1. **Identifying software life cycle model (SDLC), Agile methodology used**

Hospital Management System follows INCREMENTAL MODEL because initially software requirements are reasonably well defined but the overall scope of development effort is a purely linear process. There may be other requirements of the user which will be known later. So, those requirements can the implemented and delivered in the following next increments. Our project is a short term project and staffing available is also low .

**b) SRS document preparation, User Stories created**

**Software Requirement Specification Hospital Management System**

**The Software is for the automation of Hospital Management.**  
Purpose• The Software is for the automation of Hospital Management.•

It maintains two levels of users:-

Administrator Level

User Level

• The Software includes:-Maintaining Patient details. Providing Prescription, Precautions and Diet advice.Providing and also test suggestions .

It also includes maintaining doctors info , departments ,assigning a doctor to patient etc

**Product functions**  
Maintaining patient’s injection entry records.

Maintaining patient’s prescription, medicine and diet advice details.

Maintaining backup of data as per user requirements (between mentioned dates)

Results of tests, prescription, precautions and diet advice will be automatically updated in the database ,related test reports, patient details report, prescription can be generated as per user requirements.

User or Administrator can search a patient’s record by his/her ID .

**Goals of proposed system**  
1. Planned approach towards working: - The working in the organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.

2. Accuracy: - The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.

3. Reliability: - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.

4. No Redundancy: - In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.

5. Immediate retrieval of information: - The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires.

6. Immediate storage of information: - In manual system there are many problems to store the largest amount of information.

7. Easy to Operate: - The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

**Project Requirements**

User Characteristics Every user should be:

• Comfortable of working with computer.

• He must have knowledge in medical field

. • He must also have basic knowledge of English too.

Constraints

• GUI is only in English.

• Login and password is used for identification of user and there is no facility for guest.

**Definitions of problems**  
Problems with conventional system

1. Lack of immediate retrievals: -The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient’s history, the user has to go through various registers. This results in inconvenience and wastage of time.

2. Lack of immediate information storage: - The information generated by various transactions takes time and efforts to be stored at right place.

3. Lack of prompt updating: - Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

4. Error prone manual calculation: - Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient’s bill based on various treatments.

5. Preparation of accurate and prompt reports: - This becomes a difficult task as information is difficult to collect from various registers.

**Front-end and back-end selection**

**Front-end selection:**

1. It must have a graphical user interface that assists employees that are not from IT background.

2. Scalability and extensibility.

3. Flexibility.

4. Robustness.

5. According to the organization requirement and the culture.

6. Must provide excellent reporting features with good printing support.

7. Platform independent.

8. Easy to debug and maintain.

9. Event driven programming facility.

**Back-end Selection:**   
Back-end Selection:

1. Multiple user support.

2. Efficient data handling.

3. Provide inherent features for security.

4. Efficient data retrieval and maintenance.

5. Stored procedures.

6. Popularity.

7. Operating System compatible.

8. Easy to install.

9. Various drivers must be available.

10. Easy to implant with the Front-end.

**Technologies to be used**  
Python

Streamlit api for User Interface

SQLite for database storage

**System Specifications**

H/W Requirement -

Core i5 processor

2GB Ram.

