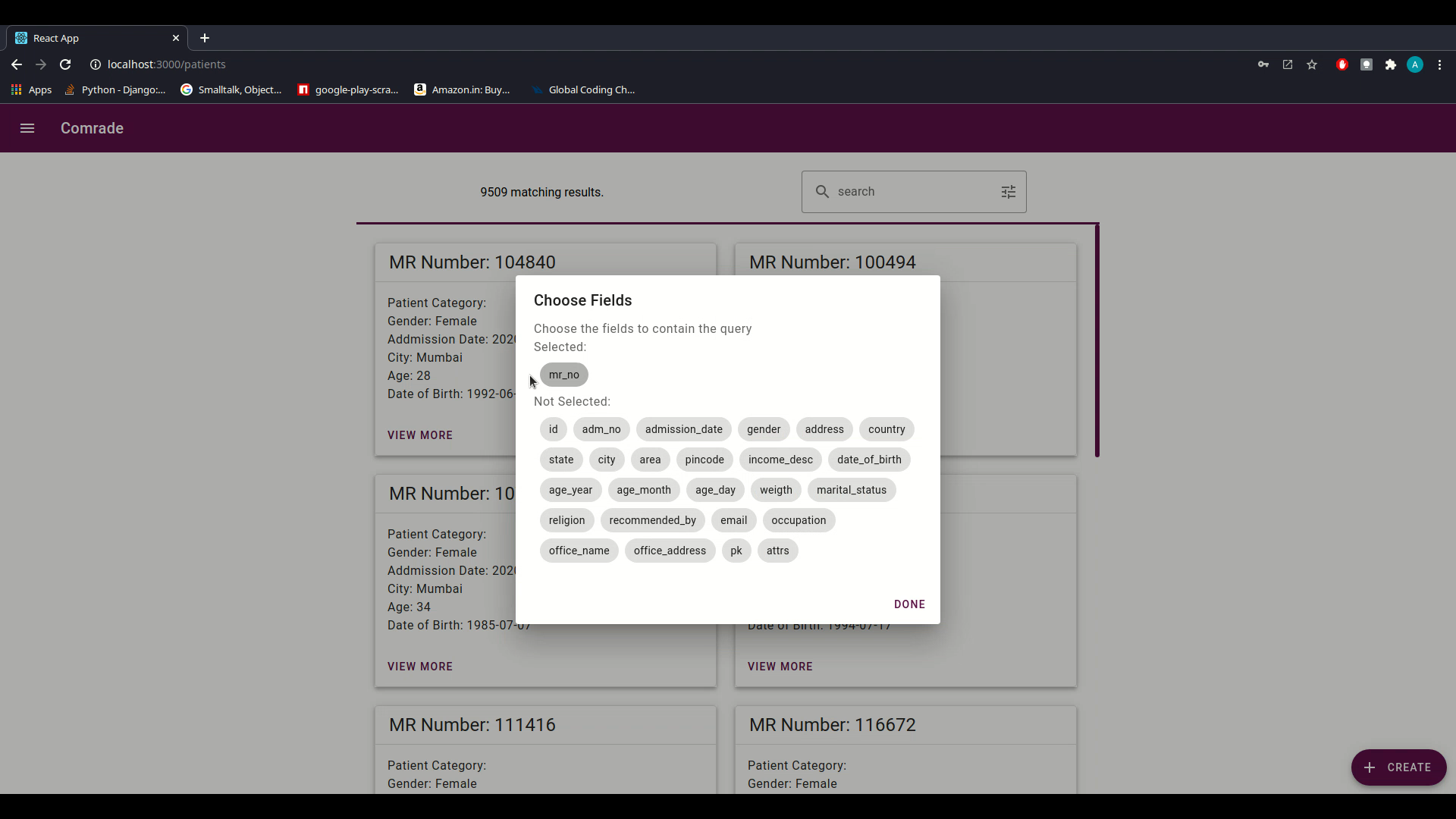
Patient Details

