Chrysalis high

KADUGODI

MINDCOLOGY

BOOKING SYSTEM FOR THERAPY AND MENTAL HEALTH CONSULTATIONS

BY ARYA HARIHARAN, SHAMILI GANDE AND RAJ AMMU

ABSTRACT

AIM AND OBJECTIVE

The intention behind the computer science project is to create a program-driven system that is unique and one that may be beneficial to the common people when thought over on a large scale. In this era of the pandemic, access to therapy and aid for mental health issues has been limited and the main intention of this project is to help curb that problem.

This project is essentially a booking system by which a user can book sessions with associated therapy institutions and connect with support groups. The information available to the user will include doctor's qualification, the price for consultations, specialization and so on. A price will be calculated based on the user's preferences (senior/junior aid, online/offline and so on), and will be displayed to the user as the final output.

● WHY DID WE CHOOSE THIS PROJECT?

Pre-existing, similar ideas and programs include Practo, TataHealth and so on. All these projects involve the idea of making healthcare more accessible and widespread. However, there is a serious lack of accessibility to proper mental health care resources. Many are often apprehensive to ask for help and through this idea, we hope to influence and eradicate this worry by showcasing that help is always there for you when you need it and there are many others who are going through this with you. We hope to normalise this practice and help others prioritise their mental health.

● OVERVIEW OF PROJECT COMPONENTS

1. CHOICE OF VARIETY OF DOCTORS AND INSTITUTIONS

This software allows one to choose between a variety of doctors and institutions, all specializing in mental health aid. They range from general therapists to specialized doctors. A user may also choose from several different associated workshops, if they feel they'd rather prefer a group interaction with other people like them. The user is automatically added to the mailing list on the system for receiving future updates. However, a user may choose to unsubscribe any time. Our interactive software makes navigation extremely simple.

2. SIMPLE CALCULATION OF FINAL BILL AMOUNT

Once the user has finalised which doctor/institution/workshop they would like to consult with, they shall be directed to a confirmation page where they can choose the date and the doctor. Based on the individual price for each doctor, a simple calculation incorporating GST and convenience fee is done to display the final price to the user in the invoice.

SYSTEM SOFTWARE AND HARDWARE

SOFTWARE USED TO RUN THE PROJECT-

1) Windows 10.0 and iOS

2) Python IDLE and Visual Studio Code

3) Front end

- Tkinter
- Python

4). Back end

- SQL (Structured Query Language)

HARDWARE USED TO RUN THE PROJECT-

1) 1.10 GHz Dual-Core Processor

2) 4/8 GB RAM

3) 250 GB HDD

FEATURES OF OUR PROJECT

The main idea behind this project is to provide a user with the choice of choosing from a variety of different consultation methods and doctors so as to increase comfort and to cater to the user's needs. For this purpose, we have categorised the information displayed in the project into four broad categories - welcome pages, general doctors, specialist doctors and workshops. These are the four main segments or 'features' of our project, in which the code has been written in a single .py file. A brief idea of each segment is given below -

WELCOME PAGES

These are the first set of pages that are a part of our application. This particular set includes the home page, login page, register page, welcome page and available services page. The first page of our application is the home page which displays the name, motto and login/register buttons. On clicking on these buttons, the user is directed towards the login and register page respectively. This is the segment of the project where there is a core utilisation of SQL. If it is a new user, the data from the register page (name, username, password and email) is split and stored in two tables - one for user info (name, username and password) and the other for mailing list info (email, name). The information from these tables are then accessed when an existing user tries to login by entering their username and password. Only if both the username and password are correct and matching are they allowed to proceed. Else, an error message is displayed. Once the user has successfully registered or logged in, they are taken to the 'Welcome Page' where a brief description of the application is written. This page also has an 'About Us' button at the bottom which takes the user to another page where we provide a brief introduction about ourselves. Then, users can navigate to the 'Available Services Page' where we display all the services we display - consultations with general doctors, specialists and workshops.