20GB of hard disk space in terminal machines

1TB hard disk space in Server Machine

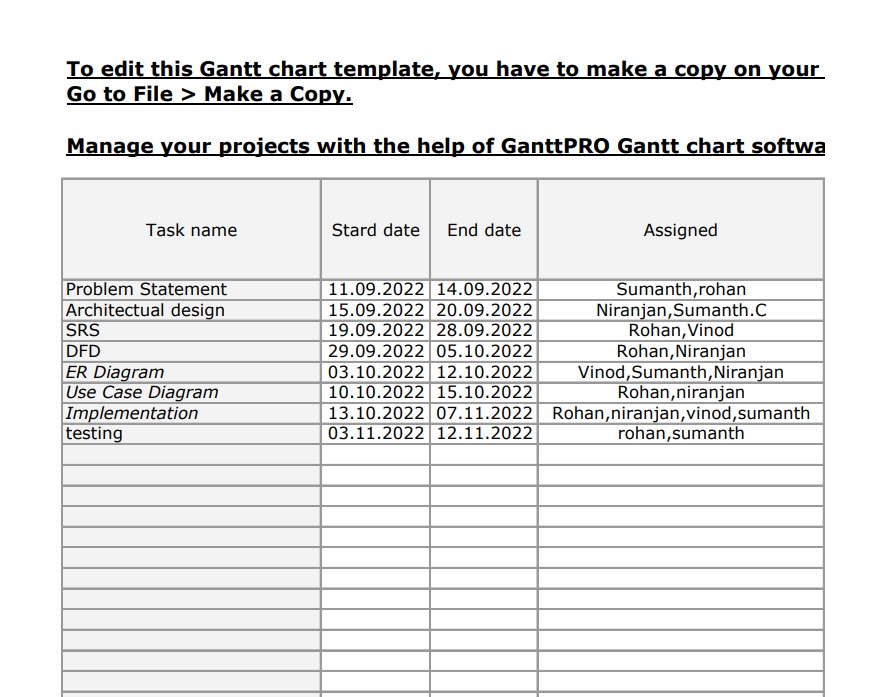
S/W Requirement -

Windows 7 or above operating system

streamlit

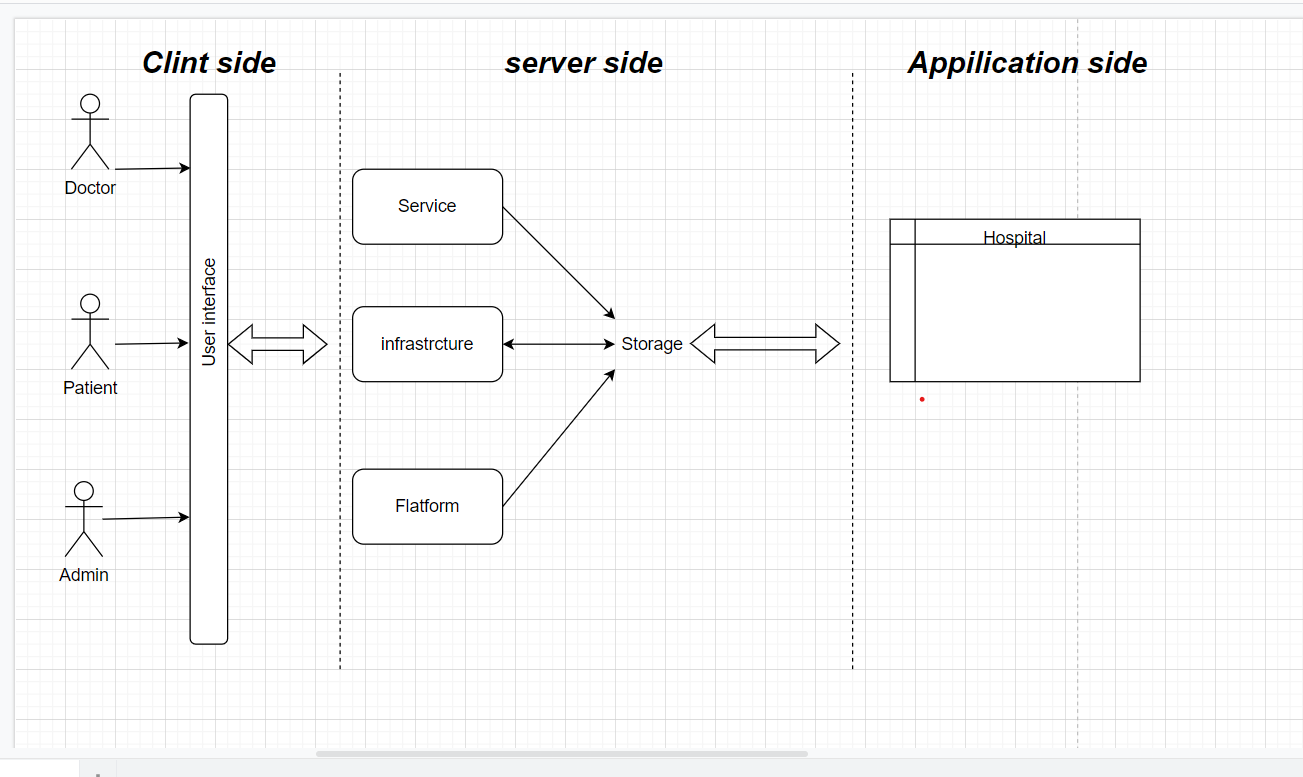
Mysql server

**d. WBS using GanttPRO**

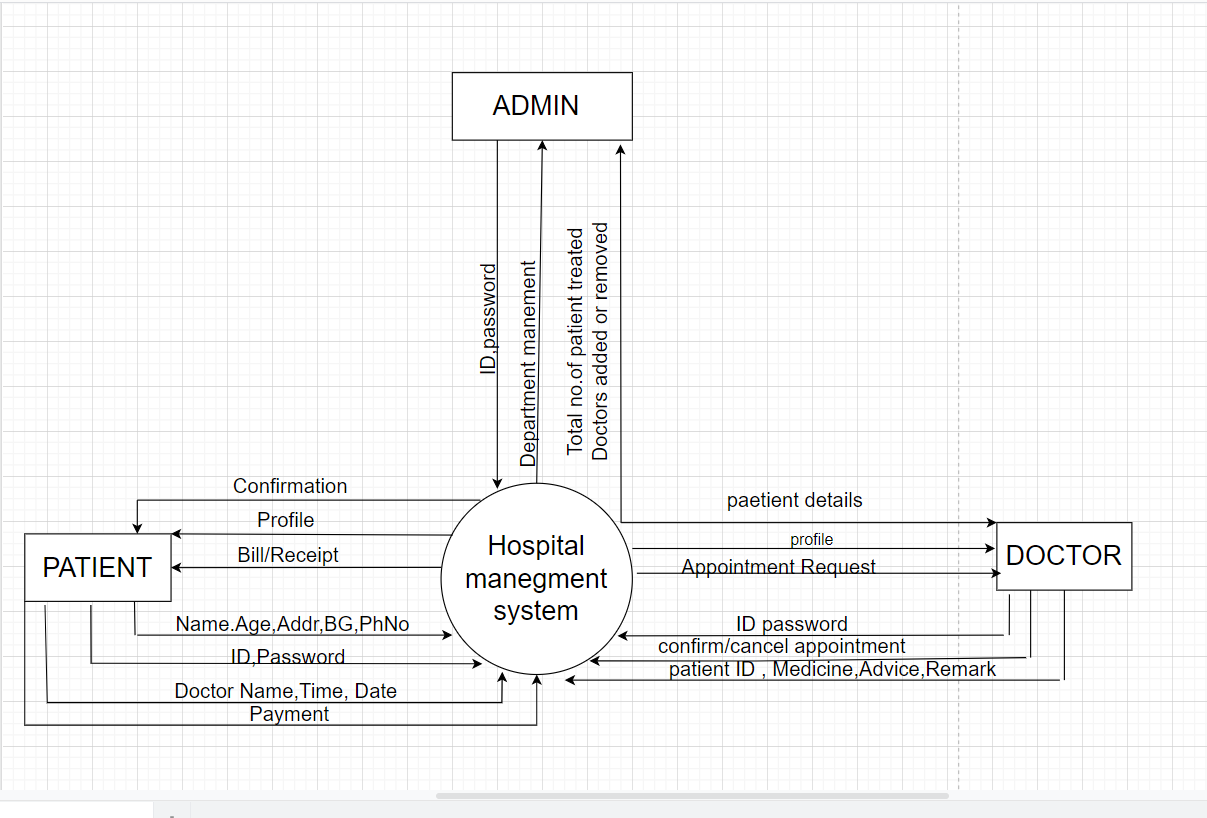