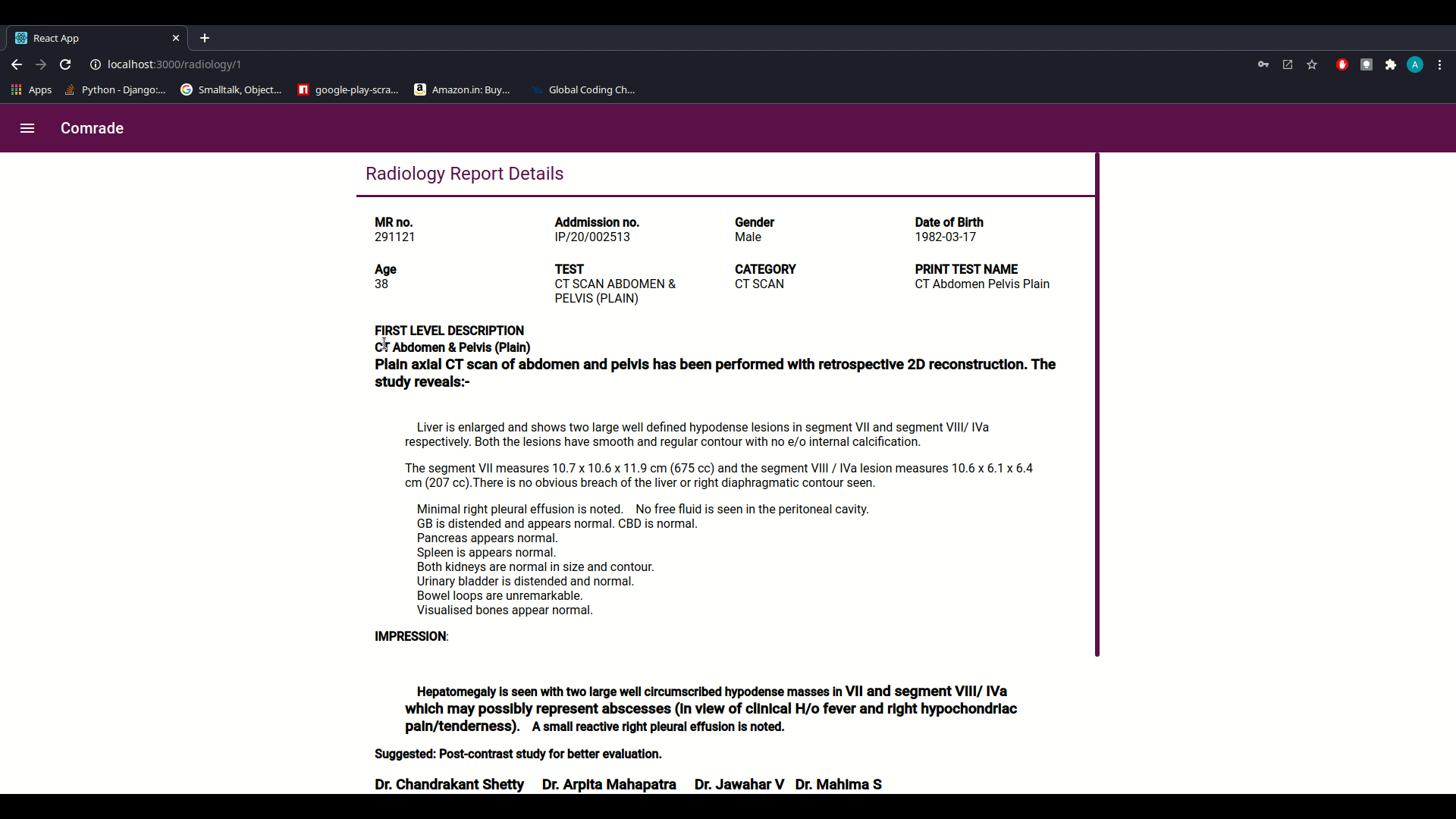
 

Pathology reports

