CS 432: Databases

Assignment-3: Implementing a Web App using MySQL

Group Members

Aishwarya Omar	20110008
Akshat Shrivastava	20110009
Amaan Ansari	20110011
Mahesh Dange	20110050
Dhakad Bhagat Singh	20110055
Govardhan Ingale	20110070
Ronak Hingonia	20110170
Ojas Washimkar	20110234
Yash Adhiya	20110235
Yash Kokane	20110237
Shubham Kumar	19110137

Project Description

The given problem statement required team to develop a web application using Flask and MySQL to manage a dispensary. The team is divided into two groups, G1 and G2, with G1 responsible for the front-end and G2 for the back-end. The application should support the dynamic execution of database functions, such as insert, rename, update, delete, (CRUD) and where clause. Additionally, the application should have a login page with user and stakeholder authentication.

To develop this application, the team was first needed to design and implement the database schema. We identified the tables and fields required to store information related to the dispensary, such as products, suppliers, employees, customers, and transactions.

Once the database schema is in place, G2 began working on the back-end development. We have used Flask to create the API endpoints that interact with the database. These endpoints should implement the required functions, such as inserting new records, updating existing records, deleting records, and executing where clause queries.

Meanwhile, G1 worked on the front-end development. They used Bootstrap, HTML, and CSS to create a user-friendly interface for the web application. The interface allows users to interact with the database by providing forms and buttons to add, update, and delete records. The interface displays the data stored in the database in a clear and organized manner.

As a part of the project, the team implemented user authentication to ensure that only authorized users can access the application. The login page should prompt users to enter their credentials, and upon successful authentication, the user should be redirected to the main page of the application.

Finally, the team tested the application to ensure that all functions are working correctly, and the changes made through the application are correctly reflected in the main database and web page. Once the application is ready, it was pushed to GitHub.

Steps to run this project:

Installation requirements:

- python3 -m venv env_flask
- pip3 install flask
- pip3 install flask-mysqldb
- pip3 install pyyaml

Modules used:

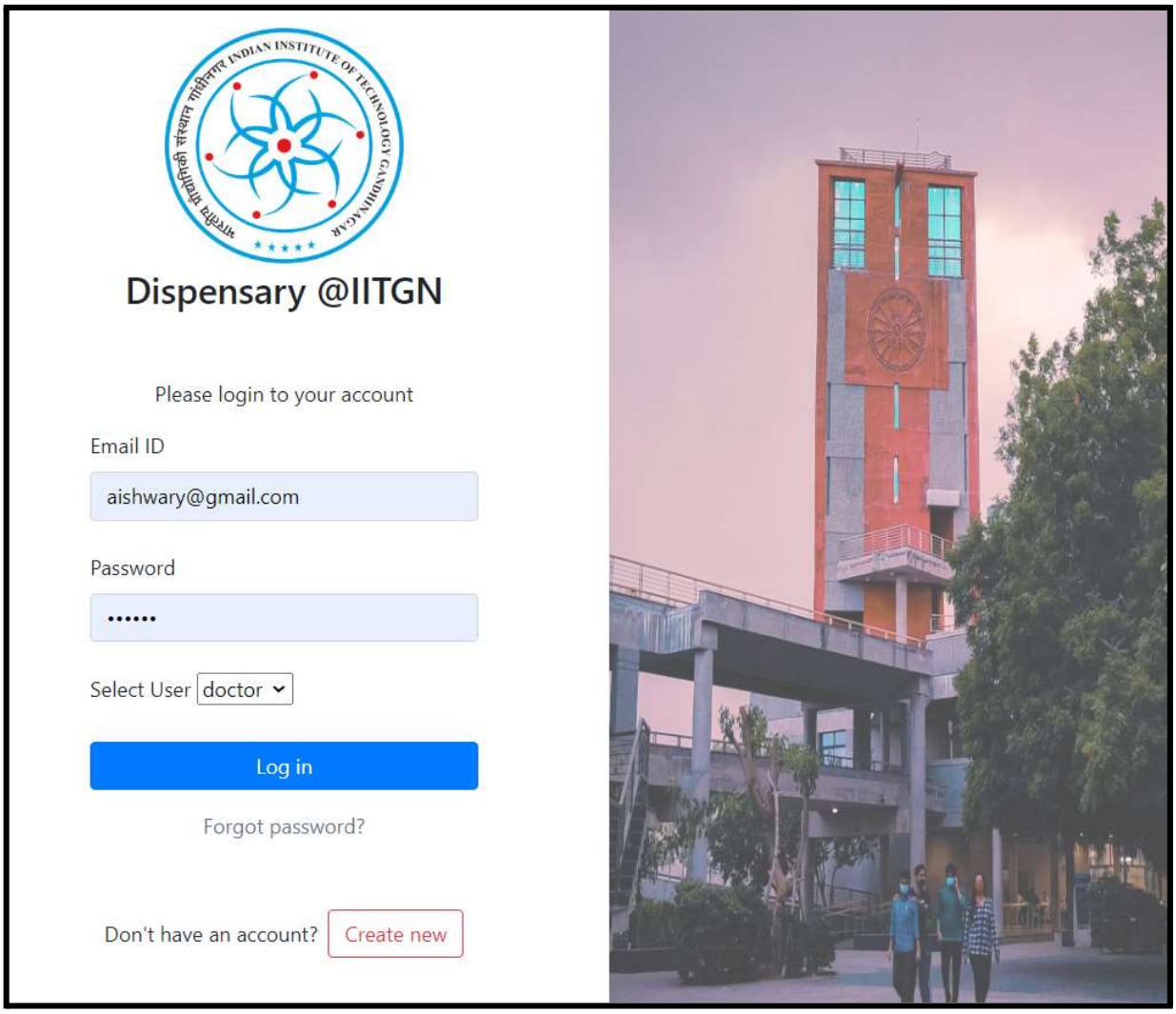
- from flask import Flask, render_template, request, redirect, url_for, session
- from flask_mysqldb import MySQL
- from sql_tools import *
- from html_tools import *
- import flask
- import yaml
- import MySQLdb.cursors
- import re

> Note: MySQL Server(workbench) should be installed on the PC. Dump the "dispensary.sql" in the same directory.

These are the major steps to launch the website:

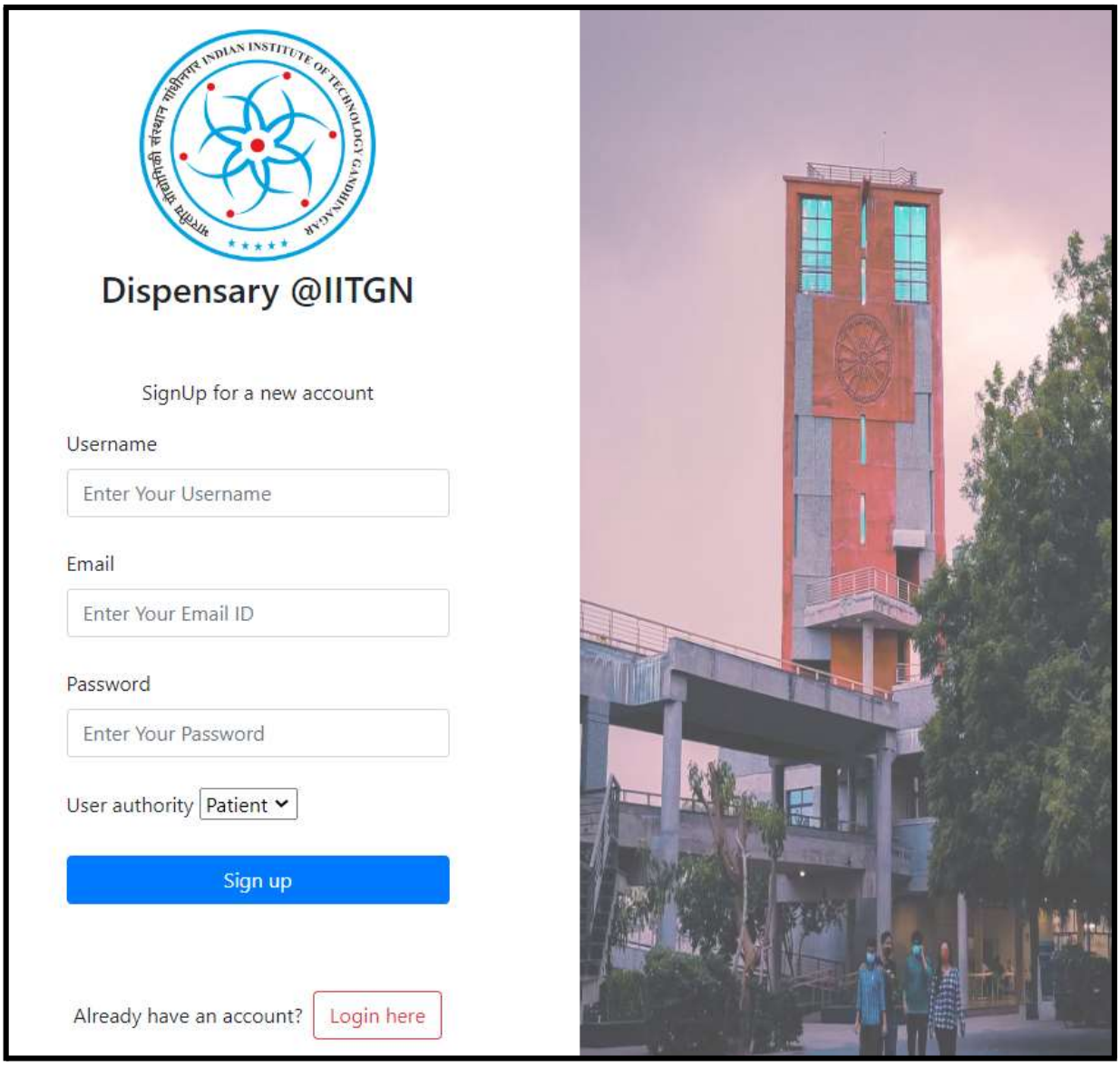
- Move all the files in one folder. Open folder as code in VS Code.
- Create virtual environment using command "*python -m venv venv*"
- Activate environment using command "*./venv/Scripts\activate.ps1*"
- Install required libraries (mentioned above)
- Change the database configuration i.e., password, file, server host and user.
- Run the "app.py" file using command "*python ./app.py*"

Now after running the server you will get a login page:



For the existing users(login details should be available in the database), they can login to their account. Insert email and password and **select** the role correctly. You will be redirected to the home page.

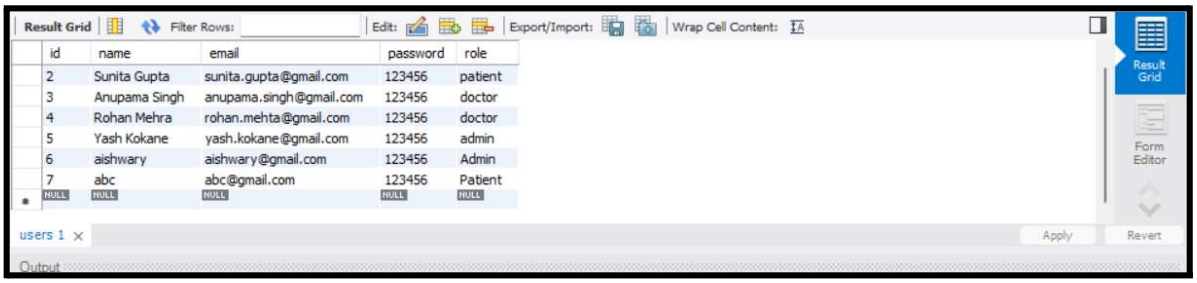
If you are new user, you can click on “create new” on the bottom of the login page, you will be redirected to registration page:



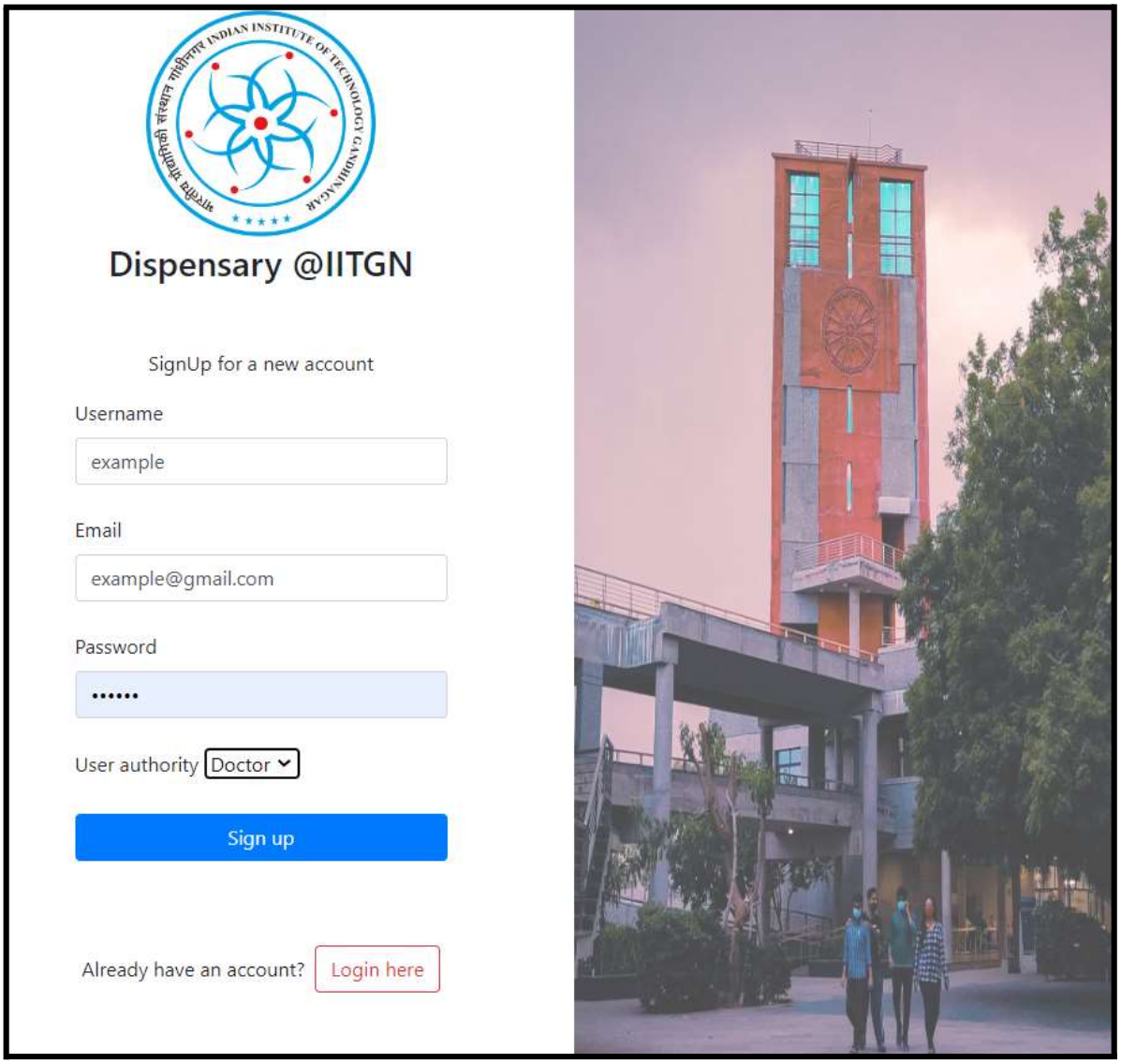
Here, user need to insert username, a not existing email, password and role. You credentials will be saved in the database and now you can click on login. Login with your credentials.

For example, adding these credentials:

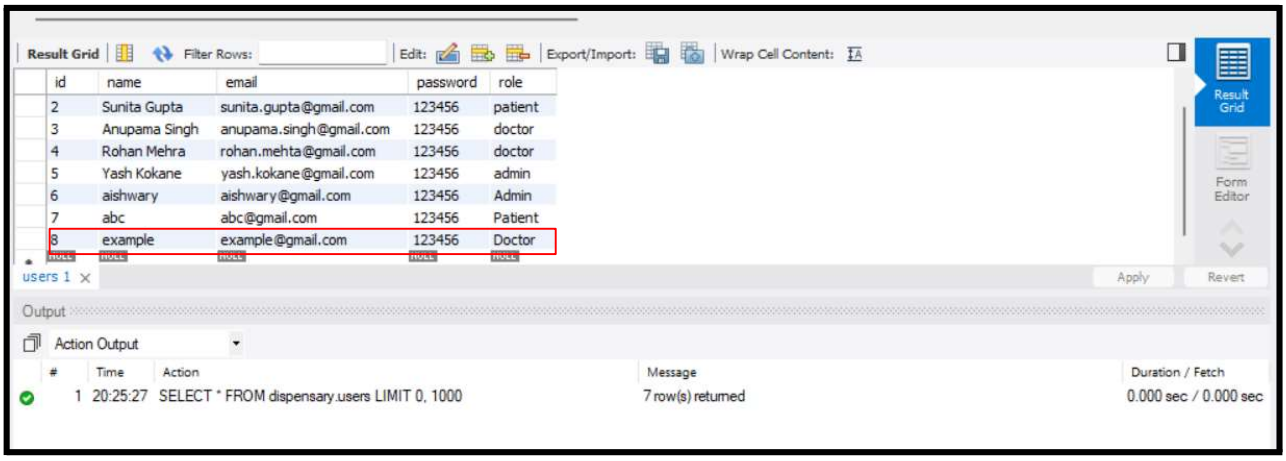
Database previously:



id	name	email	password	role
2	Sunita Gupta	sunita.gupta@gmail.com	123456	patient
3	Anupama Singh	anupama.singh@gmail.com	123456	doctor
4	Rohan Mehra	rohan.mehta@gmail.com	123456	doctor
5	Yash Kokane	yash.kokane@gmail.com	123456	admin
6	aishwary	aishwary@gmail.com	123456	Admin
7	abc	abc@gmail.com	123456	Patient

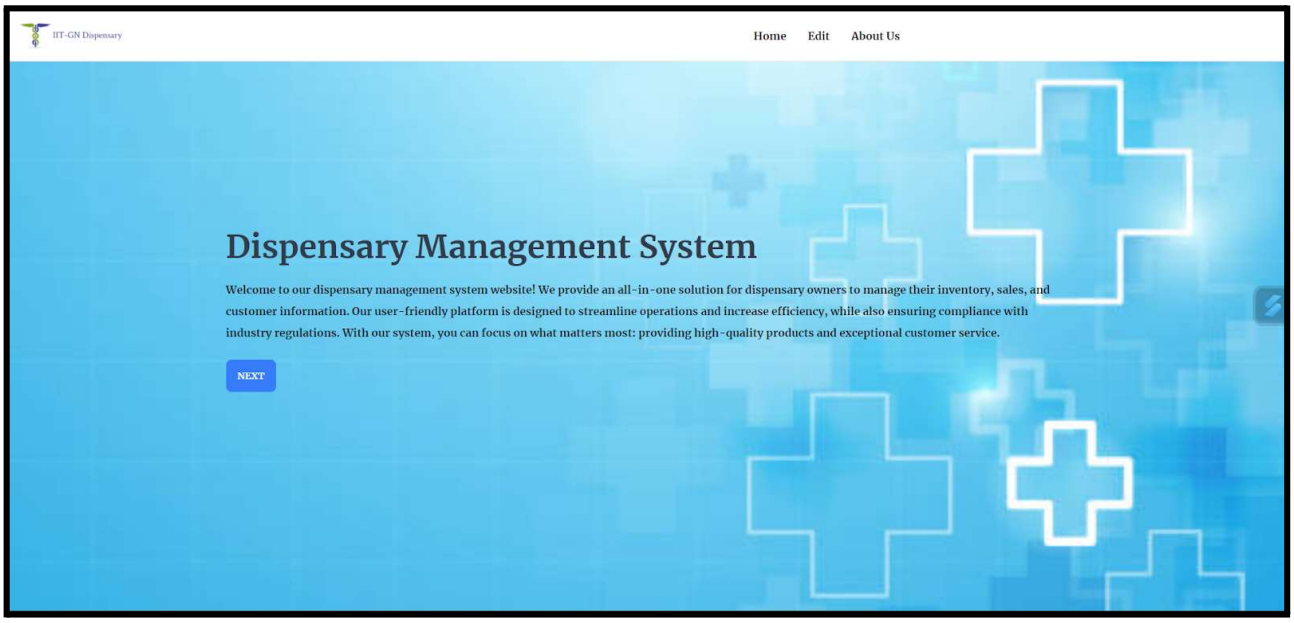


After hitting “Sign up” and refreshing database, new database snapshot, with updated entry:



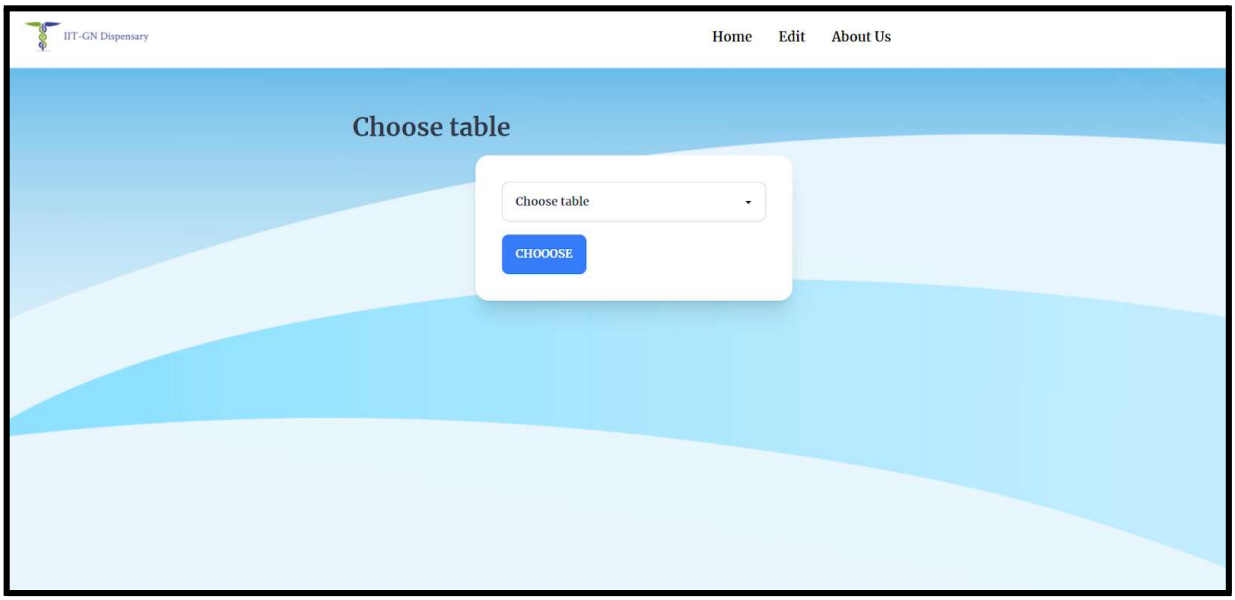
id	name	email	password	role
2	Sunita Gupta	sunita.gupta@gmail.com	123456	patient
3	Anupama Singh	anupama.singh@gmail.com	123456	doctor
4	Rohan Mehra	rohan.mehta@gmail.com	123456	doctor
5	Yash Kokane	yash.kokane@gmail.com	123456	admin
6	aishwary	aishwary@gmail.com	123456	Admin
7	abc	abc@gmail.com	123456	Patient
8	example	example@gmail.com	123456	Doctor

After you are logged in, you will be able to see **home page:**

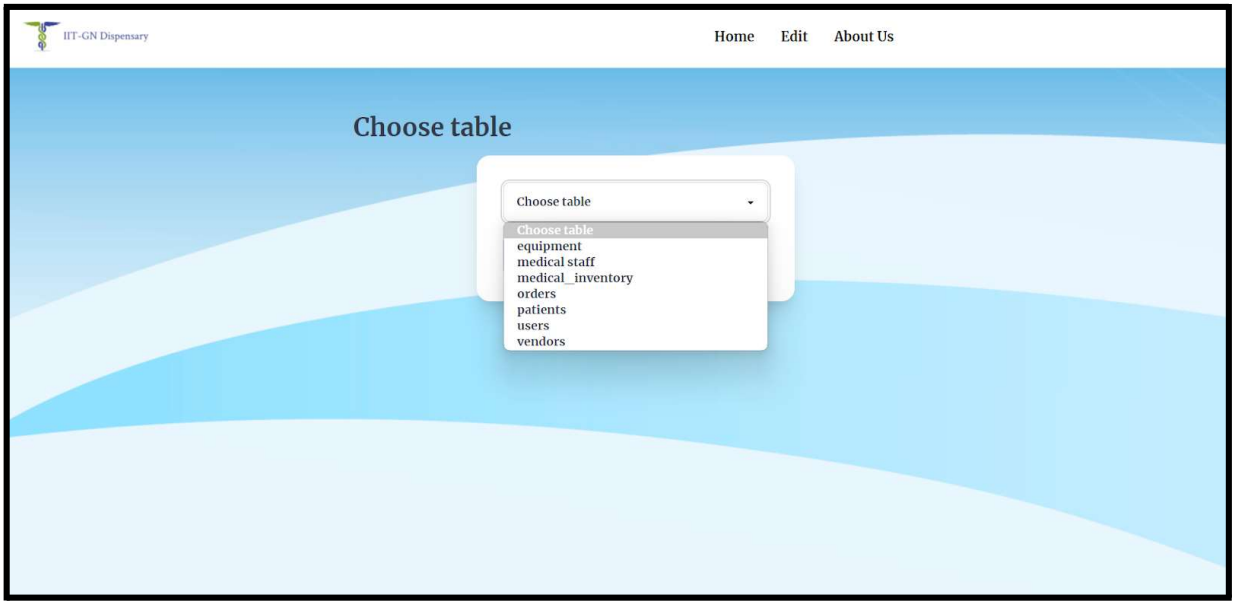


From here you can explore options. Clicking on “**IIT-GN Dispensary logo**” will refresh the page, “**Home**” will return to the same page. “**Edit**” allows to to update/delete the data from the tables. And “**About_us**” contains basic information about our project, teammates and motivation.

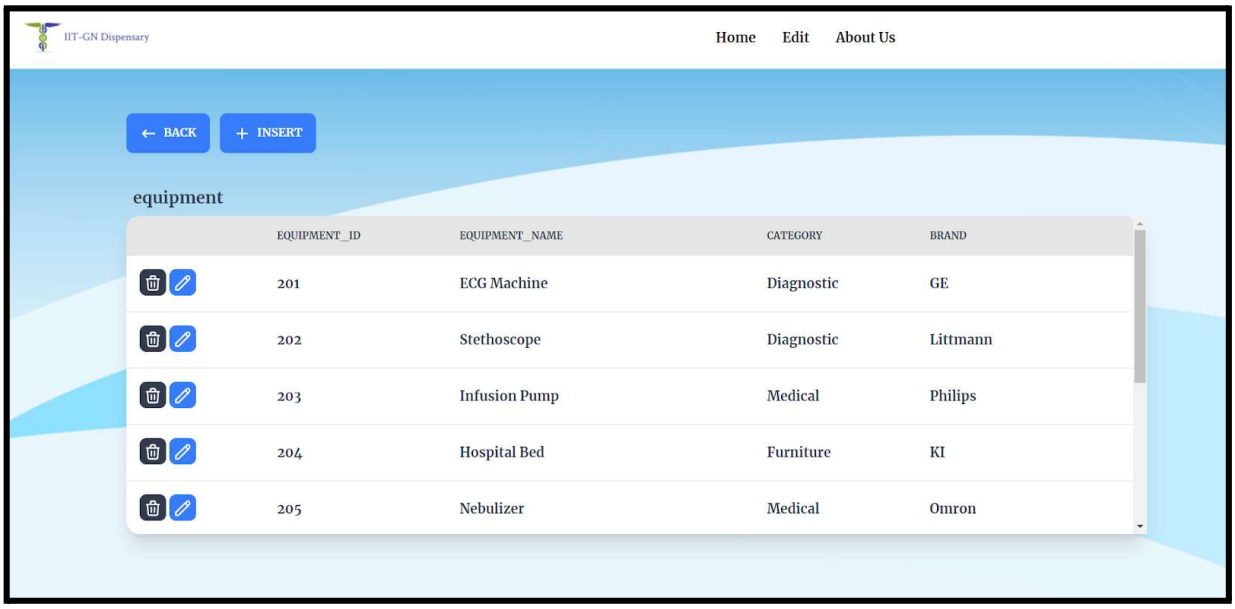
“Edit” View1:



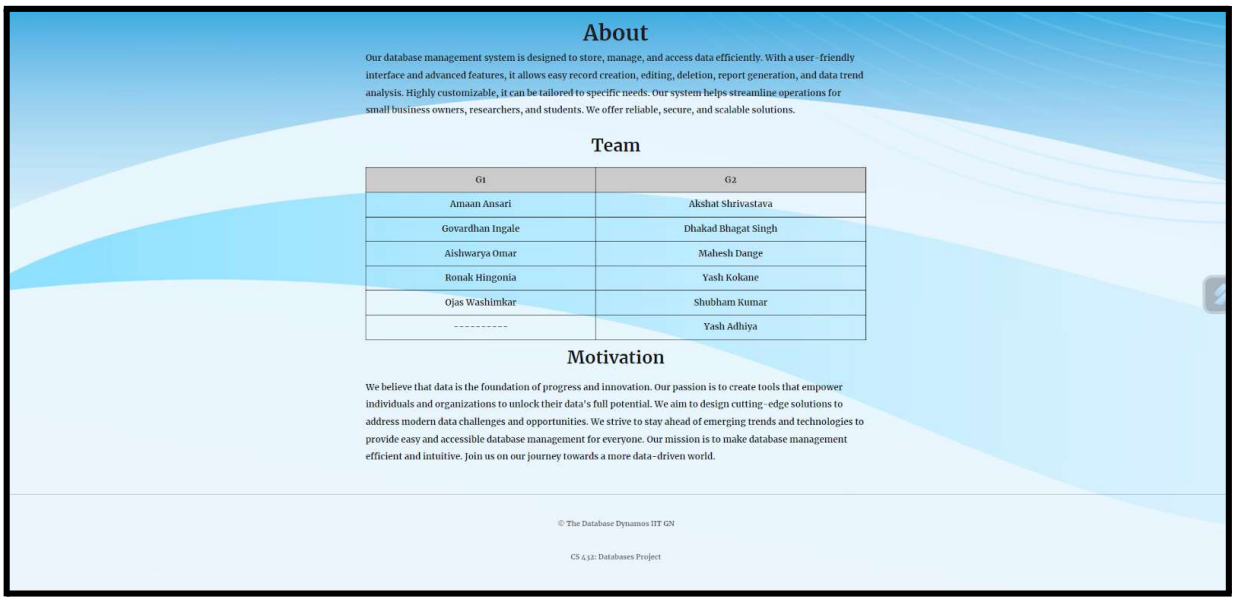
“Edit” View2:



Editing particular table(here “equipment”):



“About us” view:



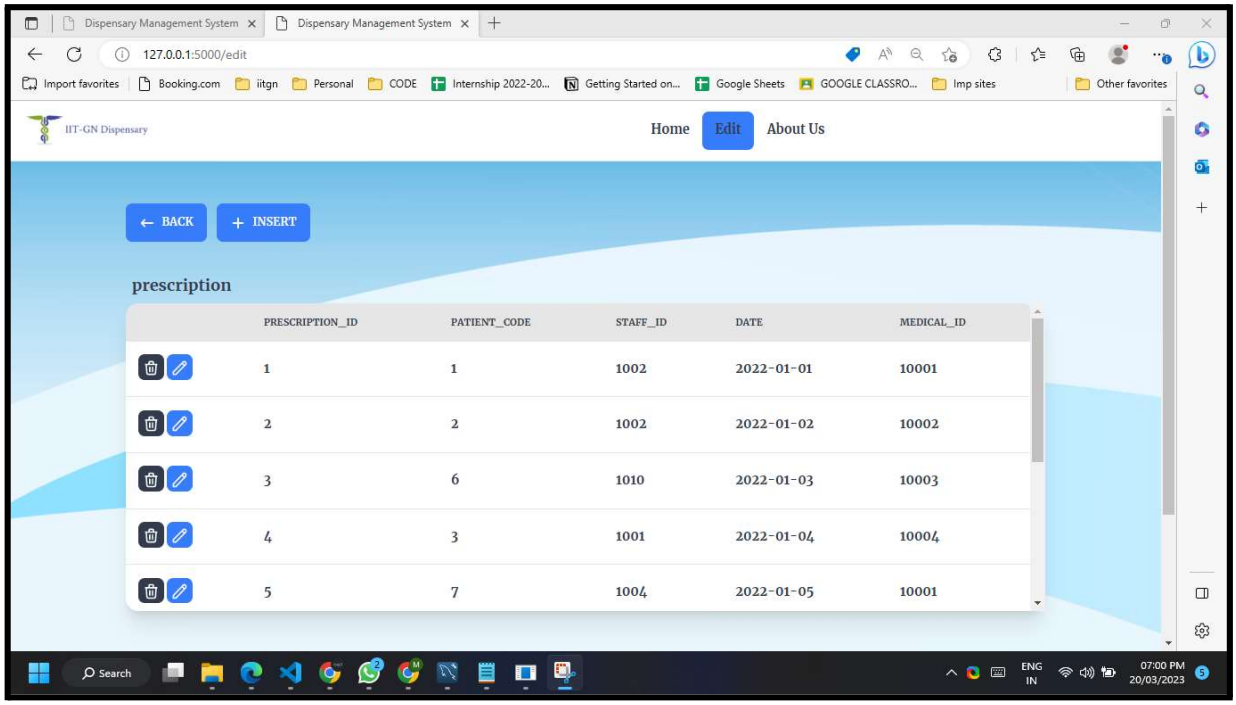
Content of the repository:

Screenshots of the Web Application showing all the Functionalities

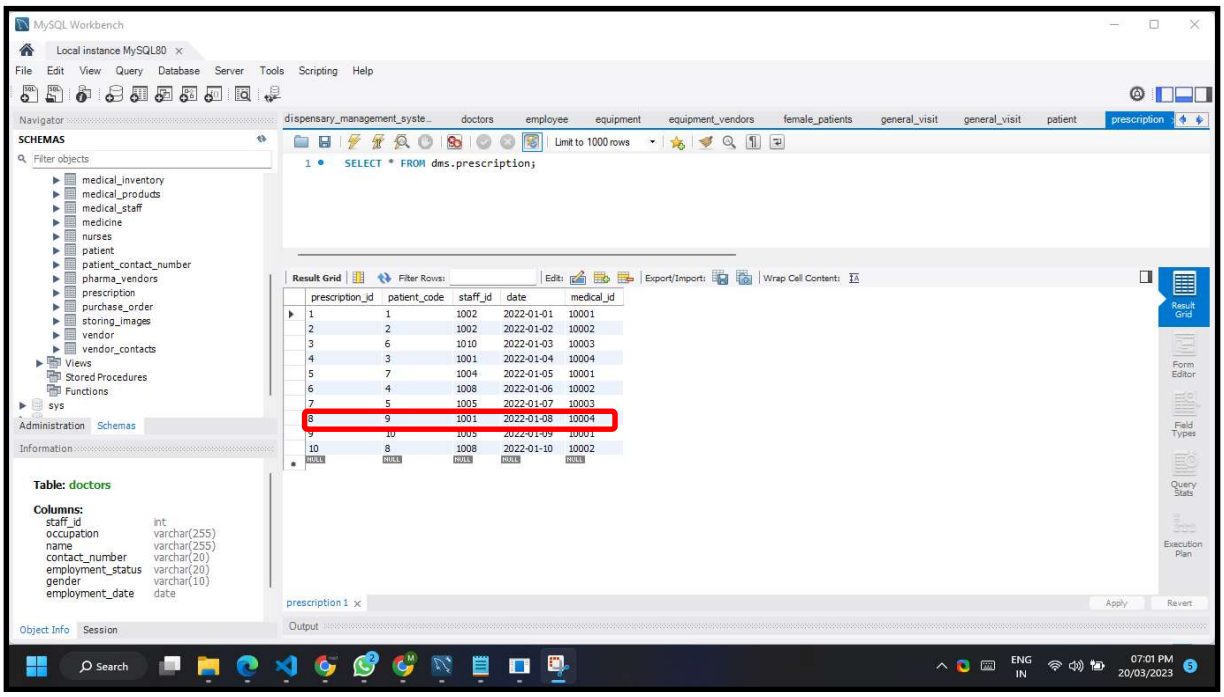
The web application shows the view of the different relations with the Insert, Update, and Delete operations on these relations. Screenshots include the view of the relation with the following operations and the results with the changes in the MySQL database itself. We also include the results which give an error due to some constraints defined by us while making the database, such as primary-key error etc.