GENERAL DOCTORS PAGES

These pages contain information regarding the general doctors that are available for contact through our program. These doctors are for those users who feel like their issue is not specific or localised on some feeling and would just like to have a general talk. The main segmentation begins from the page which we call the 'Available Services Page' where a user can navigate to the list of general doctors by clicking a button. On the general doctors page, we have listed out the name, qualification, associated institution and price of the doctor. One can click on the doctor of their choice and navigate to a 'Confirmation Page' where they must reconfirm their doctors via radio buttons and choose a date from a calendar implemented using the tkCalender module. Once done, they are directed to a final invoice page and then to a logout page where they can choose to exit the application. Each button in this page, as well on most other pages, is actually an image converted to a button and then added on to the main page to give the user a broader area on which to click to take them to their desired page. We have also enabled a feature where hovering over an image-button will cause its borderwidth to change from 0 to 1, acting as further indication that the user is over a button and is free to click on it. On the invoice page, the final price after calculations will also be shown. [PRICE+GST+CONVENIENCE FEE]

SPECIALISTS PAGES

These pages contain information regarding the specialist doctors that are available for contact through our program. These doctors are for those users who feel like their issue is specific and would like to contact a doctor whose expertise lies in helping curb that specific issue. There are a total of eight doctors displayed - two doctors for each specialisation i.e. anxiety, depression, post-trauma or addiction. The main segmentation begins from the page which we call the ‘Available Services Page’ where a user can navigate to the list of specialist doctors by clicking a button. On the specialists page, we have listed out the name, qualification, associated institution and price of the doctor. One can click on the doctor of their choice and navigate to a ‘Confirmation Page’ where they must reconfirm their doctors via radio buttons and choose a date from a calendar implemented using the tkCalender module. Once done, they are directed to a final invoice page and then to a logout page where they can choose to exit the application. Each button in this page, like the other pages, is actually an image converted to a button and then added on to the main page to give the user a broader area on which to click to take them to their desired page. We have also enabled a feature where hovering over an image-button will cause its borderwidth to change from 0 to 1, acting as further indication that the user is over a button and is free to click on it. On the invoice page, the final price after calculations will also be shown. [PRICE+GST+CONVENIENCE FEE]

WORKSHOP PAGES

These pages contain information regarding the workshops/support groups that are available for contact through our program. These are for those users who feel like their issue is helped better by identifying people like themselves around them and moving forward together towards healing as a group. There are a total of eight workshops displayed - two workshops for each specialisation i.e. anxiety, depression, post-trauma or addiction. The main segmentation begins from the page which we call the ‘Available Services Page’ where a user can navigate to the list of workshops by clicking a button. On the workshops page, we have listed out the name, brief description and associated institution of the workshop. One can click on the workshop of their choice and navigate to a ‘Confirmation Page’ where they must reconfirm their workshop via radio buttons and choose a date from a calendar implemented using the tkCalender module. Once done, they are directed to a final invoice page and then to a logout page where they can choose to exit the application. Each button in this page, like the other pages, is actually an image converted to a button and then added on to the main page to give the user a broader area on which to click to take them to their desired page. We have also enabled a feature where hovering over an image-button will cause its borderwidth to change from 0 to 1, acting as further indication that the user is over a button and is free to click on it. On the invoice page, the final price after calculations will also be shown.

Apart from these groups, we also have some other stand alone pages like the ‘Terms and Conditions Page’ and ‘Unsubscribe From Mailing List Page’ which can be accessed by clicking on the user logo displayed on every single page on the top-right corner. The ‘Unsubscribe From Mailing List Page’ also accesses SQL to delete the mailing info of a user who no longer wishes to receive emails from Mindcology.

INBUILT LIBRARIES AND PACKAGES USED

TKINTER: Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. GUI is nothing but a desktop app that provides you with an interface that helps you to interact with the computers and enriches your experience of giving a command to your code.

METHODS USED:

a. Geometry(): This method is used to set the dimensions of the Tkinter window as well as it is used to set the position of the main window on the user's desktop.

b. Frame(): It works like a container, which is responsible for arranging the position of other widgets.

c. Grid(): This geometry manager organizes widgets in a table-like structure of the parent widget.

d. Pack(): This geometry manager organizes widgets in blocks before placing them in the parent widget.

e. place(): This geometry manager organizes widgets by positioning them in a specific position, in the parent widget.

f. Label(): This widget implements a display box where you can place text or images. The text displayed by this widget can be updated at any time you want.

g. Button(): The Button widget is used to add Buttons in Python. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.

h. Calendar(): This widget implements a calendar on a desired page from which one can extract the chosen date using functions.

i. Canvas(): This method is used to create a canvas which is a rectangular area intended for drawing pictures or other complex layouts. We can place graphics, text, widgets or frames in a canvas.

j. Label(): This widget implements a display box where you can place text or images. The text displayed by this widget can be updated at any point of time.

k. Button(): This widget is used to add Button in a Python application. These buttons. These buttons can display text or images to convey their purpose. We can attach a function or method to a button which is executed once the button is clicked.