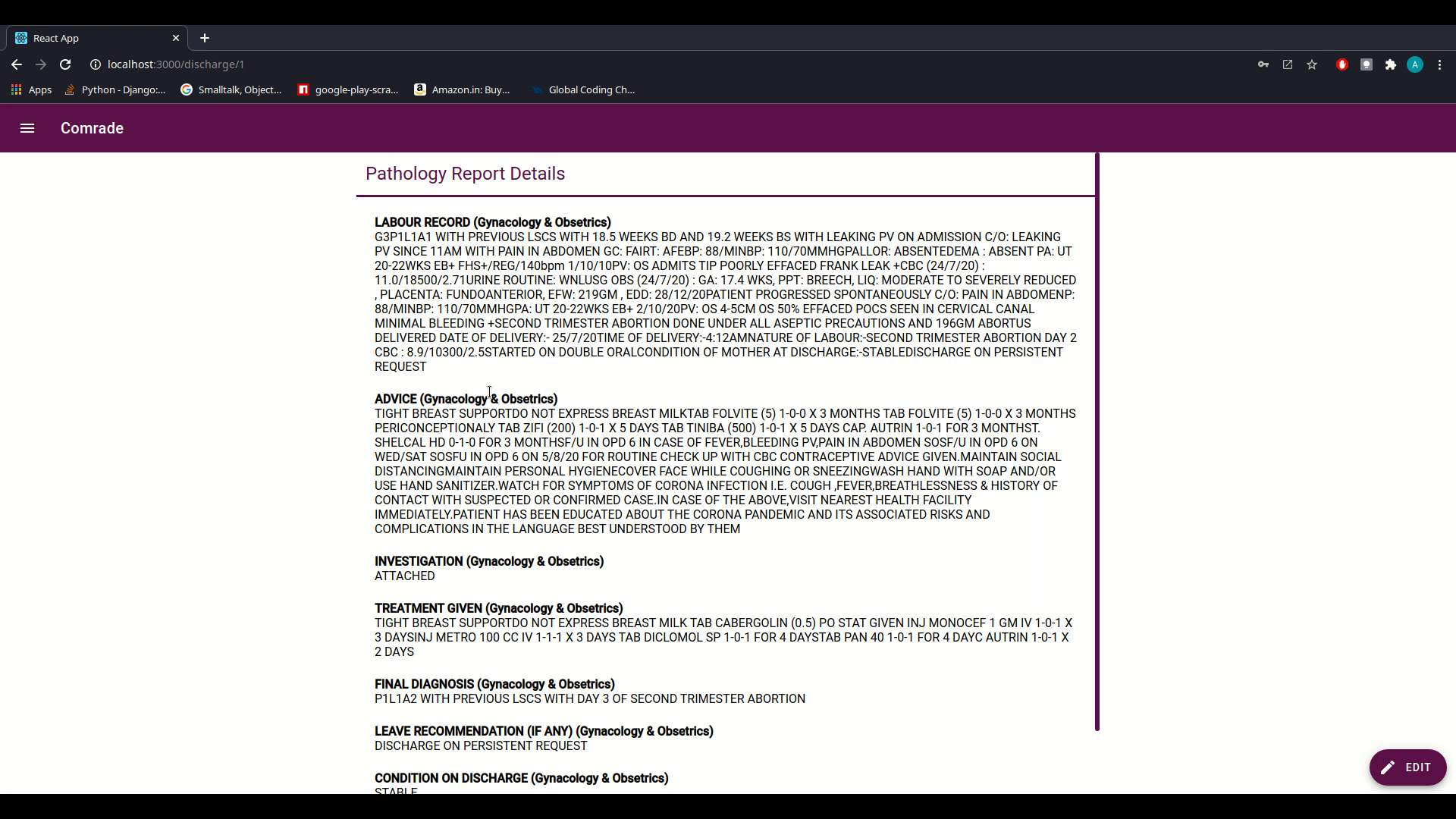
 



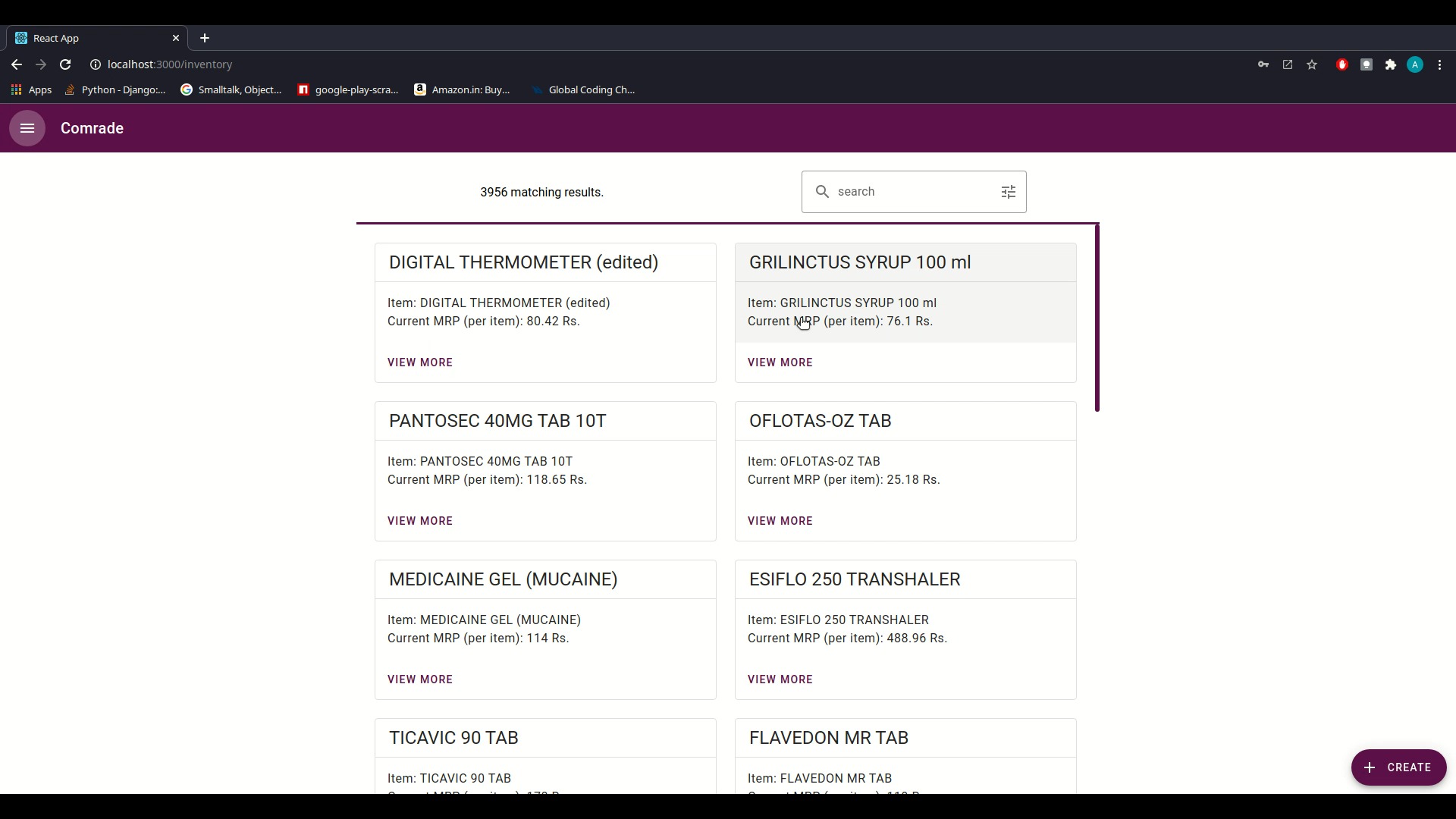
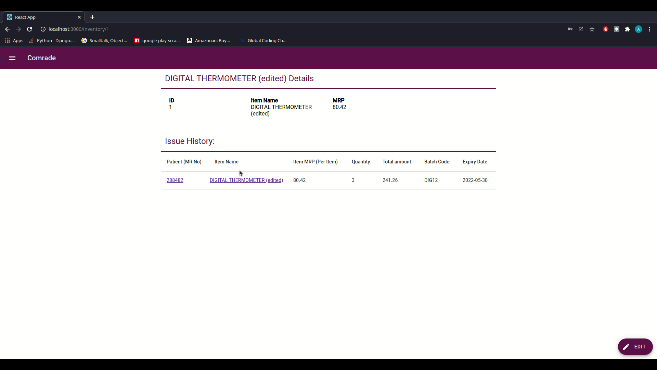
Radiology reports