Prescription Table

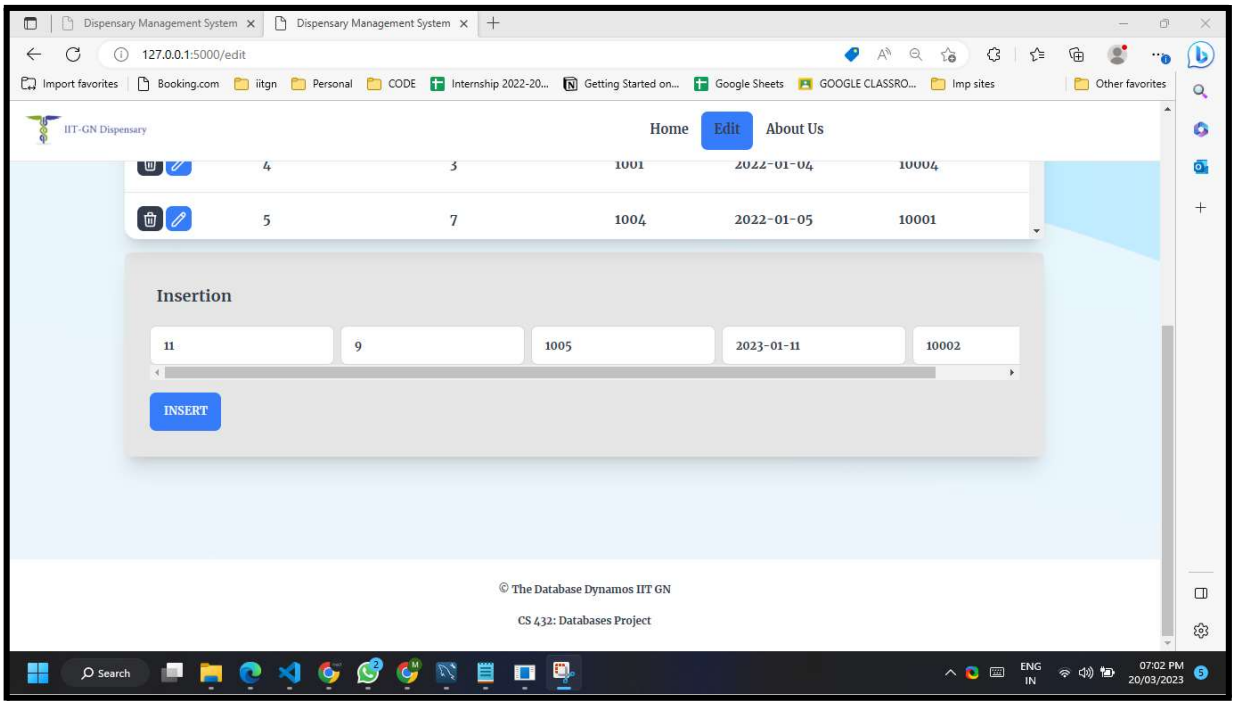
- View of Prescription Relation

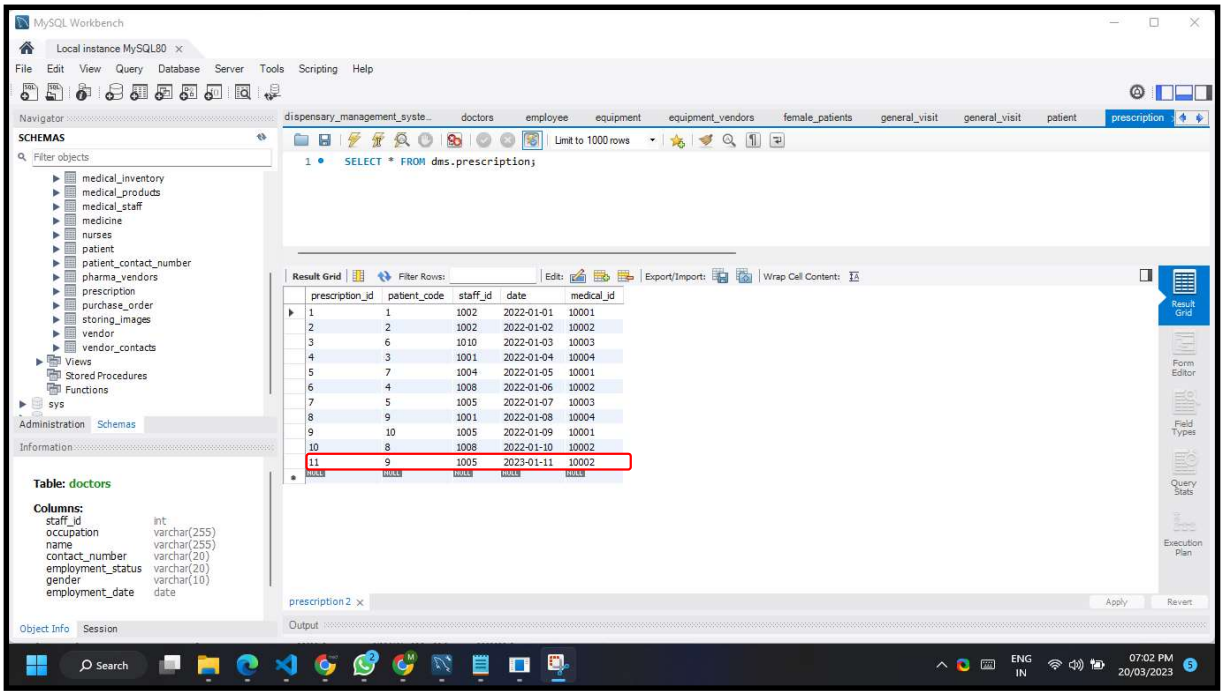


- View of Prescription Relation in Workbench

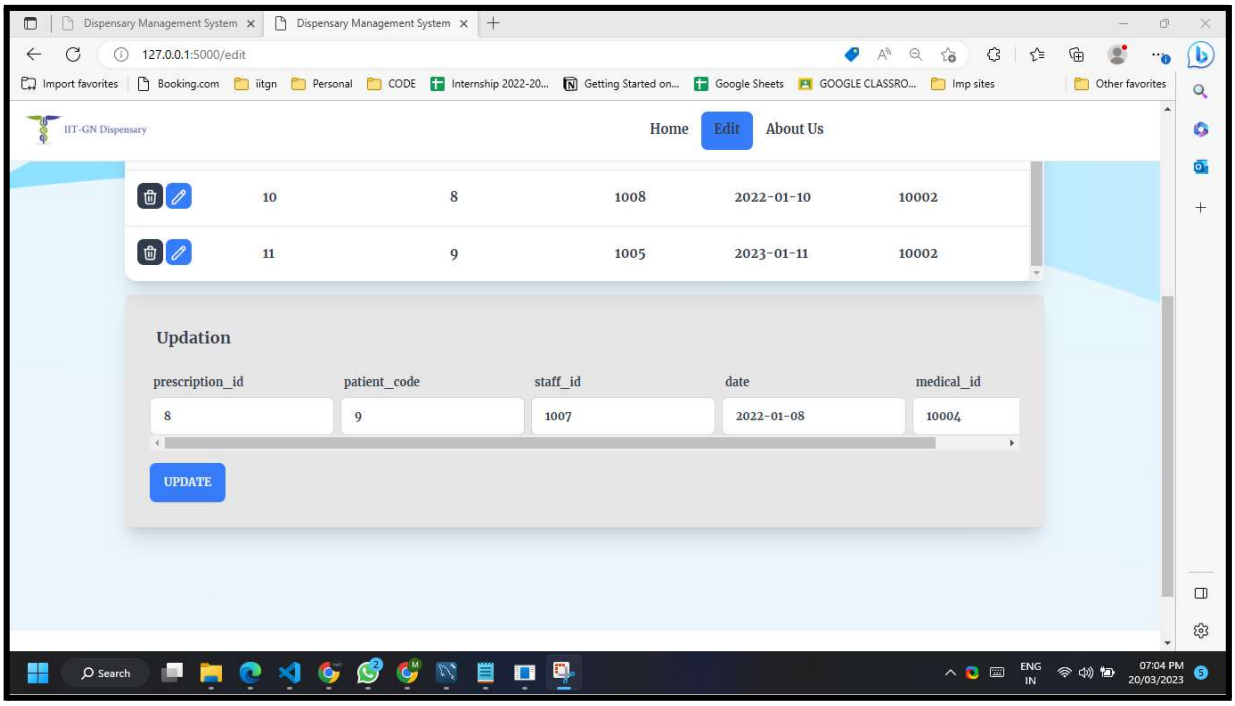


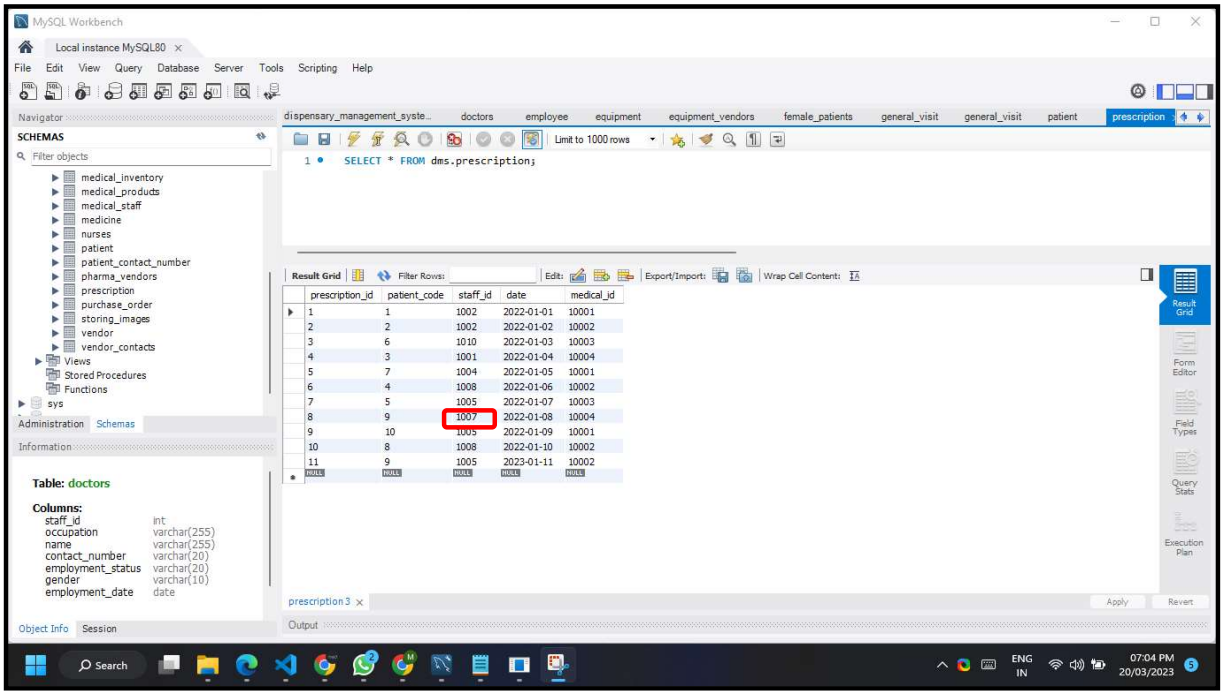
- Inserting Entries in Prescription Relation



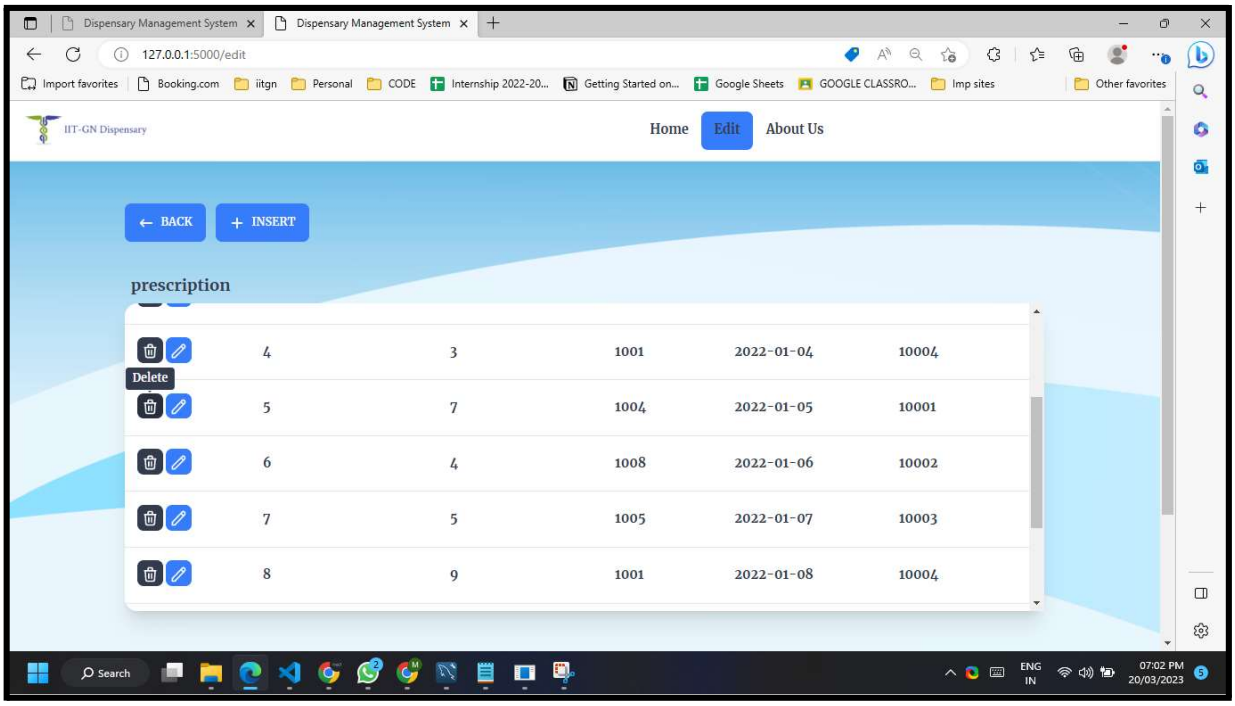


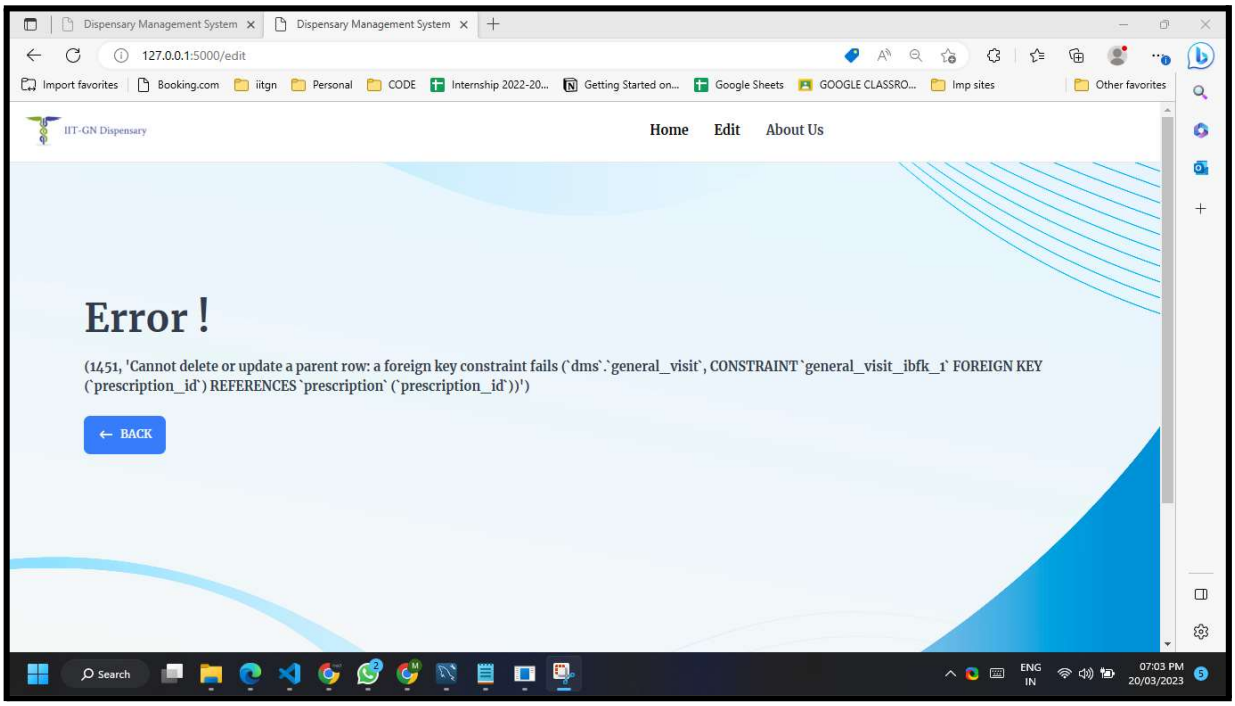
- Updating Entries in Prescription Relation





- Deleting Entries in Prescription Relation



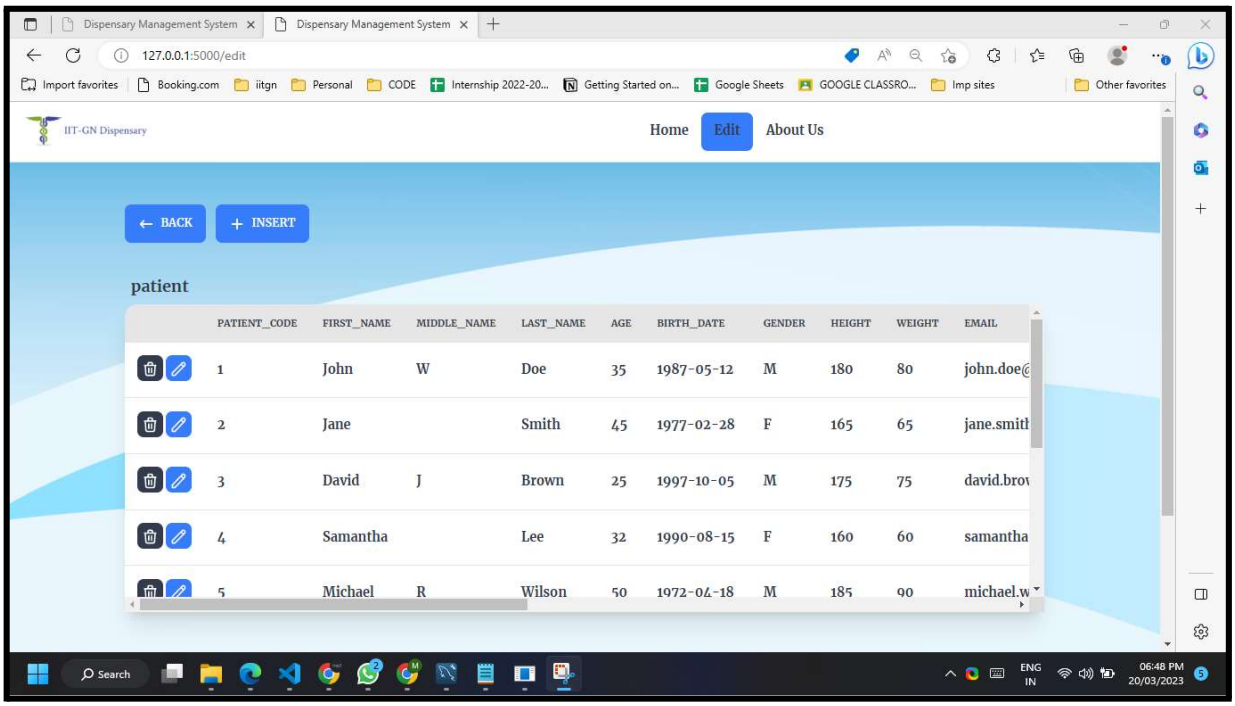


Reason for the error!

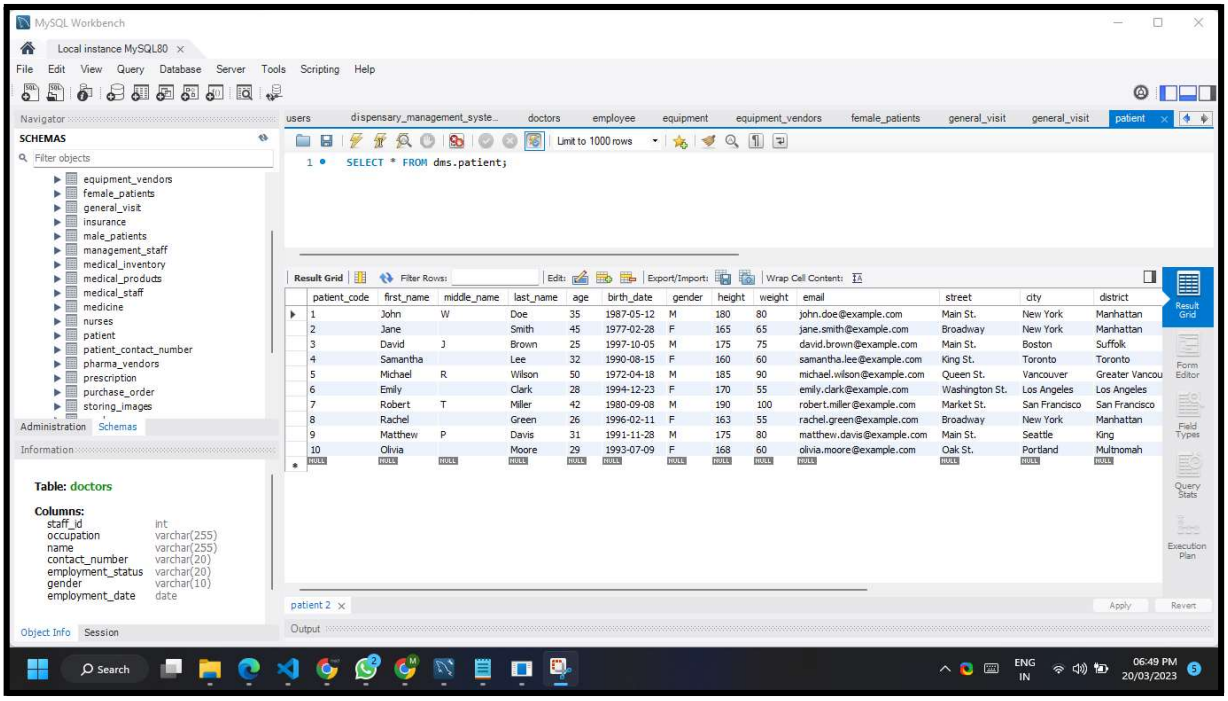
- Since the delete operation is performed in the Prescription Relation, which consists of the primary key of the patient, in order to hold the foreign key constraint in the delete operation, The above error is shown from MySQL and it is reflected on the page.