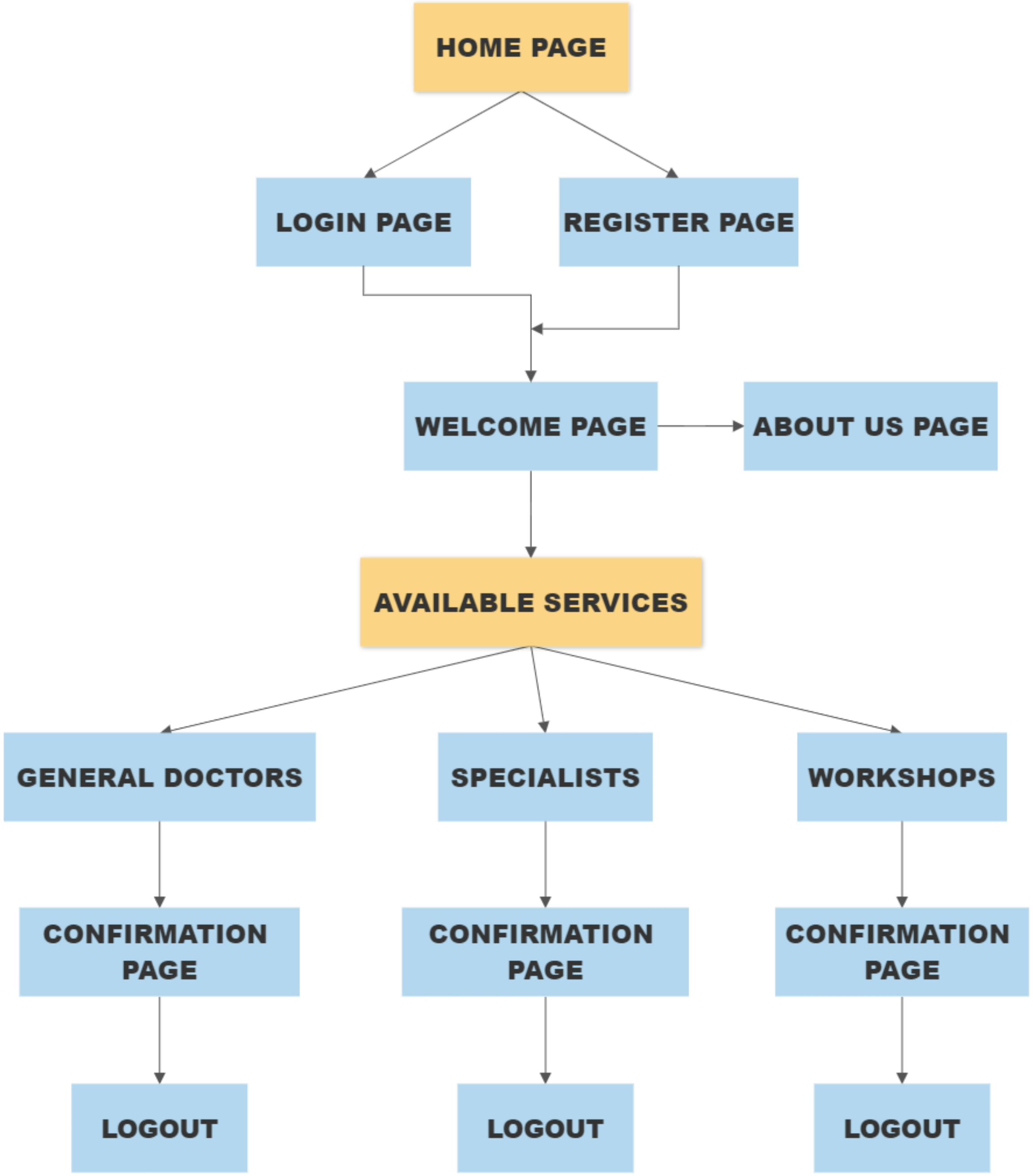
l. Radiobutton(): This widget implements a multiple choice button, which is a way to offer many possible selections to the user but lets them choose only one of them.

m. Entry(): This widget is used to accept single-line strings from the user.

n. Scrollbar(): This widget is used to scroll down content. It is possible to create both horizontally and vertically oriented scrollbars.

o. bind(): This function is used to deal with events. It is possible to bind functions and methods to an event and in turn, bind these functions and methods to any particular widget.