**e.Design for the project selected – Show Architectural diagram, Design Diagram, UML diagrams, DFD, etc**

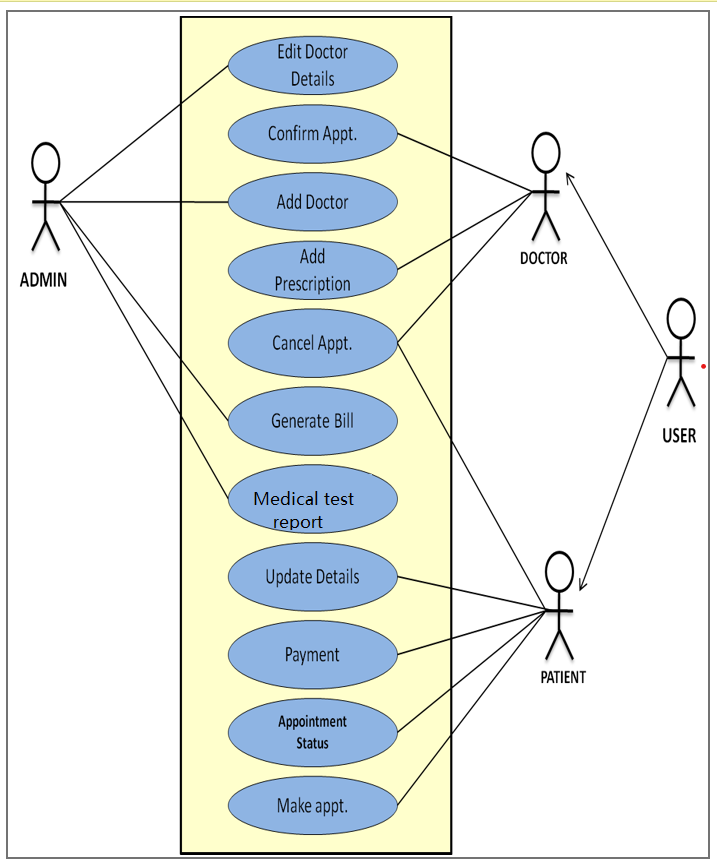
**Architectural diagram**

****

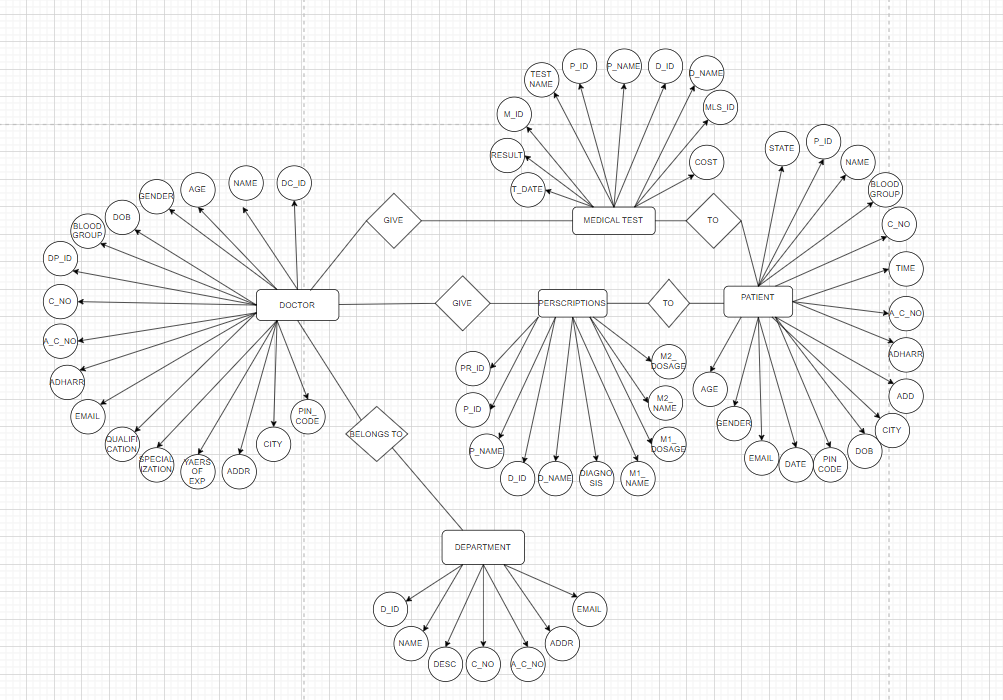
**DFD 0**

****

**Use Case**

****

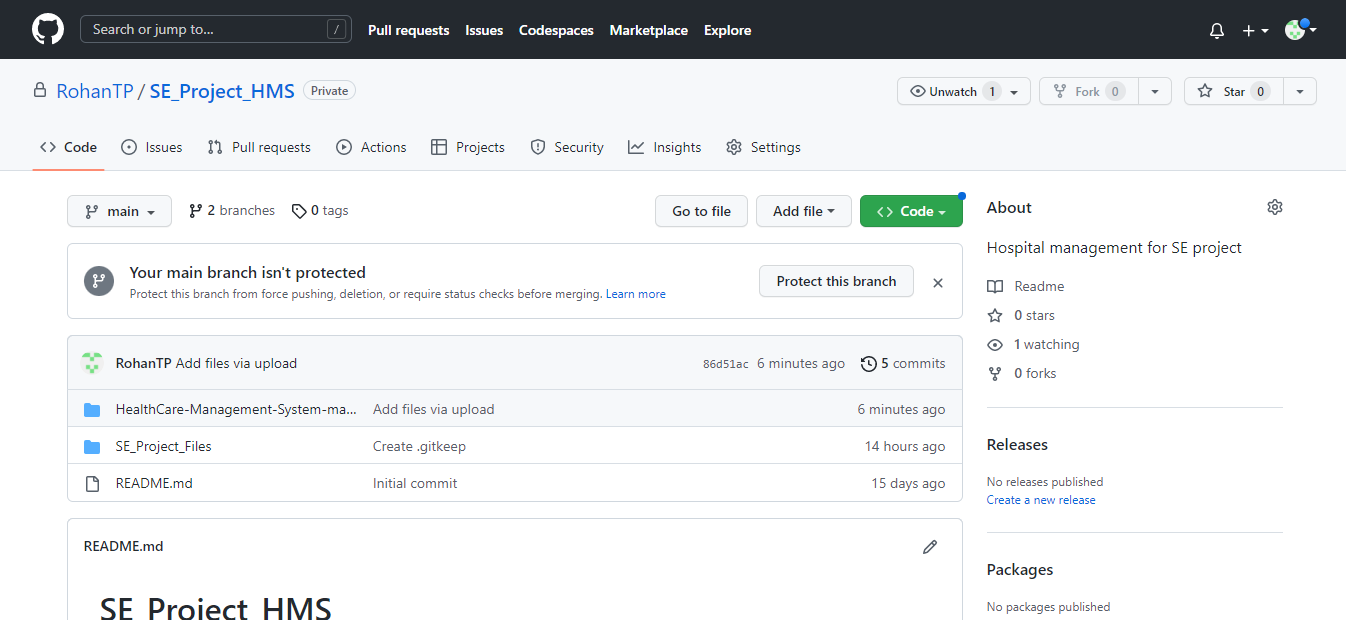
**ER diagram**

****

**f. Explain Coding practices/standards used**

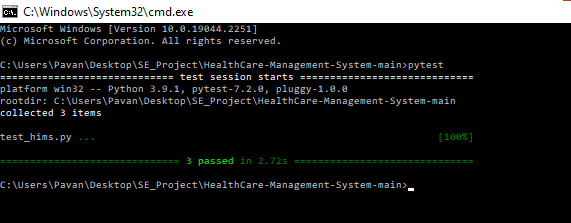
