Devendra Ramkelawan

CIS 344

12/19/2023

Report on Hospital’s Portal

When starting the MySQL Workbench database, the process was smooth and there was only one roadblock, which was implementing the procedures, since it was something, I struggled with but after reading up on Delimiters it became easier to implement the functions.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

The MySQL database creation went smoothly; however python was an issue right from the start, an unknown error was found after trying to connect my Database to Portal Database python file, the MySQL server was uninstall and reinstall which fixed the issue, the I changed the third line in the portalServer.py which said hospitalDatabase, it was changed to PortalDatabase which allowed for the server to start. The first thing changed was the scheduleAppointment method, the function used for addPatient method was reused and changed to match the information in the MySQL database for appointments.

A close-up of a computer screen

Description automatically generated

The getAllPatients function was reused and manipulated to ensure the function for the viewAppointment and ViewPatients function, lastly the dischargePatient function uses the stored procedure from MySQL Workbench.

A computer code with colorful text

Description automatically generated

In the portalServer.py the challenges begun, to begin with, the comment for calling the database method for new patients was not seen, and took hours to notice after trouble shooting, after finding the comment and implementing the method, submitting information into addPatient, scheduling appointments, and discharging patients ran successfully.

A screenshot of a medical form

Description automatically generated

The do\_post function for schedulingAppointments and discharging patients was implemented with the code used from addPatient.

A screenshot of a computer program

Description automatically generated

The do\_get functions worked without any issues after implementation, the scheduleAppointment, viewAppointments, dischagepatient and viewPatients was from the already implemented code, and changed to match the Database.

A screenshot of a computer program

Description automatically generated

When doing a final test run on Hospital’s Portal Database, the functions were evaluated, and all worked as intended. The home page displays all the patients in the database including the ones added from the Add Patient Form.

A screenshot of a medical appointment

Description automatically generated

A screenshot of a medical form

Description automatically generated

The Schedule and View Appointments features worked as intended as well, the form submitted the information inputted into the view appointments table.

A screenshot of a medical appointment

Description automatically generated

A screenshot of a medical appointment

Description automatically generated

For the python files, it was hard at the start, after doing quick refreshers the codes became easier to edit and implement the functions, which was the most time consuming process in the creating the portal, it was through trial and error with multiple indentation errors, expect error and typos that all the mistakes were removed and redundancies in code were erased, with one being two separate code in the do\_get function for scheduling appointments the second version was supposed to be viewAppointments. In the end the portal was up and running and the functions implemented worked as intended.