TASK 3

For creation pages use **Bootstrap5 and DataTables (**https://datatables.net/**)**

Expand database qr_restaurant with admins tables. Insert three admins.

Create login system for admin part. In this part admins and waiters can log in.

Create New submenu Orders with option *All orders* (all_orders.php)

Create *qr_codes.php* file which has form with fields:

table number – text field

send – submit button

cancel – reset button

Form sends data with POST method to *create_qr_codes.php* file which check if data is entered. If not redirect user and write proper message. If everything is ok, create token (40 chars), create qr code and insert data in *restaurant_tables* and *qr_codes* tables. Images with QR codes put in *codes* folder. For creating token you can use **sha1** algorithm to create hash from pattern. Pattern is: *current timestamp-random number between 100 and 1000-day in week-salt.*

Salt is variable with value of string type. It will be defined in *config.php* file. Server is a web address of server. It will be defined in *config.php* file as constant.

Content of each QR code is URL in format:

`http://server/scan.php?t=token`

Define the file name as you wish but it must be unique.

You must insert the content of QR code in *url* field in *qr_codes* table.

Create *waiters.php* file which has form with fields:

name – text field

password – password field (minimum 8 chars)

send – submit button

cancel – reset button

Form sends data with POST method to *create_waiters.php* file which check if data is entered. If not redirect user and write proper message. If everything is ok, insert data in *waiters* table. For hashing password use BCRYPT.

Create *menu_categories.php* file which has form with fields:

name – text field

send – submit button

cancel – reset button

Form sends data with POST method to *create_menu_categories.php* file which check if data is entered. If not redirect user and write proper message. If everything is ok, insert data in *menu_categories* table. You must not have duplicate names in the table. Before entering the data, check whether the data already exists.

Create *menus.php* file which has form with fields:

name – text field

category (select list with values from menu_categories table)

description – textarea (optional)

image – file field (optional and accept only JPG format)

send – submit button

cancel – reset button

Form sends data with POST method to *create_menu.php* file which check if required data is entered. If not redirect user and write proper message. If everything is ok, insert data in *menu_categories* table. You must not have duplicate names in the table. Before entering the data, check whether the data already exists. For image upload you need to check JPG format with EXIF. If it is ok rename image in following way: current timestamp-id_menu_category-random number between 10 and 100.jpg. Put images in *menus* folder.

Create *prices.php* file which has form with fields:

menu (select list with values from menus table)

size – text field

price – text field

send – submit button

cancel – reset button

Form sends data with POST method to *create_price.php* file which check if data is entered. If not redirect user and write proper message. If everything is ok, insert data in *prices* table.

If you want you can add size and price fields in *create_menus.php* file. The numbers of size and price fields should be added dynamically with pressing PLUS button. This is just an optional possibility.

Create *restaurant_tables_waiters.php* file which has form with fields:

restaurant table (select list with values from restaurant_tables table)

waiter (select list with values from waiters table)

send – submit button

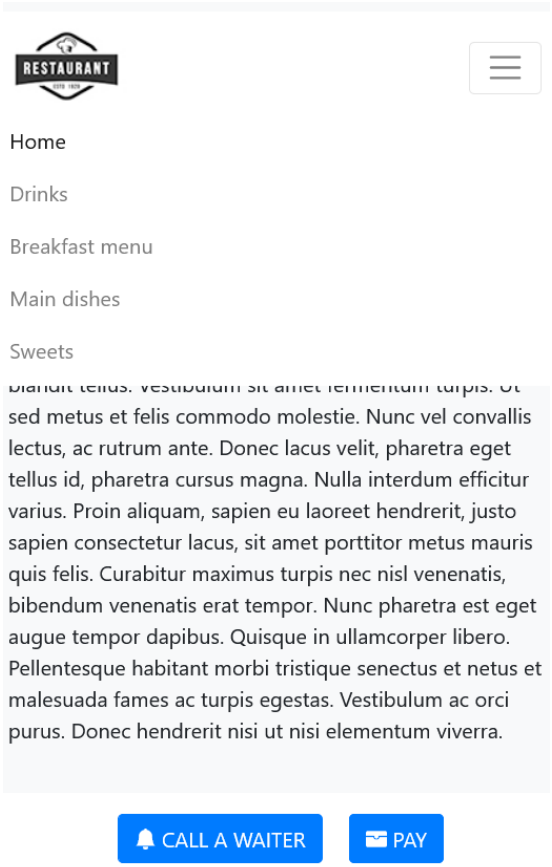
cancel – reset button

Form sends data with POST method to *create_restaurant_tables_waiters.php* file which check if data is selected. If not redirect user and write proper message. If everything is ok, insert data in *restaurant_tables_waiters* table.

In file *all_categories.php* finish the edit functionality in modal. After editing reload the table.

Create files *all_waiters.php*, *all_qr_codes.php*, *all_waiters_tables.php*, *all_menus.php*, *all_statistics.php* based on *all_categories.php* file. Implement *edit* and *delete* options. ***Delete is important!***

Create *scan.php* file for client. It should be outside *admin* folder. *Scan.php* file detects device, inserts data into *statistics* table and shows the categories and menus for that category. Create folders includes and other necessary folders. Create *functions.php* file. For getting data create functions *getCategoryData*, *getCategories*.



Create api folder inside root folder. In this folder create following endpoints:

getStatistics.php – which outputs json with data from statistics table

getMenuCategories.php – which outputs json with data from menu_categories table

getMenu.php – which outputs json with data from menus and menu_categories tables (JOIN tables)

getOrders.php – which outputs json widt data from orders and order_items tables (JOIN tables)