Patient Tabe

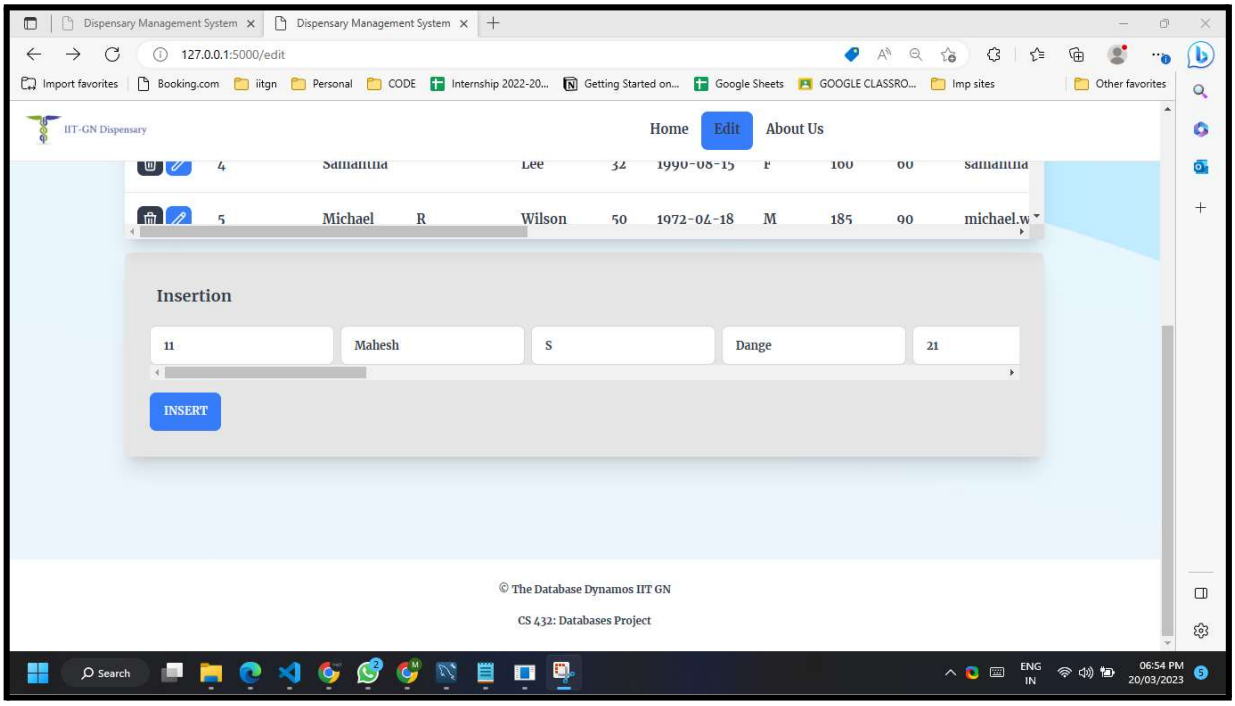
- View of Patient Relation

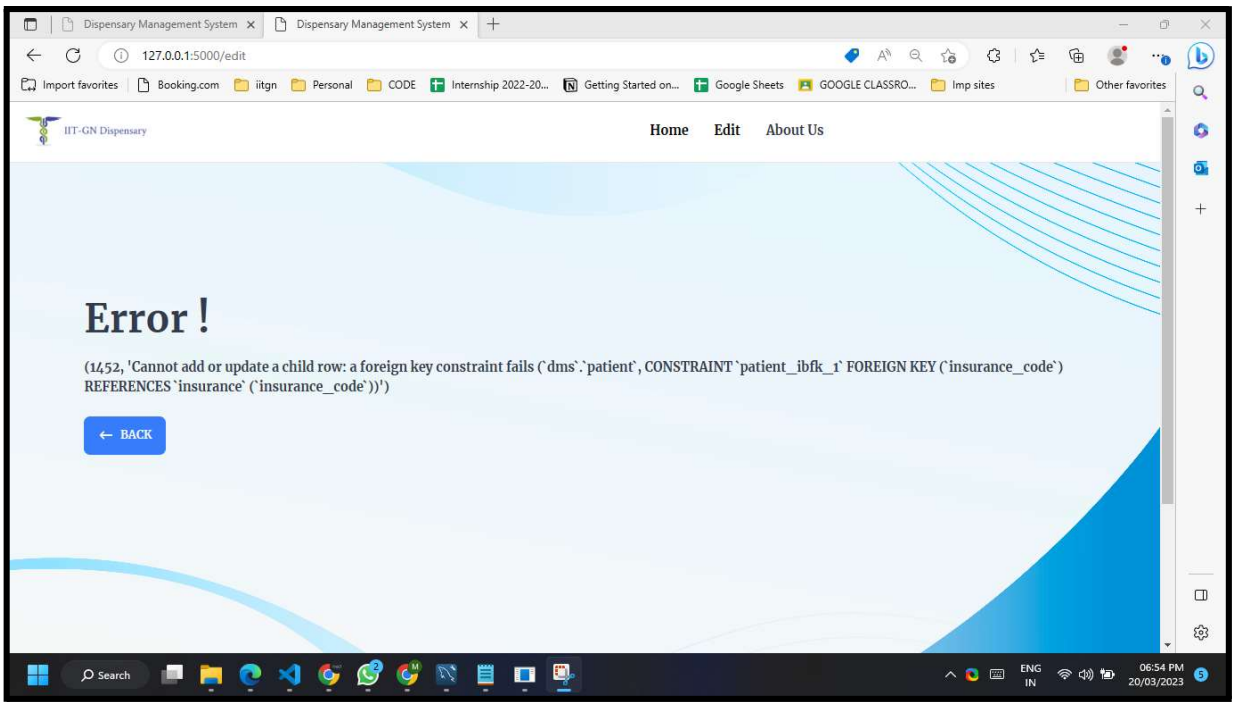


- View of Patient Relation in Workbench

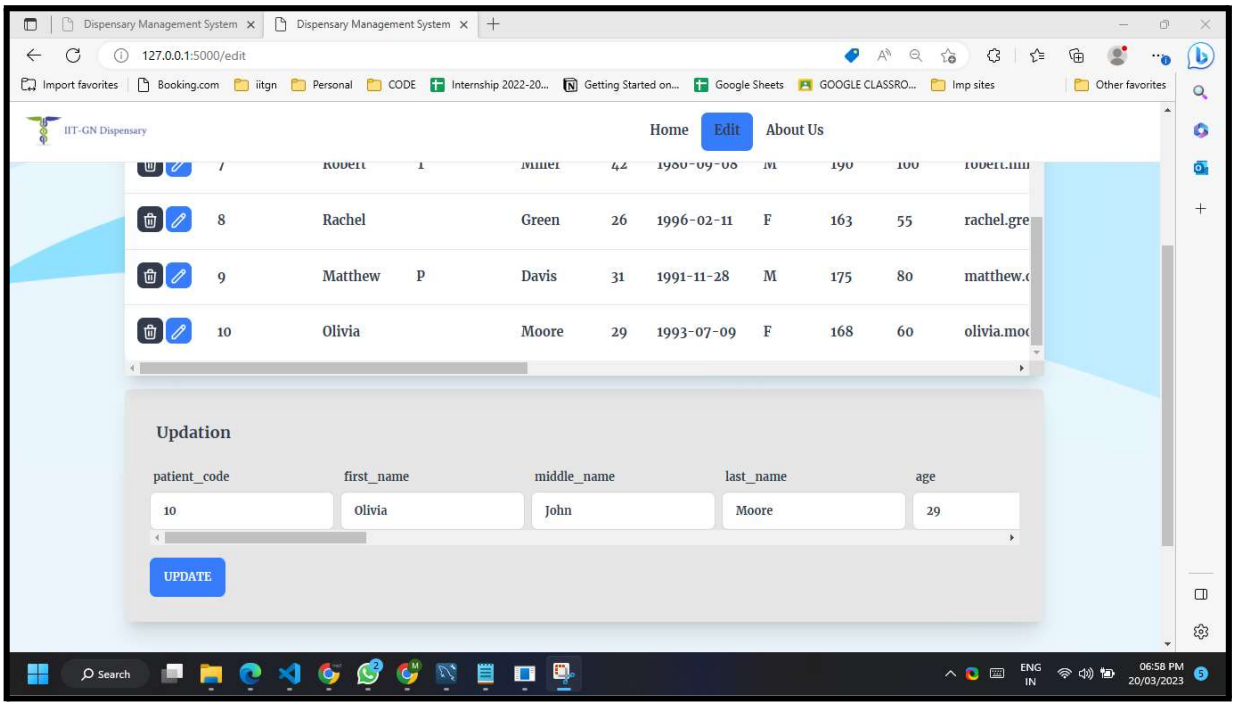


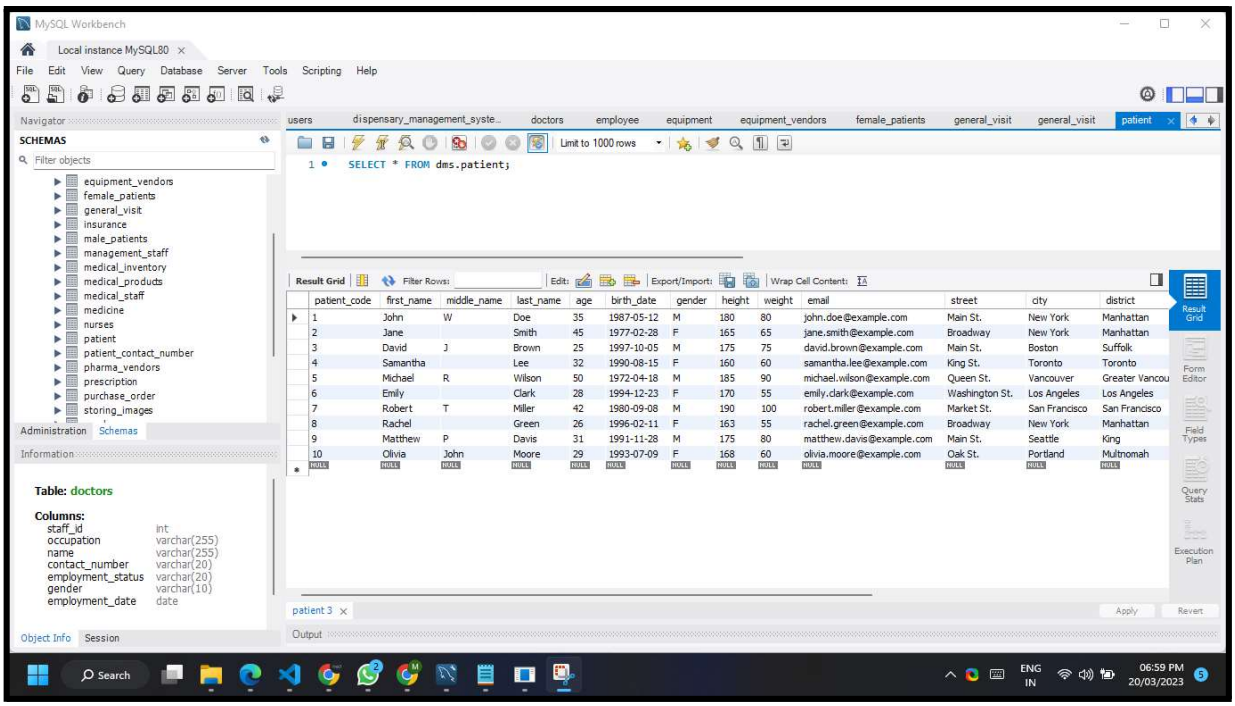
- Inserting Entries in Patient Relation



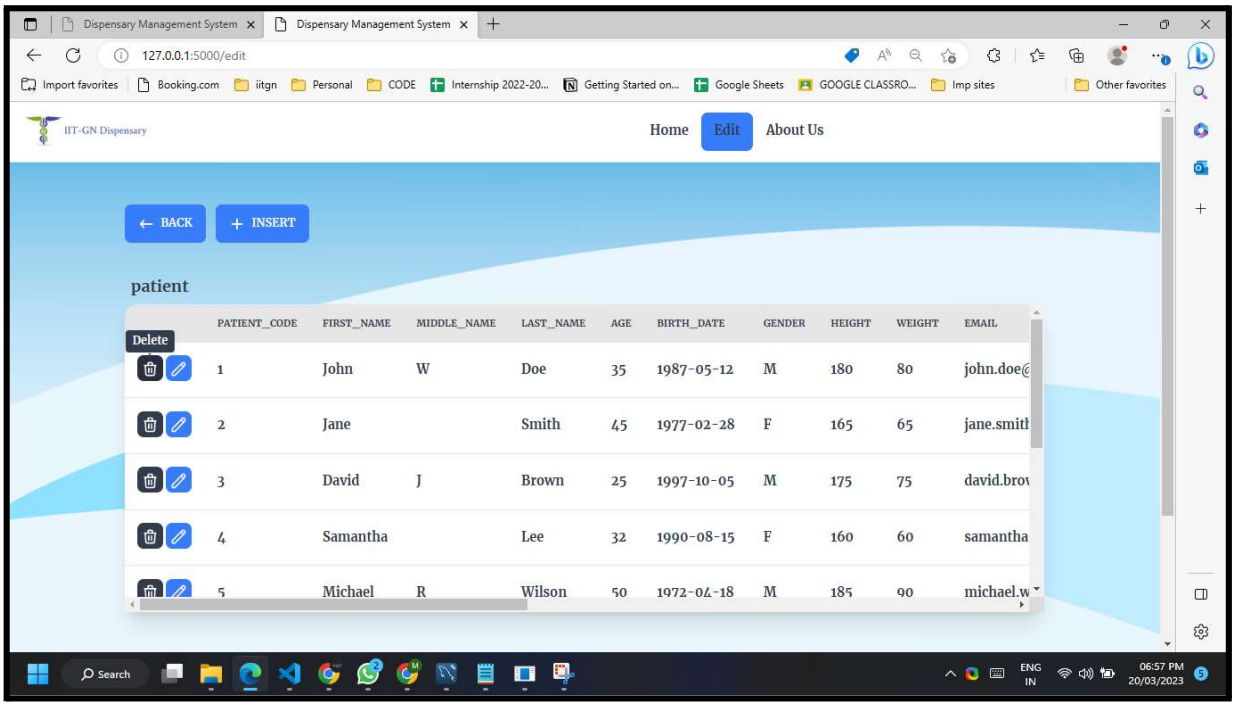


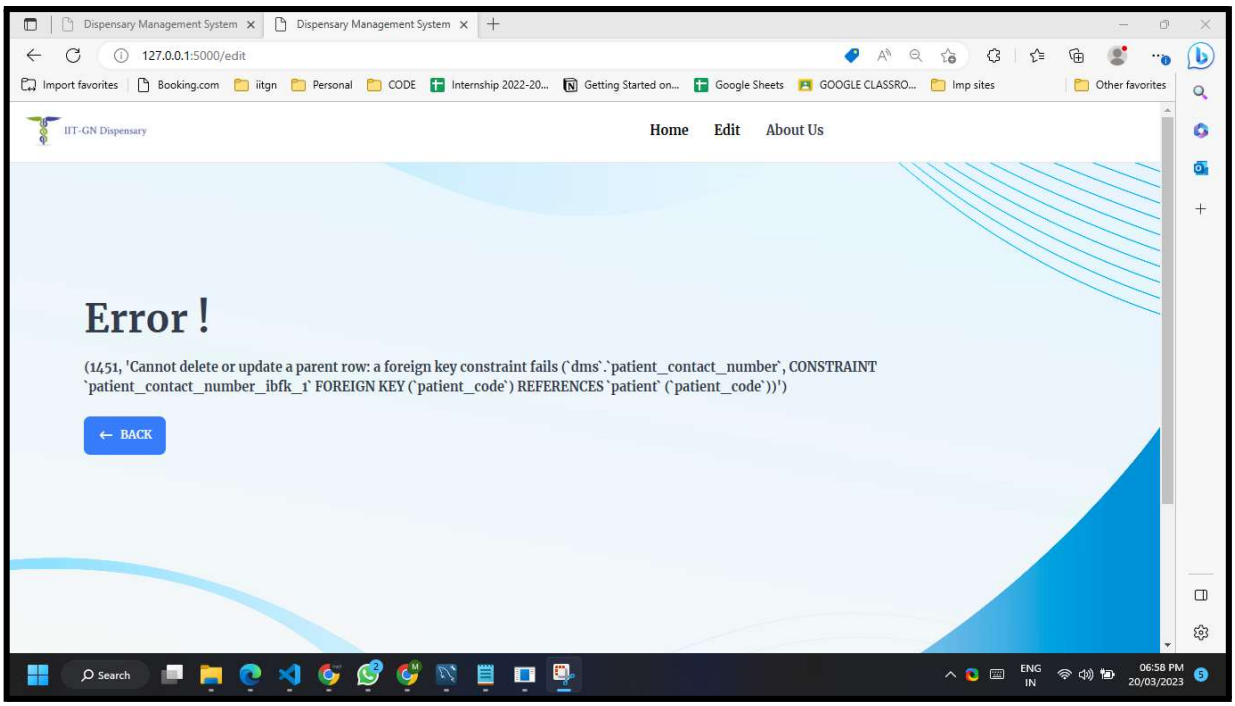
- Updating Entries in Patient Relation





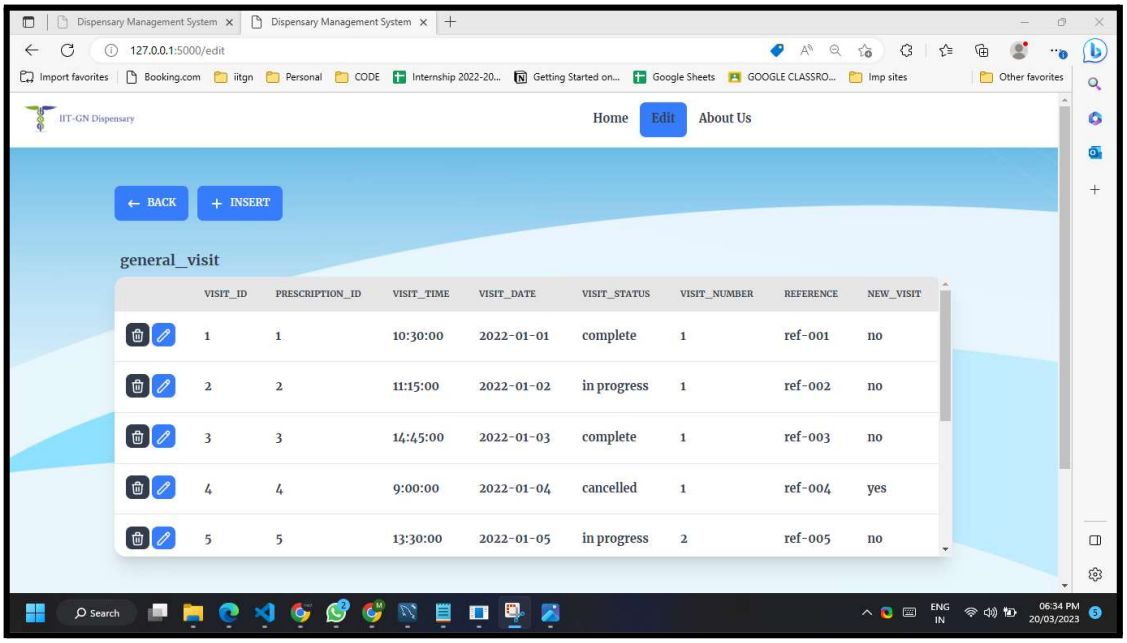
- Deleting Entries in Patient Relation



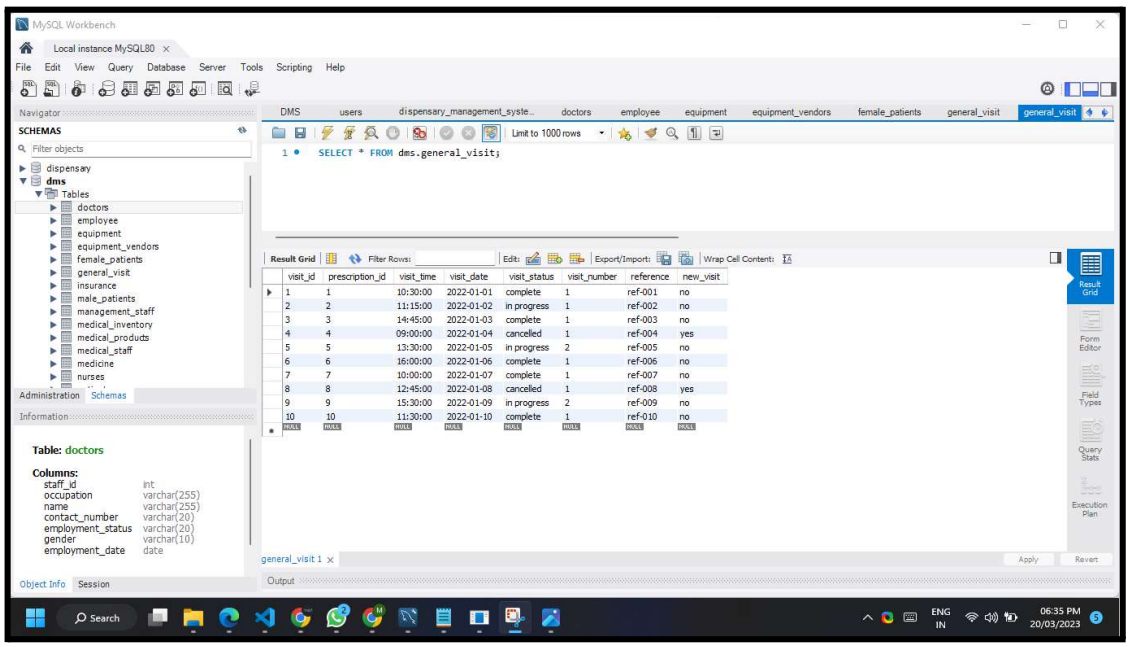

Dispensary Management System
Dispensary Management System
127.0.0.1:5000/edit
IIT-GN Dispensary
Home
Edit
About Us
Error !
(1451, 'Cannot delete or update a parent row: a foreign key constraint fails (`dms`.`patient_contact_number`, CONSTRAINT `patient_contact_number_ibfk_1` FOREIGN KEY (`patient_code`) REFERENCES `patient` (`patient_code`))')
BACK