* **Variable naming conventions** - Using names that are self -descriptive
* **Class and function naming conventions** - Unique Identifers.
* **Clear and concise comments.**
* **Indentations**
* **No deep nesting**
* **daily backups into an instinct**
* **Exception Handling - Used with blocks for error handling**
* **Portability** - Every functionality is broken down into modules so that it can imported and used with minimal changes
* **Reusability and scalability** - Make improvements and add modules to make the project bigger without need to alter existing modules
* **Testing**.

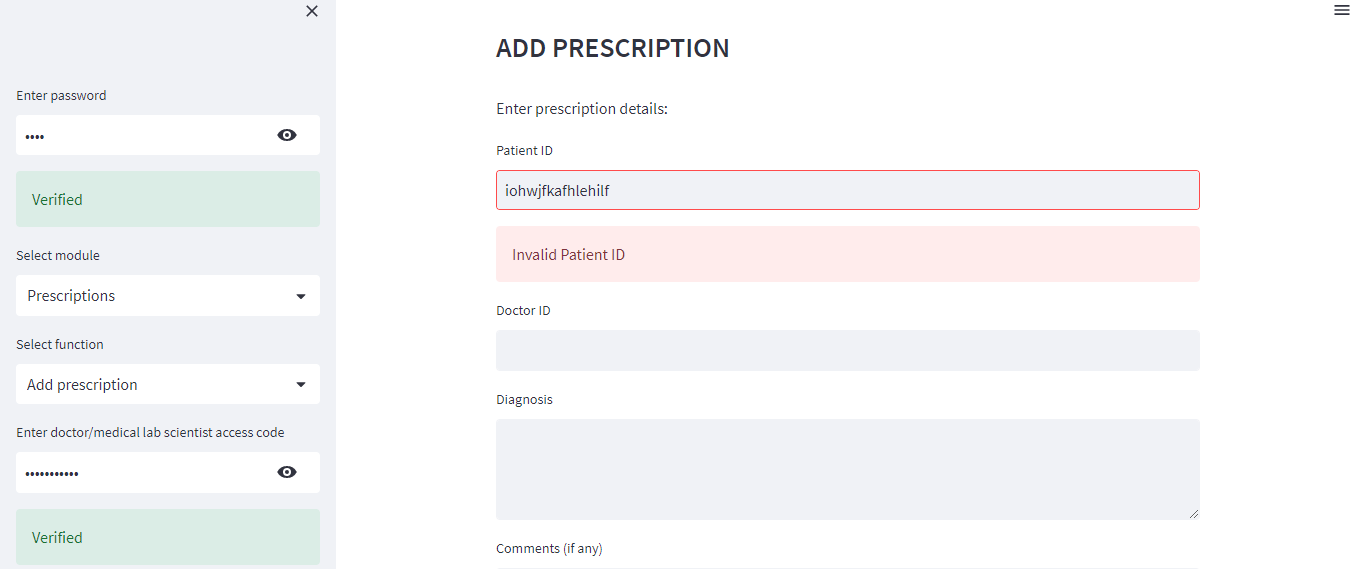
**g. SCM environment used like GitHub and any SCM concepts such as Branch Management and Versioning used for the project**