FLOWCHART OF PROJECT



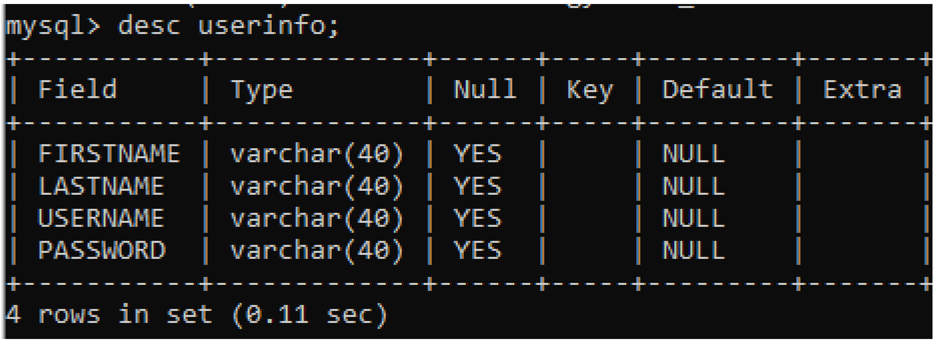
***NOTE - ALL THE LOGOUT PAGES AREN'T DIFFERENT. EACH CONFIRMATION PAGE LEADS TO THE SAME LOGOUT PAGE**

Also, apart from the flow given above, the user can also navigate to the 'Terms and Conditions Page' or the 'Unsubscribe from Mailing List Page'.

SQL TABLES

- **USERINFO TABLE TO STORE FIRSTNAME, LAST NAME, USERNAME AND PASSWORD**

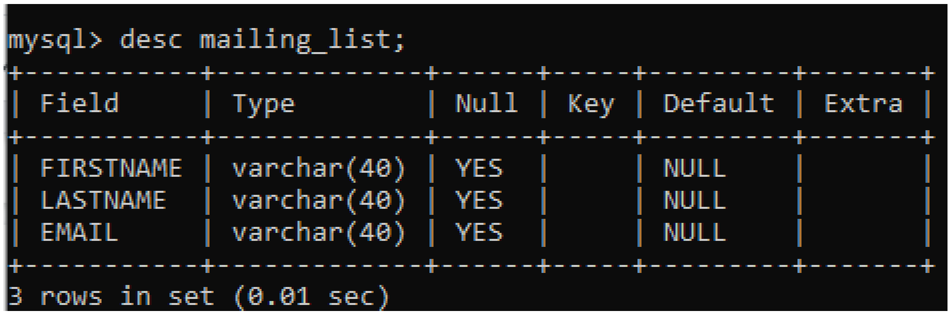
  Info is inserted into it from the registration page. The inserted info is then accessed for comparison and authentication in the login page.



```
mysql> desc userinfo;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| FIRSTNAME | varchar(40) | YES  |     | NULL    |       |
| LASTNAME  | varchar(40) | YES  |     | NULL    |       |
| USERNAME  | varchar(40) | YES  |     | NULL    |       |
| PASSWORD  | varchar(40) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.11 sec)
```

- **MAILING_LIST TABLE TO STORE FIRSTNAME, LAST NAME AND EMAIL**

  A registered user's email and name is added into this table by default. The idea behind this is that a user who is registered is subject to receive emails regarding updates from Mindcology (if fully functional) by extracting their info from this table. Moreover, if a user wishes to not receive emails anymore, they can 'unsubscribe' upon which their record will be deleted from this table. However, their userinfo will still stay in the previous table.



```
mysql> desc mailing_list;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| FIRSTNAME | varchar(40) | YES  |     | NULL    |       |
| LASTNAME  | varchar(40) | YES  |     | NULL    |       |
| EMAIL     | varchar(40) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

***NOTE -** We have not included any primary or foreign keys in the tables despite having two similar columns since each table has a different functionality and one table may have to be modified whilst leaving the other table intact.