General Visit Table

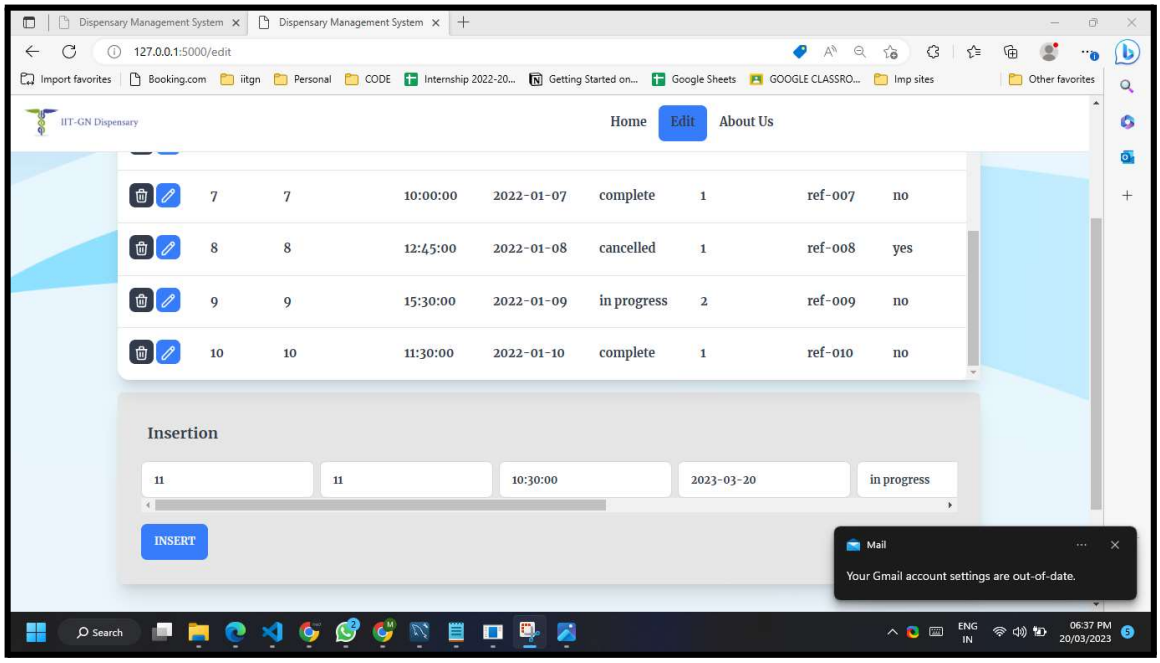
- View of General Visit Relation

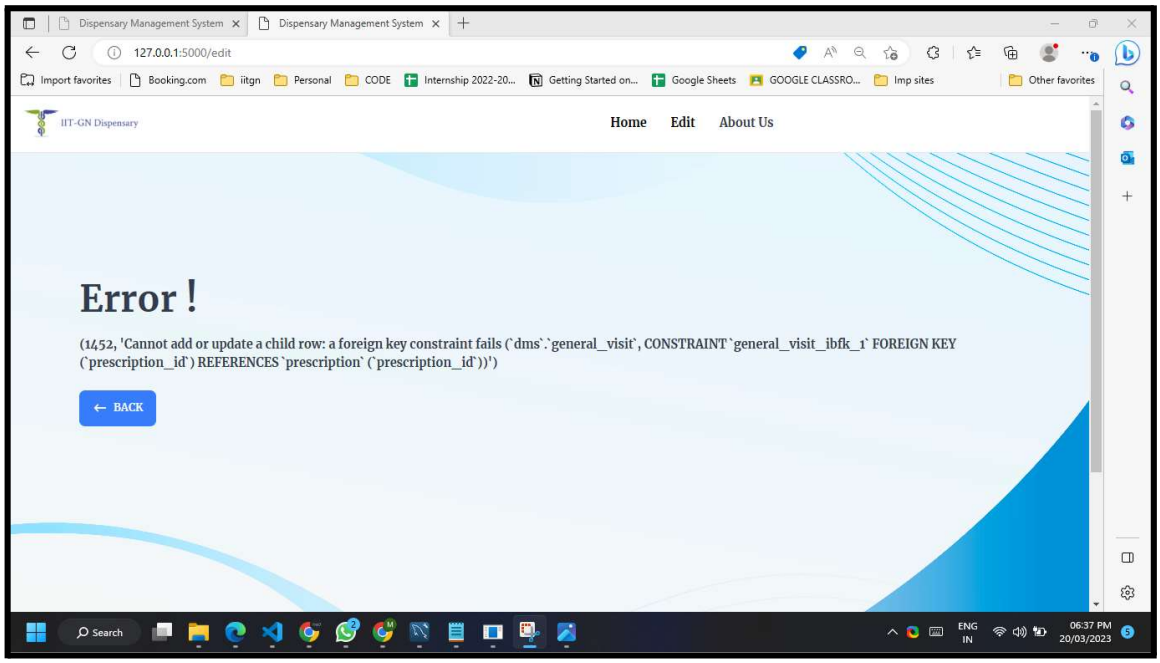


- View of General Visit Relation in Workbench

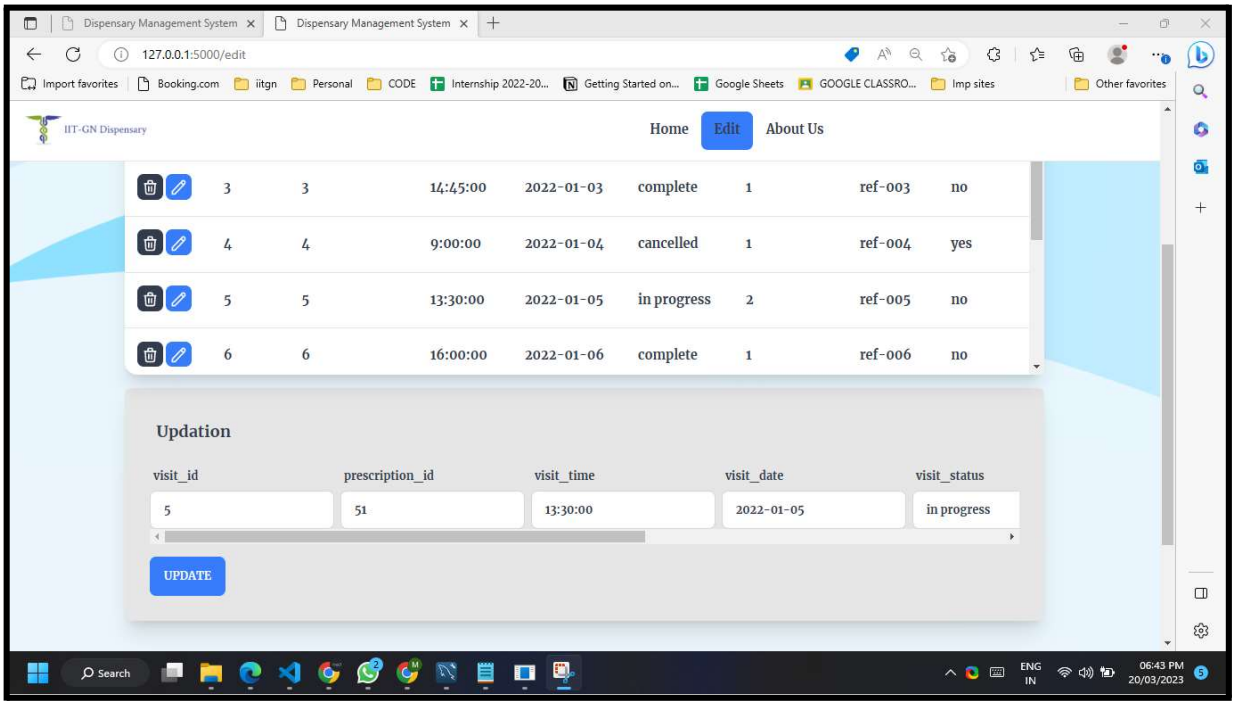


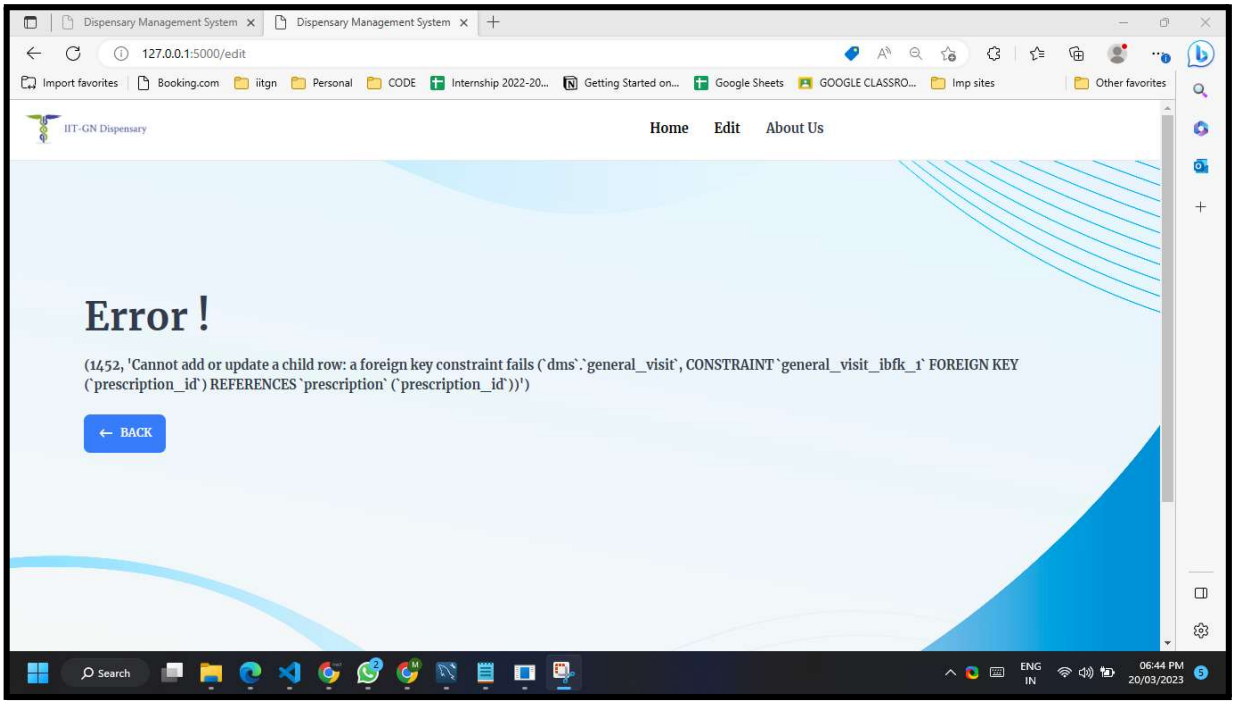
- Inserting General Visit in Prescription Relation



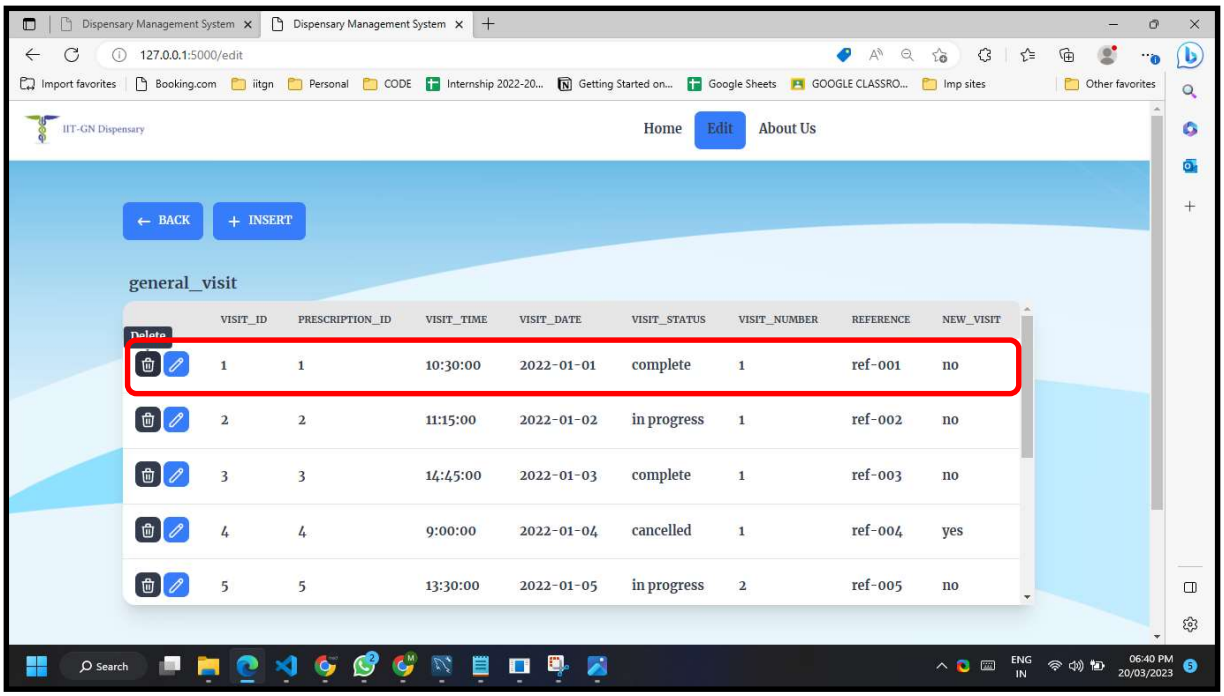


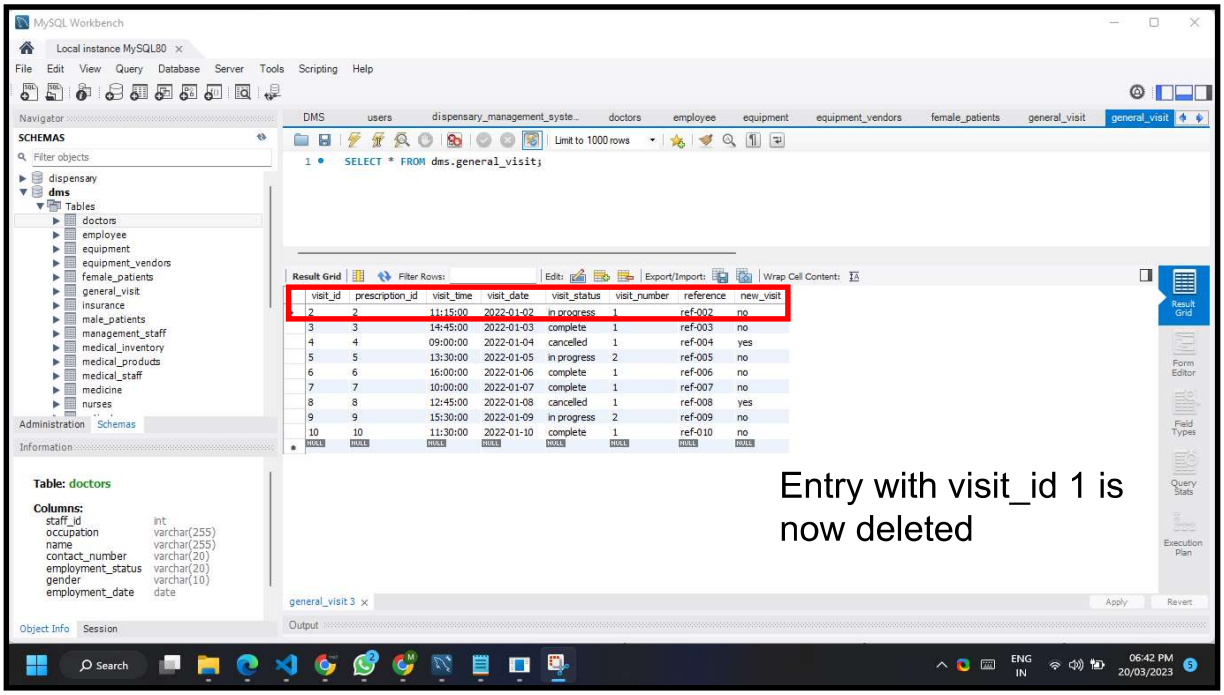
- Updating Entries in General Visit Relation





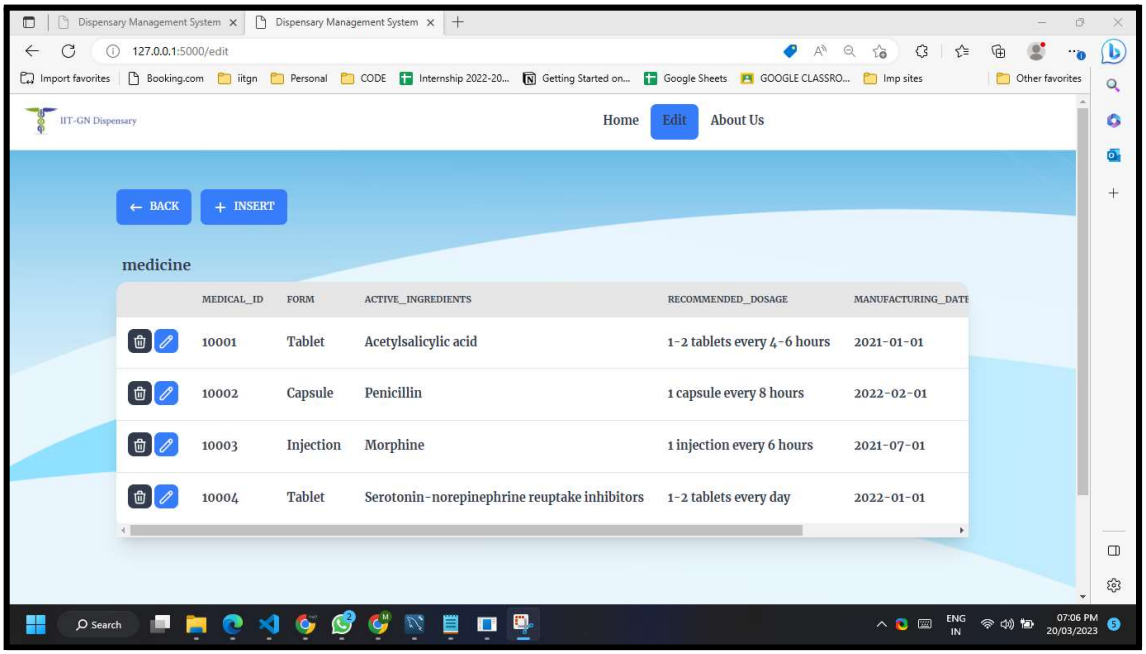
- Deleting Entries in General Visit Relation