Discharge report



Inventory

# System Test Document (STD)

## INTRODUCTION

### System Overview

COMRADE is mainly a hospital and patient management system. Therefore, it contains sensitive information making security of utmost importance. This is why security of the system is to be tested. As the application will be used by the hospital staff, it should only ease their work in these stress situations, therefore the performance of the application should always be top notch. Hence lighthouse testing is to be performed. Also, usability testing is performed to ensure that the application is easy to look at and understand even by any non-tech staff member.

### Test Approach

As every aspect of the application is to be tested, we first decide on the types of testing to be performed. When a certain type of testing is to be performed, test cases are generated for that type of testing. And the generated test cases are performed on the application and based on the test report the web application will be further improved.

## TEST PLAN

The test plan for COMRADE is to perform some specific software testing to test certain aspects of the software.

Kinds of testing to be performed include:

* + Security Testing
  + Lighthouse Testing
  + Usability Testing
  + Django Unit Testing
  + Integration testing
  + Continuous Integration Testing

### Features to be Tested

**Pre-deployment tests:**

* Django UNIT Testing:

Test all the CRUD operations i.e. Create, read, update and delete.

* Prediction Model to be tested for accuracy.

**Post-deployment tests:**

* Lighthouse Testing

Lighthouse testing is done to test the performance, usability and maintainability of the web application. As the application will be used by hospital staff, it can be expected that some of them might not be well versed with technology, hence usability should be maximum. Also the performance of the software should be good as the application was developed to help the hospital staff. Hence Lighthouse testing needs to be performed.

* Security

As the hospital data consists of sensitive data, Security is of utmost importance. Hence security is to be tested.

* Responsive

As COMRADE will be used by the hospital staff to maintain data, they may use different devices hence the web application must be tested for responsiveness.

* Latency

The latency test is done to test how much time is required for the web application to respond to the requests made to it. As this application is meant to ease the work load of the hospital staff, therefore the response time should be minimum. Hence Latency test is to be performed.

* Stress testing

Stress test is done to see how much network traffic can the web application is able to handle. As the application is expected to be available all the time, stress test is one of the important tests to perform.

### Testing Tools and Environment

* Travis CI

It is a software testing tool used for integration testing mainly.

* Django Unit Testing

It is part of the Django web framework which is used to perform unit testing.

* Lighthouse Testing

Lighthouse testing can be performed by using any web browser inspector.

* Security Testing

Sucuri software is used to perform security testing.

* Stress Testing

Junit for stress testing after deployment

## TEST CASES

Lighthouse Testing

1. Purpose

To test performance, usability, maintainability of the web application.

1. Inputs

The website link/URL.

1. Expected Outputs & Pass/Fail Criteria

Performance: 95%

Usability: 98%

Maintainability: 86%

1. Test Procedure

Go to the website using a web browser (preferably Google Chrome)

Open the chrome inspector

Go to Lighthouse Tab

Click on Perform Lighthouse testing

Security Testing:

1. Purpose

To test the security of the web application

1. Inputs

The web application

1. Expected Outputs &Pass/Fail Criteria

No errors

1. Test Procedure

The web application is to be simply passed through the sucuri software.

Django UNIT Testing

1. Purpose

To test each unit module of the web application.

1. Inputs

The source code

1. Expected Outputs & Pass/Fail Criteria

Based upon each module the expected outputs are decided. There should be no traceback errors generated.

1. Test Procedure

Write a test case code for each module and run it on the python3 shell.

Travis CI

1. Purpose

To perform integration testing.

1. Inputs

The web application

1. Expected Outputs and Pass/Fail Criteria

No errors

1. Test Procedure

Link TravisCI to the Github repo and perform the integration testing from their website.

# Conclusions and Further Work

# REFERENCES