****

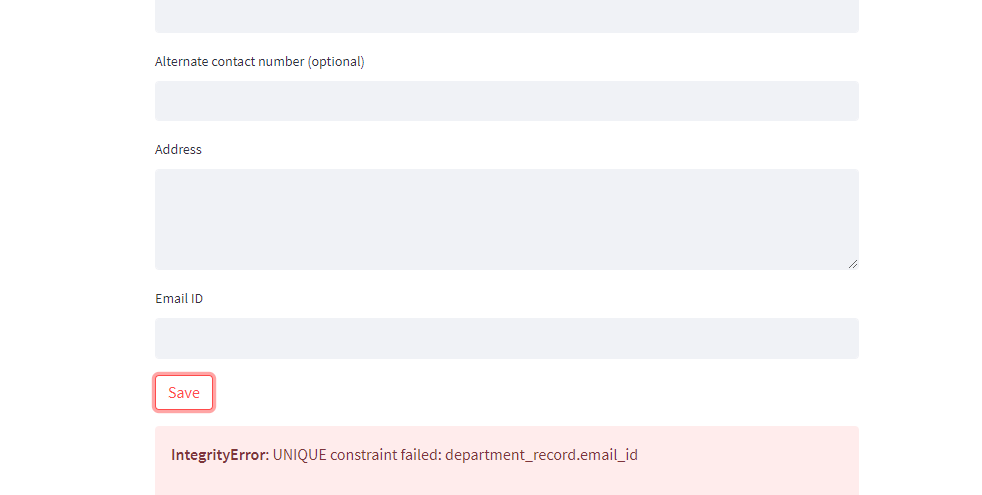
**h. Test strategy, test plan, test suite, test cases created**

**Unit testing**

****

****

Adding duplicate email id(that exists for some the separtment) or NULL values



**System Testing :**

Done manually by running the application in the environment similar to customer environment

the following streamlit bug was discovered

which was present in their latest version

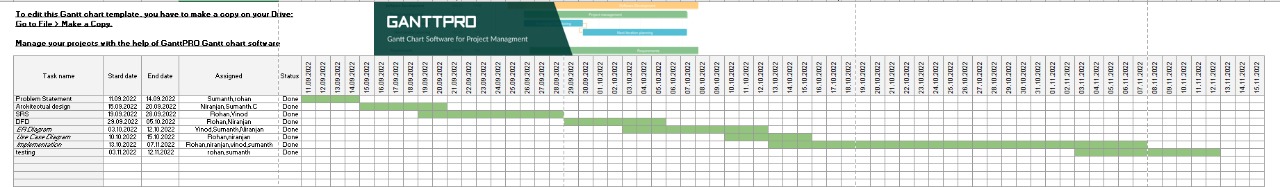
**Solution** : We resolved it by changing few configurations

in config.toml file of the streamlit app

by running it using the command

**streamlit run hims\_app.py --global.dataFrameSerialization = "legacy"**

**j. GanttPRO report showing all the tasks created, tracked, updated, monitored and closed**

****