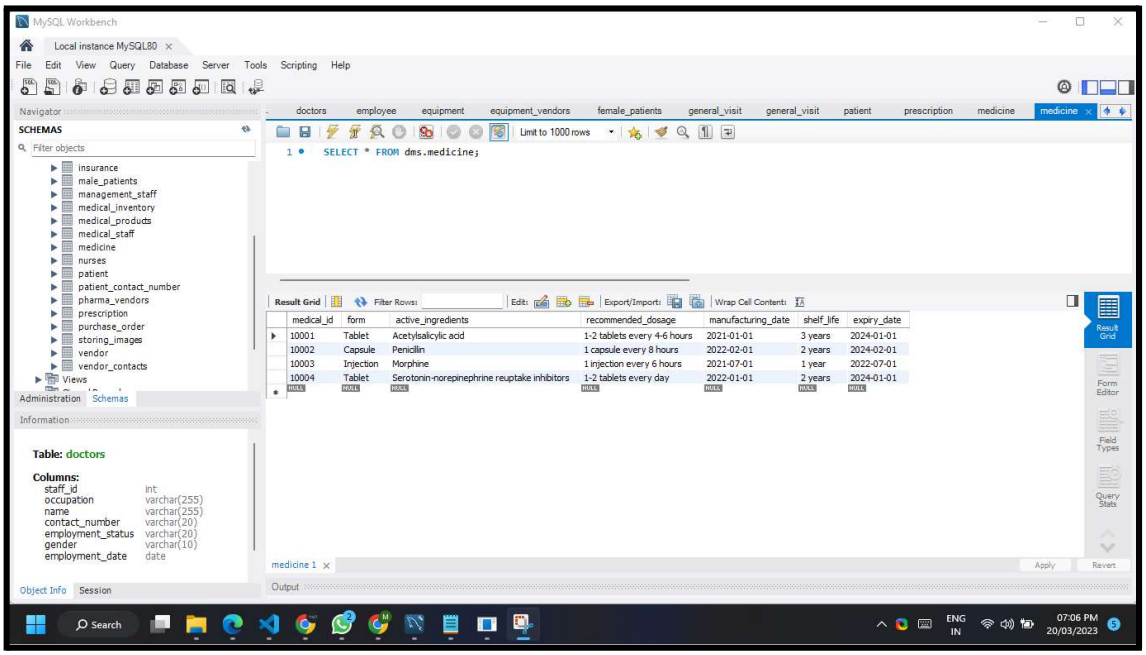


Medicine Table

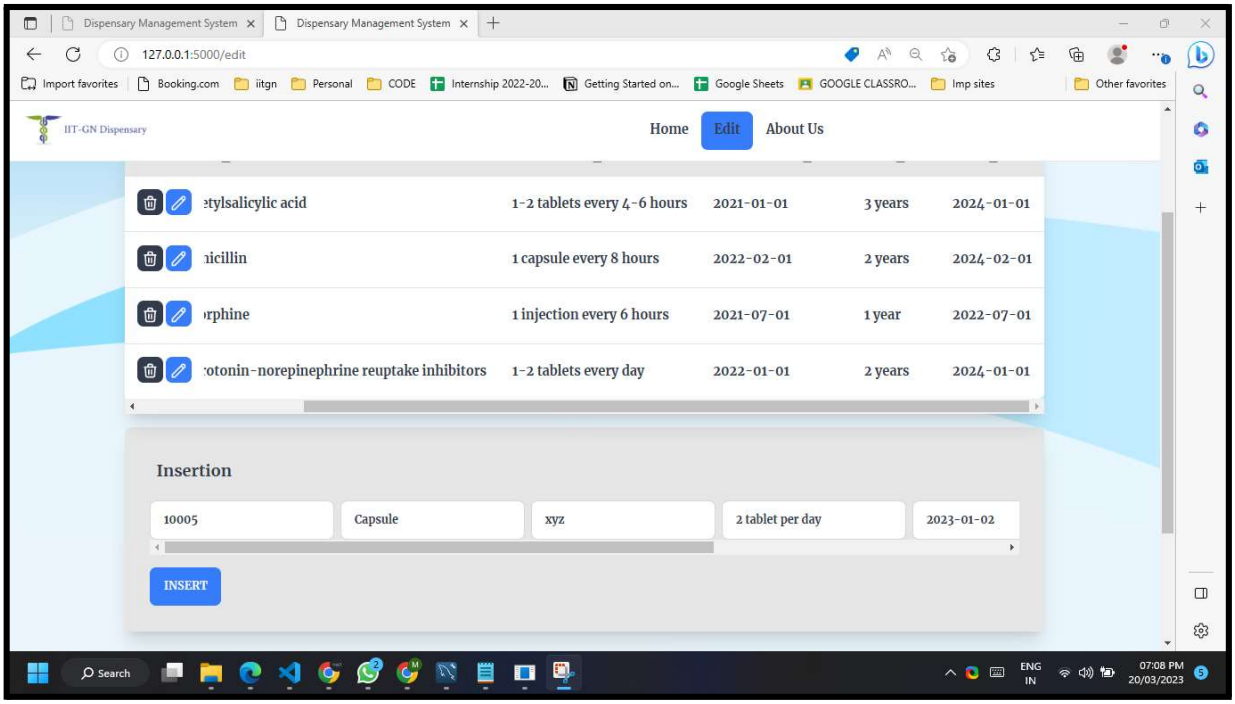
- View of Medicine Relation

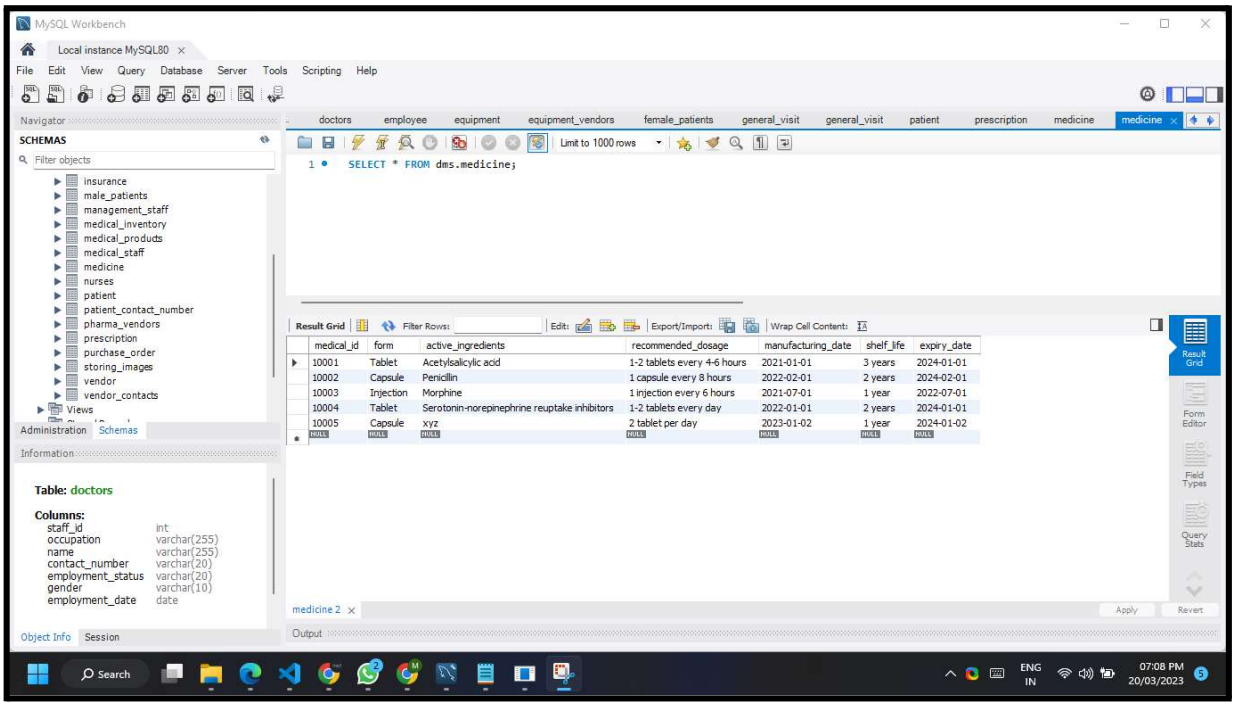


- View of Medicine Relation in Workbench

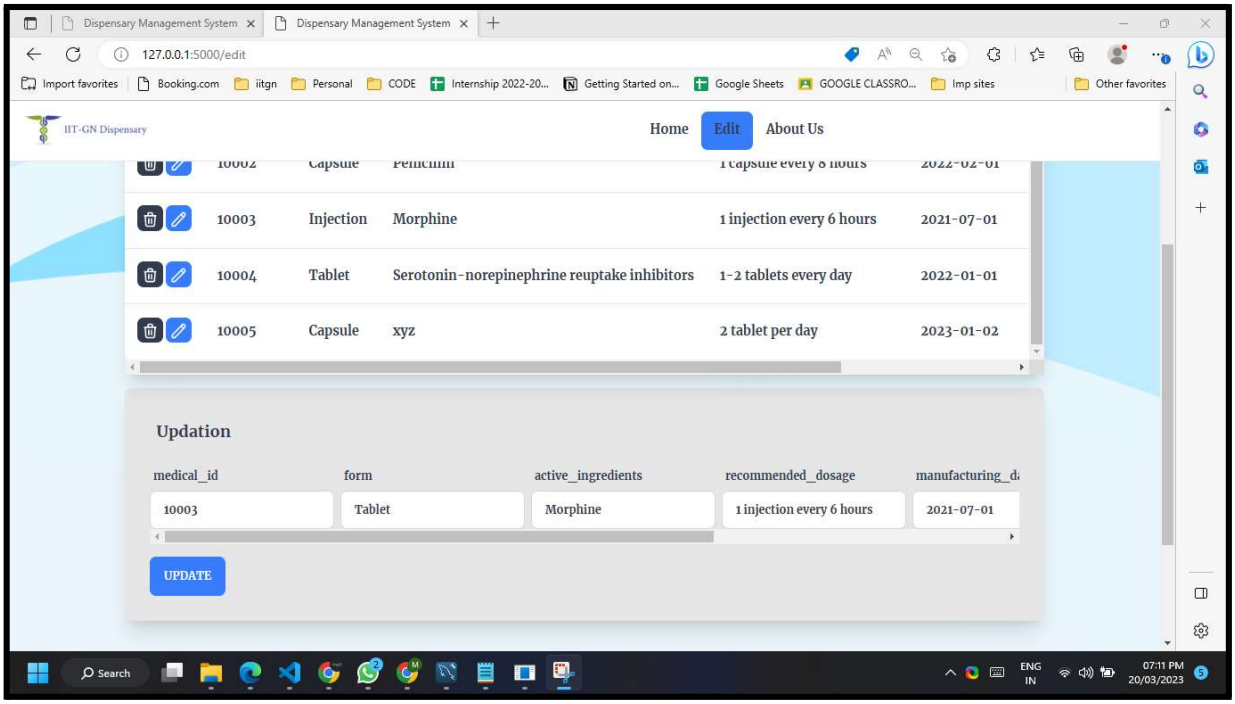


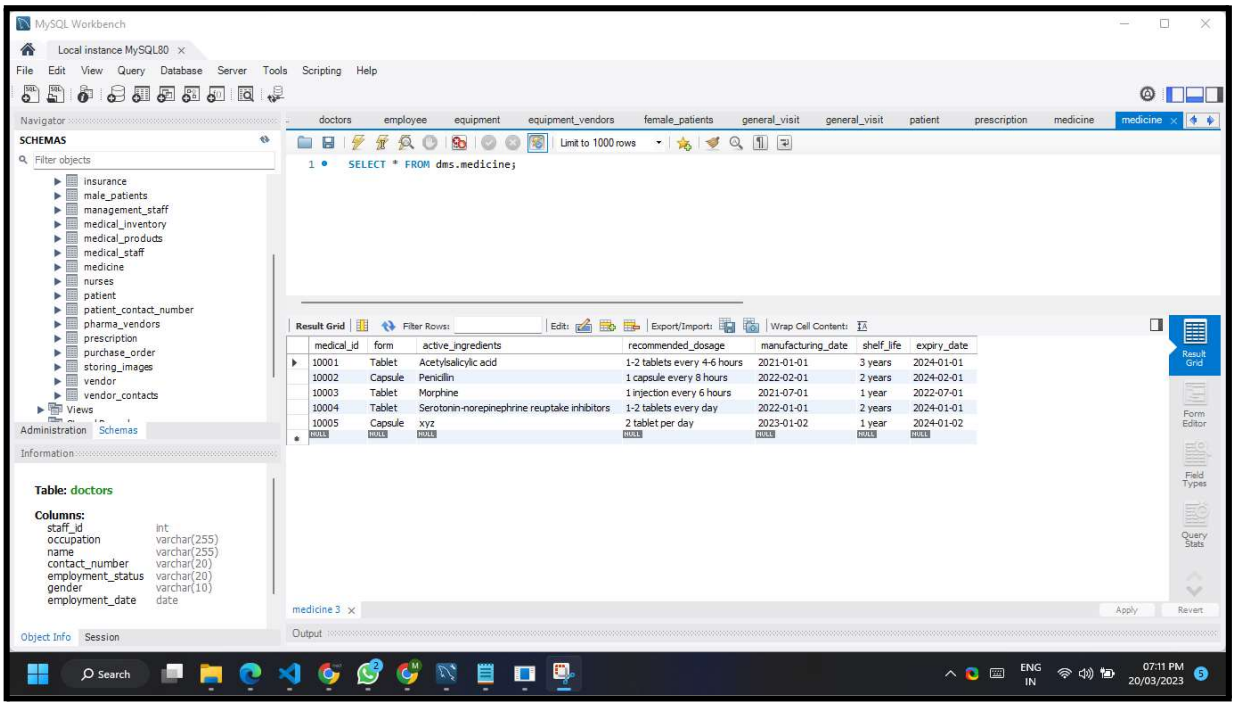
- Inserting Medicine in Prescription Relation



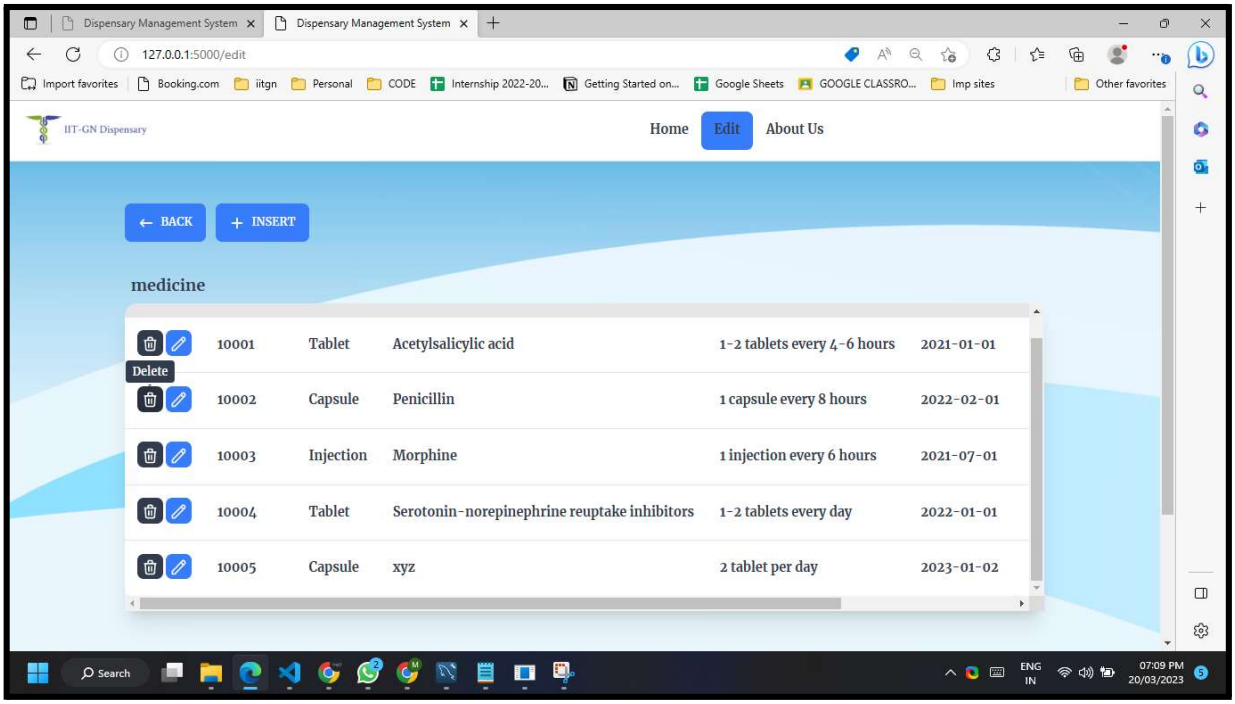


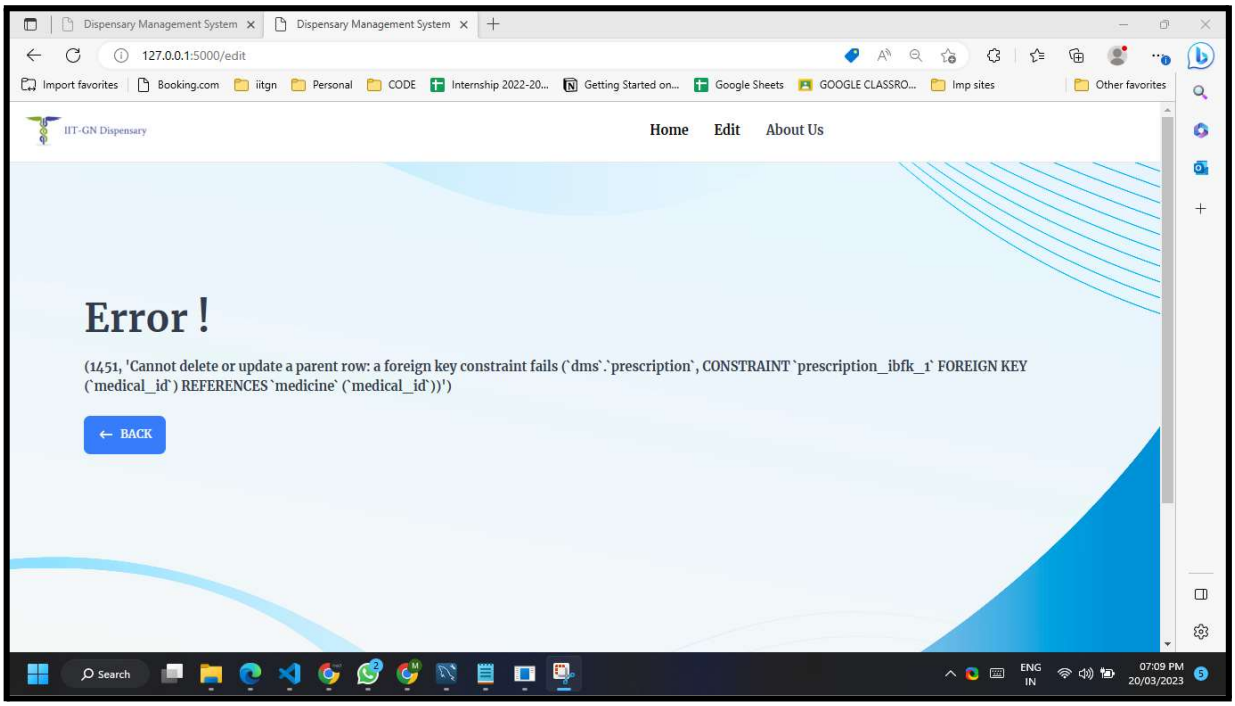
- Updating Entries in Medicine Relation





- Deleting Entries in Medicine Relation





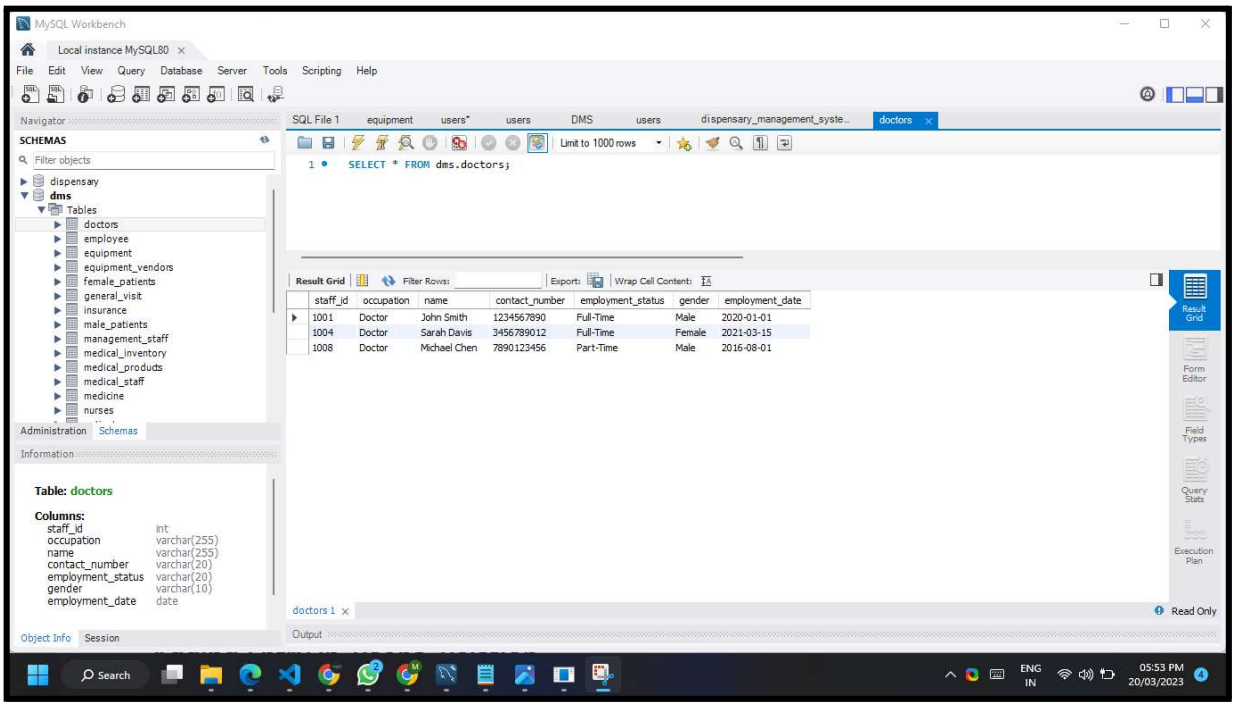
Doctor Table

- View of Doctor Relation

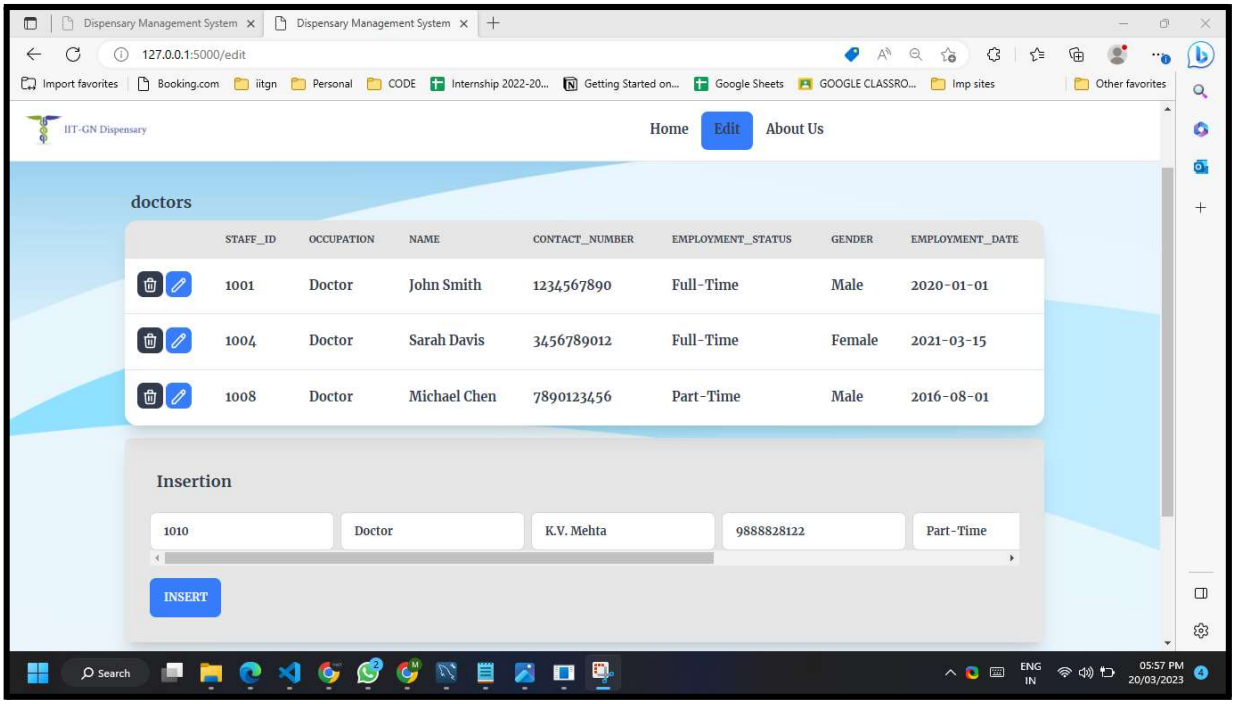
doctors

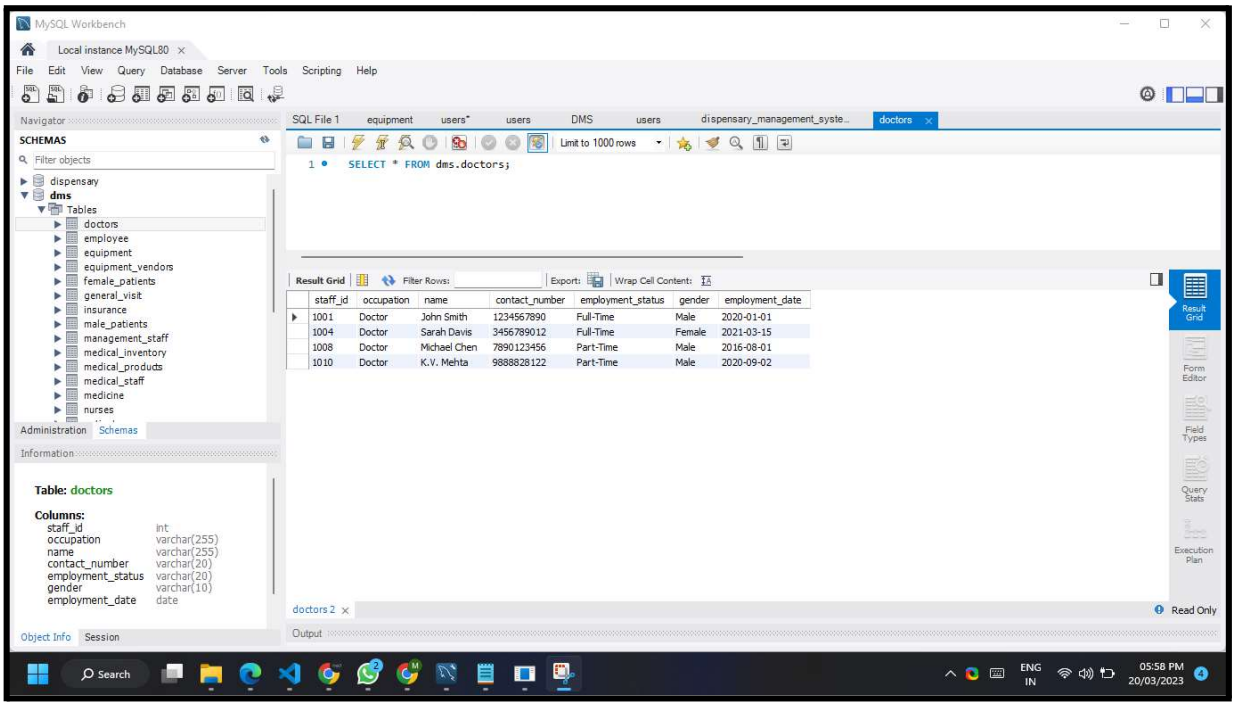
	STAFF_ID	OCCUPATION	NAME	CONTACT_NUMBER	EMPLOYMENT_STATUS	GENDER	EMPLOYMENT_DATE
	1001	Doctor	John Smith	1234567890	Full-Time	Male	2020-01-01
	1004	Doctor	Sarah Davis	3456789012	Full-Time	Female	2021-03-15
	1008	Doctor	Michael Chen	7890123456	Part-Time	Male	2016-08-01

- View of Doctor Relation in Workbench

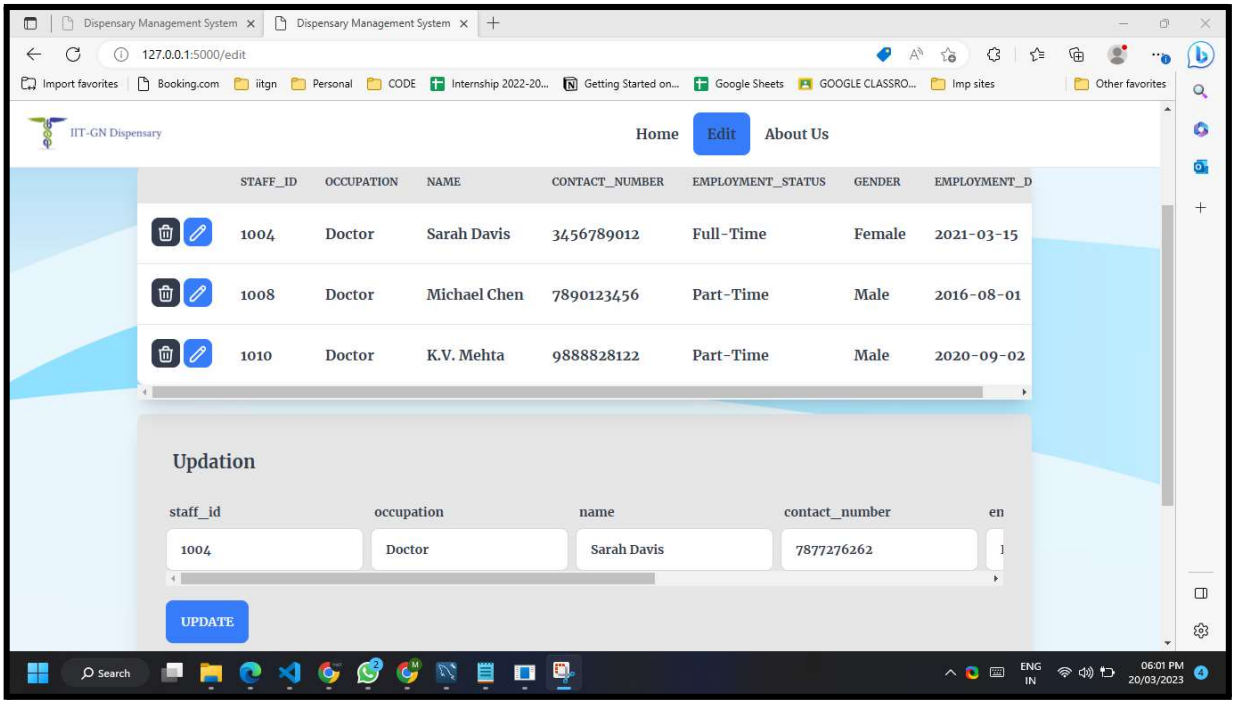


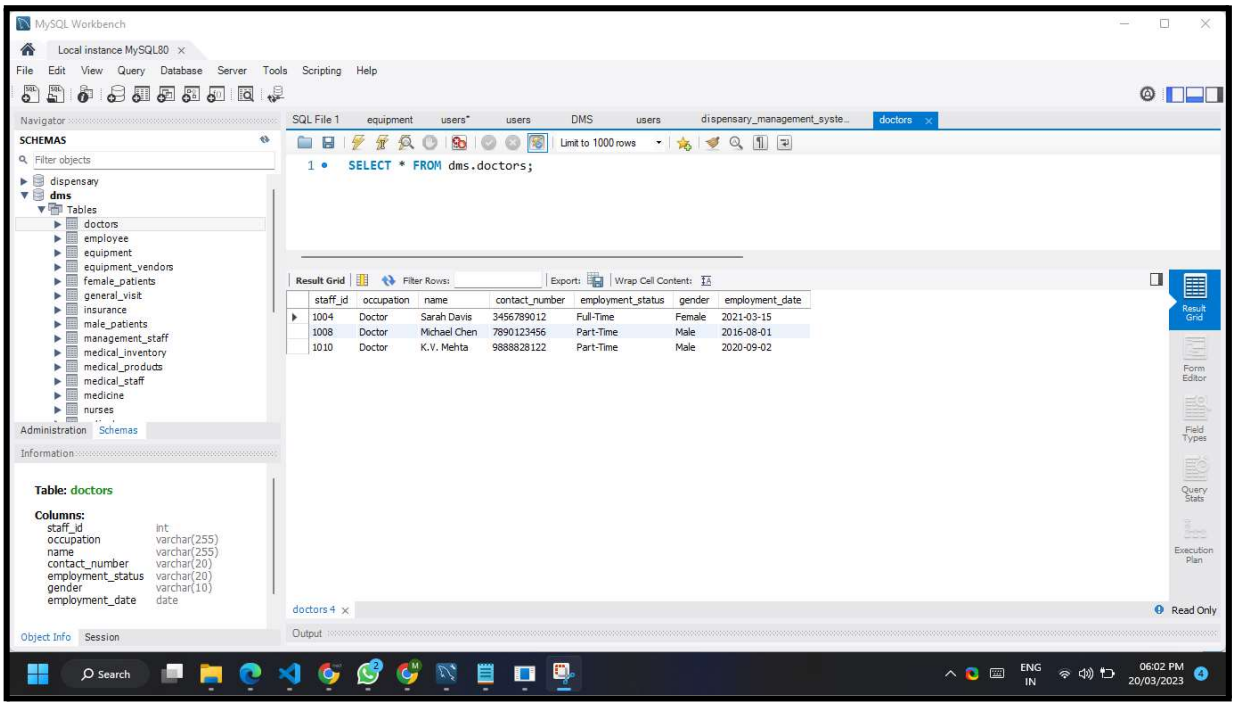
- Inserting Doctor in Prescription Relation



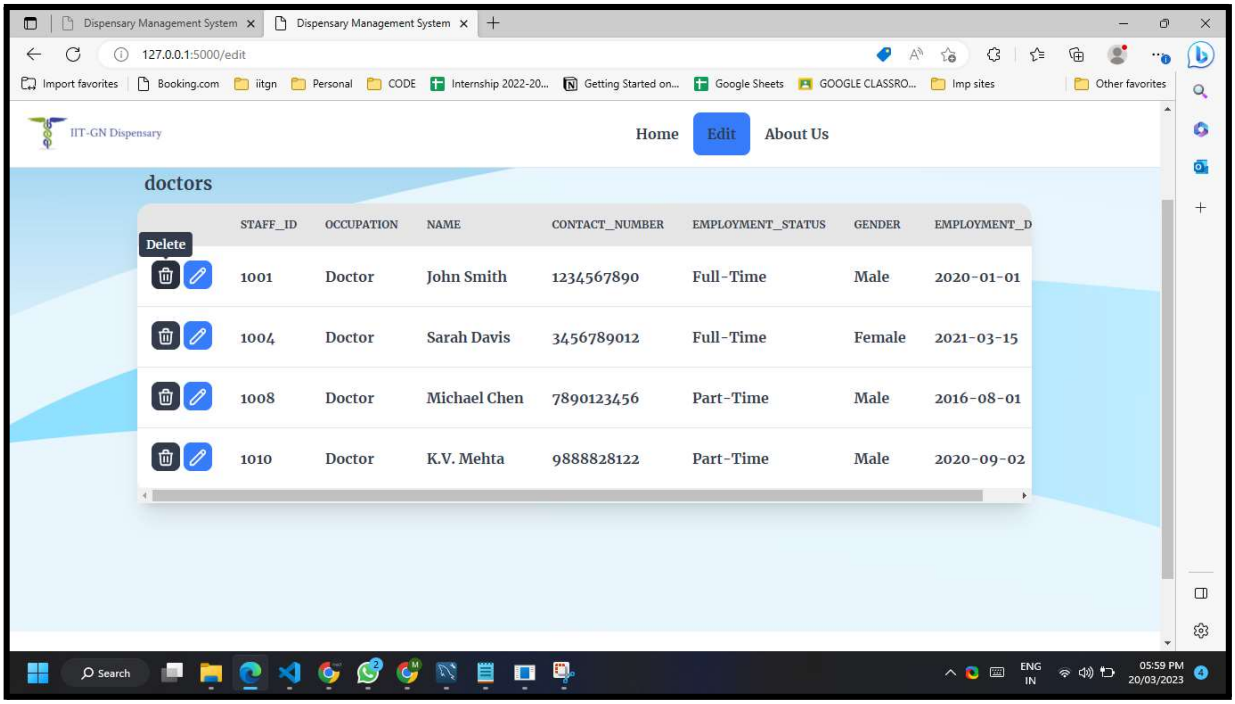


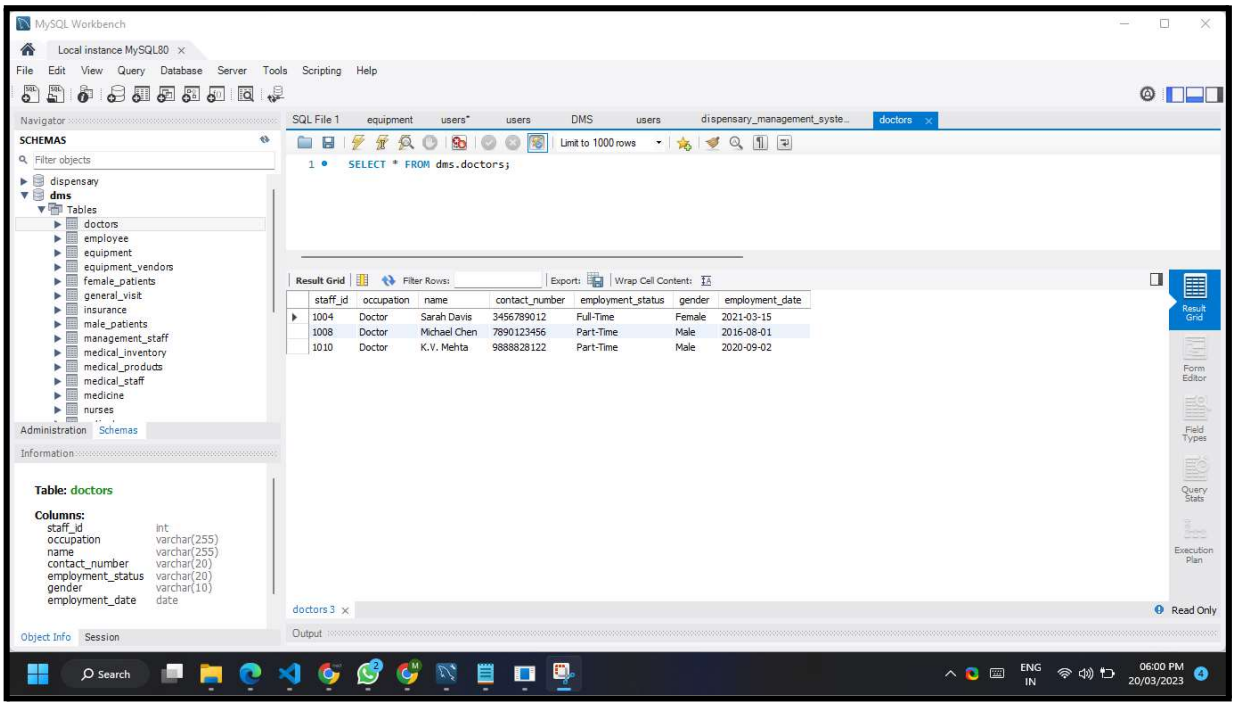
- Updating Entries in Doctor Relation





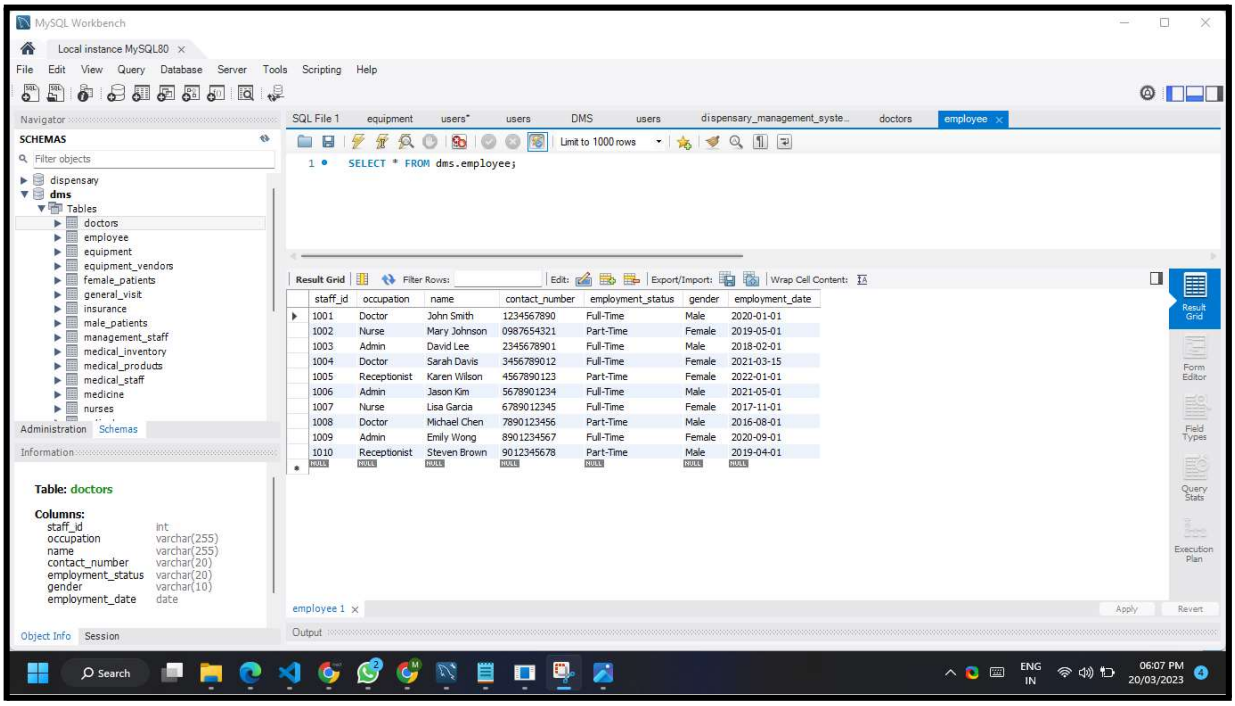
- Deleting Entries in Doctor Relation

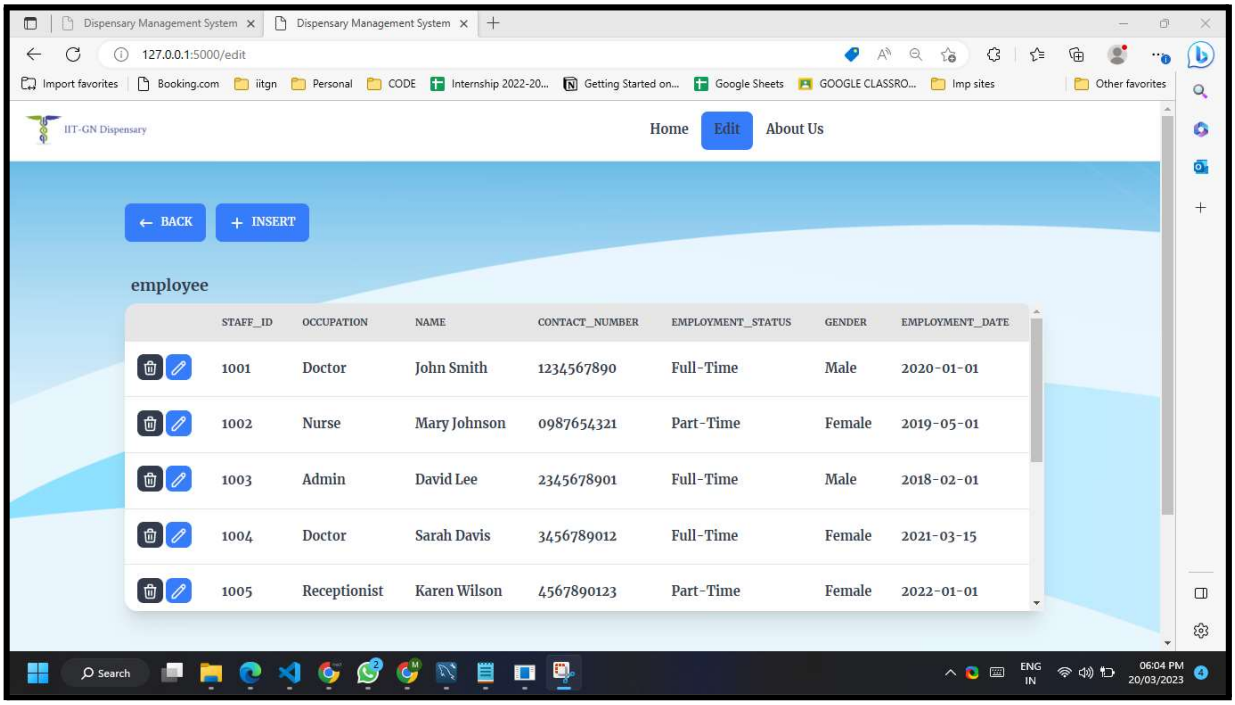




Employee Table

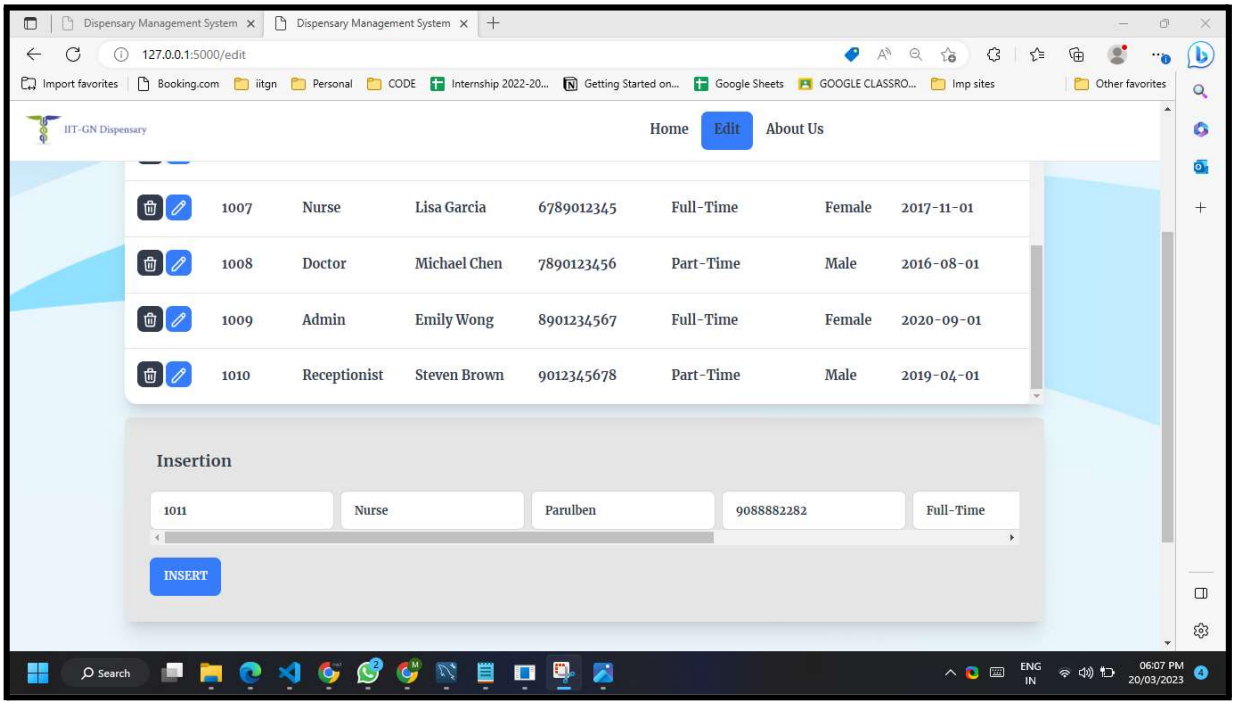
- View of Employee Relation

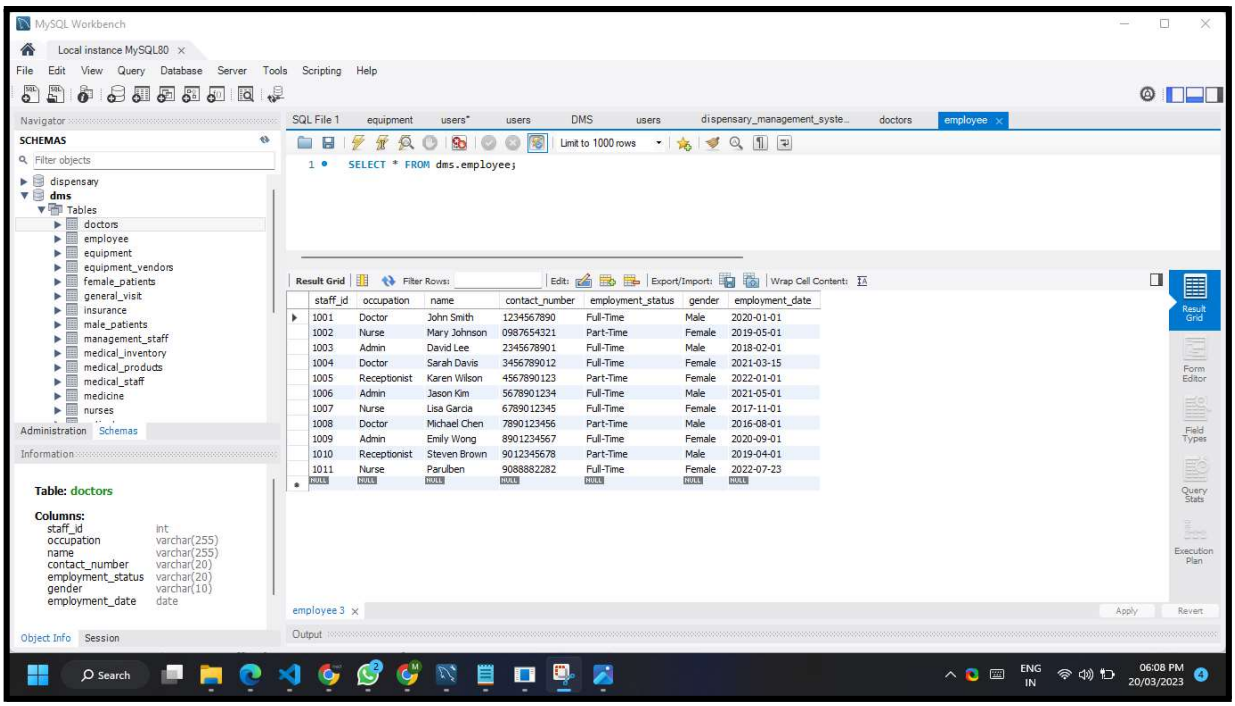




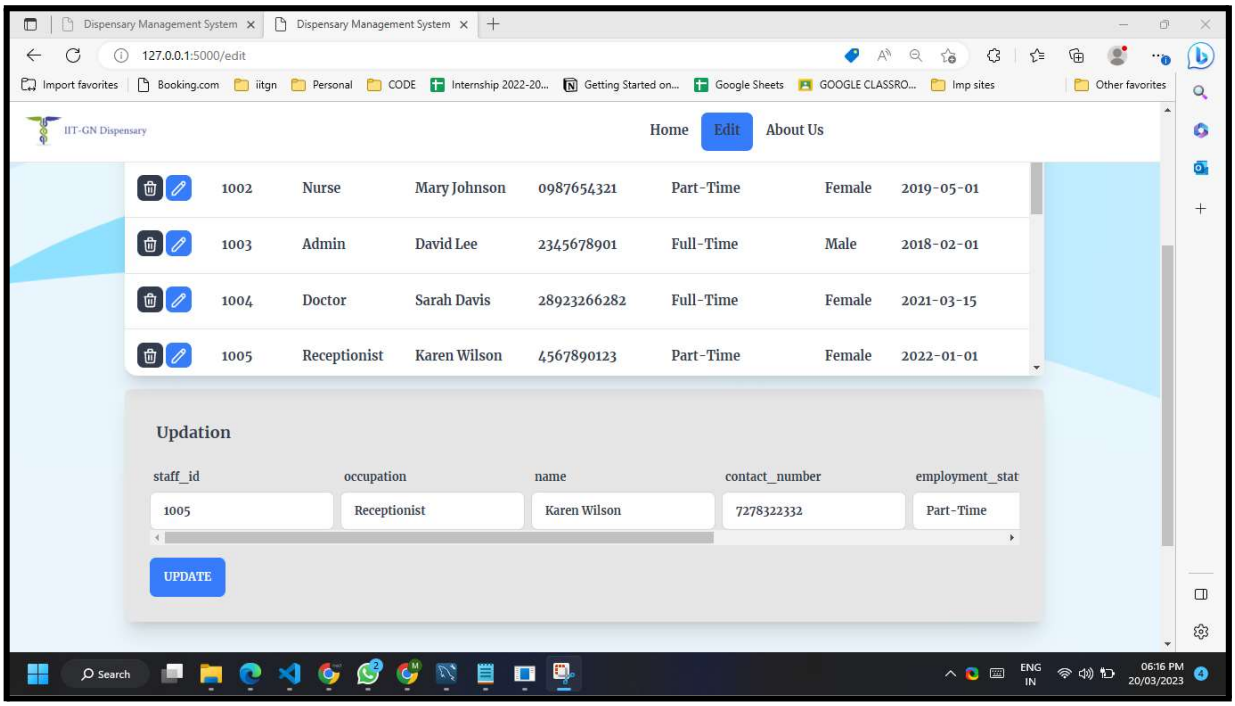
- View of Employee Relation in Workbench

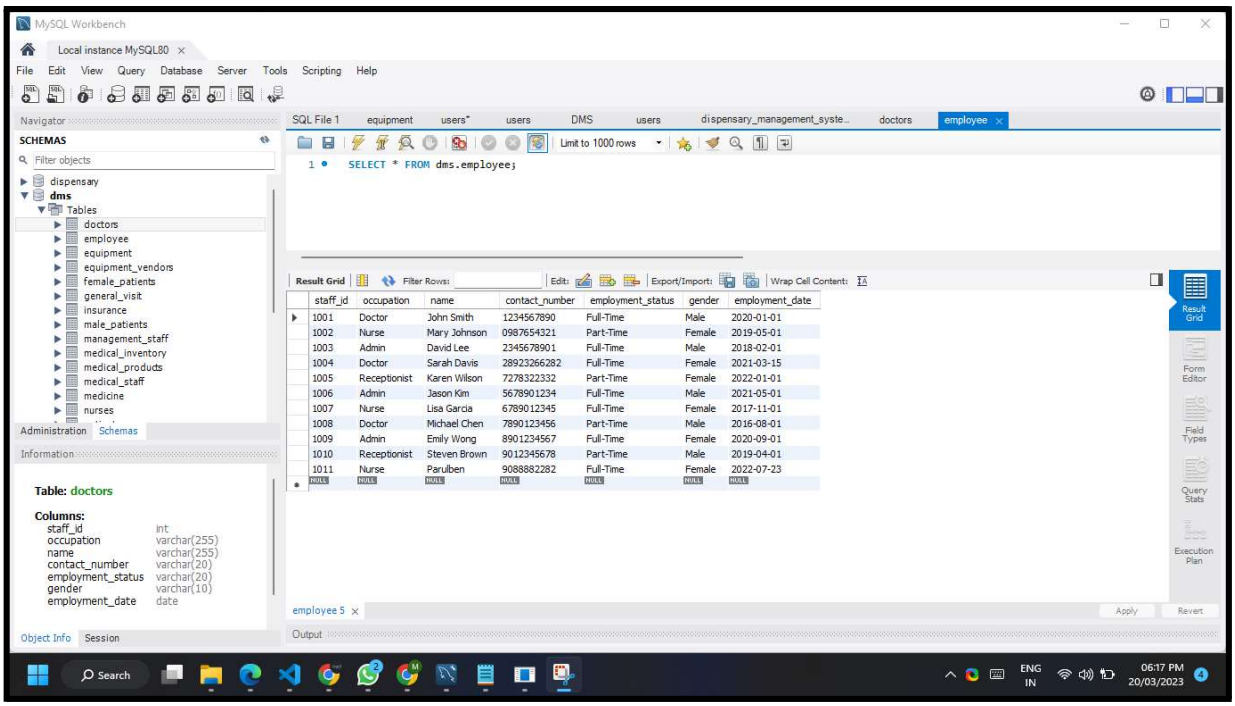
- Inserting Doctor in Employee Relation



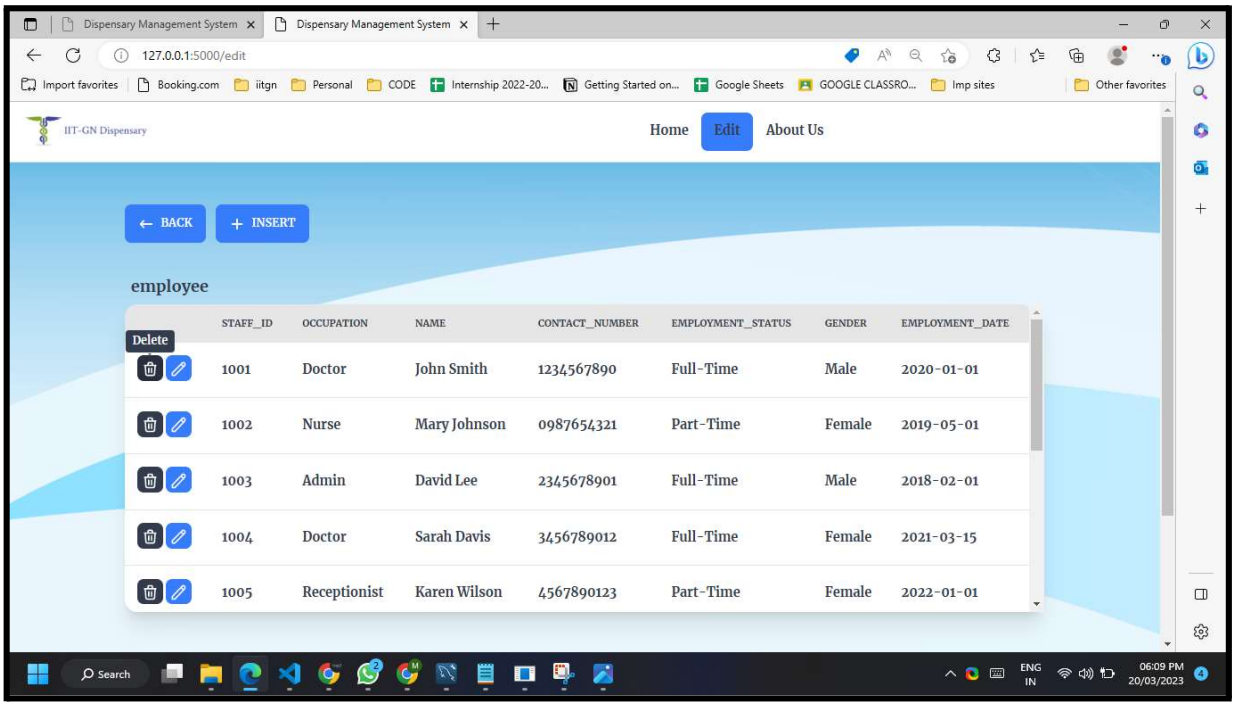


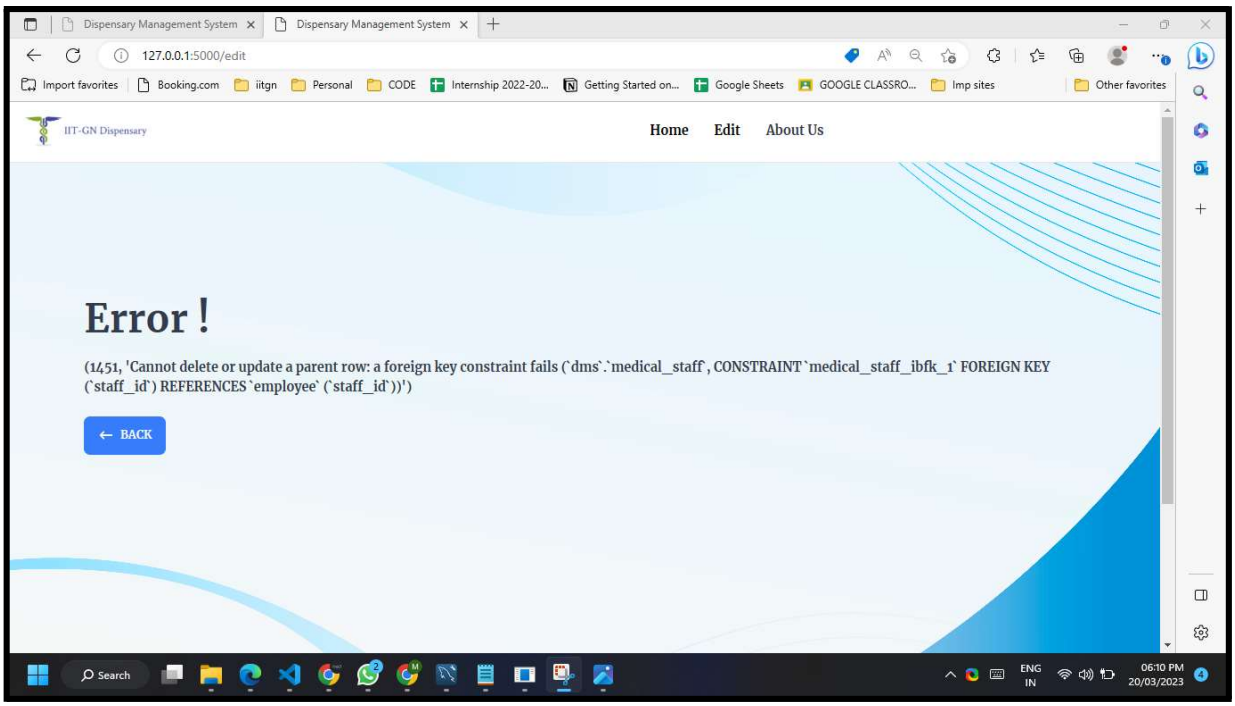
- Updating Entries in Employee Relation





- Deleting Entries in Employee Relation





Note: Because of a foreign key constraint error on insertion/deletion of a record, we faced a few issues with implementing deletion and insertion on some tables. We plan to resolve this issue in further iterations of the webpage.

Work Distribution

Team Member	Contribution
1. Aishwarya Omar	User authentication frontend, Connected authentication frontend with backend, Integrated webapp, Report Documentation
2. Akshat Shrivastava	Frontend debugging, Backend Development, Report documentation.
3. Amaan Ansari	Integrated frontend and developed the UI, Report compilation, Frontend Development.
4. Mahesh Dange	Frontend designing, Frontend Development, Backend Development, integrated frontend and backend, Made Readme File, Report documentaion
5. Dhakad Bhagat Singh	Backend debugging, Frontend Designing, Report documentation, Backend development.
6. Govardhan Ingale	Connected database backend, Frontend designing, report compilation
7. Ronak Hingonia	Report compilation
8. Ojas Washimkar	
9. Yash Adhiya	Backend Development, Frontend designing, Report documentation.
10. Yash Kokane	User authentication backend, Frontend designing, connected database backend for authentication, report compilation.
11. Shubham Kumar	Backend Development, Frontend debugging, Report compilation.