



L-Università ta' Malta
Faculty of Information &
Communication Technology

Department of
Computer Information
Systems

Principles of Web Application Architecture

Assignment: fork() Website

Evangeline Azzopardi* (33600H), Kian Parnis* (107601L)
*B.Sc. IT (Hons) Artificial Intelligence

Study-unit: Principles of Web Application Architecture
Code: CIS1054
Lecturer: Dr Chris Porter

CIS 1054: Principles of Web Application Architecture Assignment Documentation

Kian Parnis (107601L) and Evangeline Azzopardi (33600H)
B.Sc IT Artificial Intelligence

8th June 2021

Contents

1 Plagiarism Declaration	4
2 Part 1	5
2.1 Question 1	5
2.2 Question 2	5
2.3 Question 3	5
2.4 Question 4	5
2.5 Question 5	6
3 Part 2	7
3.1 Design	7
3.1.1 A General Overview	7
3.1.2 Structure of Data	8
3.1.3 Entity Relationships	10
3.1.4 Flow of Data	11
3.2 Set Up	12
3.2.1 Server to run PHP	12
3.2.2 Database	12
3.2.3 Email	12
3.3 Testing	13
3.3.1 Testing the Navigation Bar	13
3.3.2 Testing Each Page Link	15

3.3.3	Testing the Contact form	18
3.3.4	Testing the Login System and its Functions	24
3.3.5	Testing the Menu	29
3.3.6	Testing Adding to Favorites List and Mailing the List	31
3.4	Further Implementation and Improvements	35
4	Group Responsibilities	36
4.1	Individual Responsibilities	36
4.2	Group Dynamics	36
4.3	Meeting Attendance Log	36
5	Bibliography	37

1 Plagiarism Declaration

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines" (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

We, the undersigned, declare that the assignment submitted is our work, except where acknowledged and referenced.

We understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected and will be given zero marks.

(N. B. If the assignment is meant to be submitted anonymously, please sign this form and submit it to the Departmental Officer separately from the assignment).

Evangeline Azzopardi
Student Name

E Azzopardi
Signature

Kian Parnis
Student Name

Kian
Signature

CIS1054
Course Code Principles of Web Application Architecture Assignment
Title of work submitted

8th June 2021
Date

2 Part 1

GitHub Repository Link: <https://github.com/EvaWebDev/CIS1054-Part1.git>
WyvernCore: Kian Parnis
EvaWebDev: Evangeline Azzopardi

2.1 Question 1

The following outlines the process followed in order to set up a HTTP server to support PHP scripts, for this implementation:

The first step of the assignment setup involved installing a local Apache server on each participant's machine. This server includes the XAMPP package in which Apache was used to host the website on the local server, MySQL to set up the database for the required data storage and Mercury to utilise the email functionality. A PHP interpreter is included in the XAMPP Package and was used throughout the implementation, hence, no external software was required for the server to execute server-side scripts. XAMPP was installed from <https://www.apachefriends.org/index.html>. All the relevant files for the website implementation were stored in the XAMPP htdocs folder, in order to be executed via the Apache server. The Apache server is run on the ports 80 and 443, whilst MySQL runs on the port 3306.

2.2 Question 2

Implementing serverdt.php involved utilising the built-in PHP functions *DateTime()* and *date_default_timezone_get()*. The date, time and time zone are dynamic through the aforementioned functions.

Server's Timezone: Europe/Berlin
Current Time: 13:34:21
Current Date: 24-05-21

Figure 1: The output for serverdt.php

2.3 Question 3

Implementing lastvisit.php involved utilising session variables and creating a function called *since_time* which returns a string of the time since the last first visit of a website from an array *time*. *lastvisit* also checks if the session is already set to ensure that it doesn't keep resetting the timer every time.

You first used this site 4 seconds ago

Figure 2: The output for lastvisit.php

2.4 Question 4

Implementing processrequest.php and questionthree.php required the use of *POST* and *GET* session variables in order to store the data input into the HTML form. questionthree.php consists of the code for the HTML form which accepts the user's input and processrequest.php outputs the data to the user.

2.5 Question 5

To access session variable `session_start()` is added to the beginning of each of the scripts used in question 4, as well as in the script for question 5. A session variable stored and then echoed back/displayed to the user. A characteristic of HyperText Transfer Protocol, HTTP, is that it does not maintain the state of the browser. Session variables are passed from one page to another on a given website in order for the browser to retain information about the user. This allows a developer to incorporate dynamic content which is dependent of the data stored in the session variables, if any.

The image shows a screenshot of a web application interface. At the top, there is a text input field containing the value "Kian Parnis". Below it is another text input field containing the value "20". Underneath these fields is a button labeled "Save Details". The entire form is contained within a light gray rectangular box. Below this box, the output of the session is displayed. It consists of two lines of text: "Kian Parnis" and "20", which correspond to the values entered in the input fields above.

Figure 3: The output for readsession.php

3 Part 2

GitHub Repository Link: <https://github.com/EvaWebDev/CIS1054-Part2.git>

WyvernCore: Kian Parnis

EvaWebDev: Evangeline Azzopardi

3.1 Design

The following diagrams will provide an overview of the structure for the fork() website, the interactions between the customers and the administrator with the website and the structure of the MySQL database setup with its respective data types and fields.

3.1.1 A General Overview

The following Use-Case-Diagram depicts a top-level view of the basic functions and interactions.

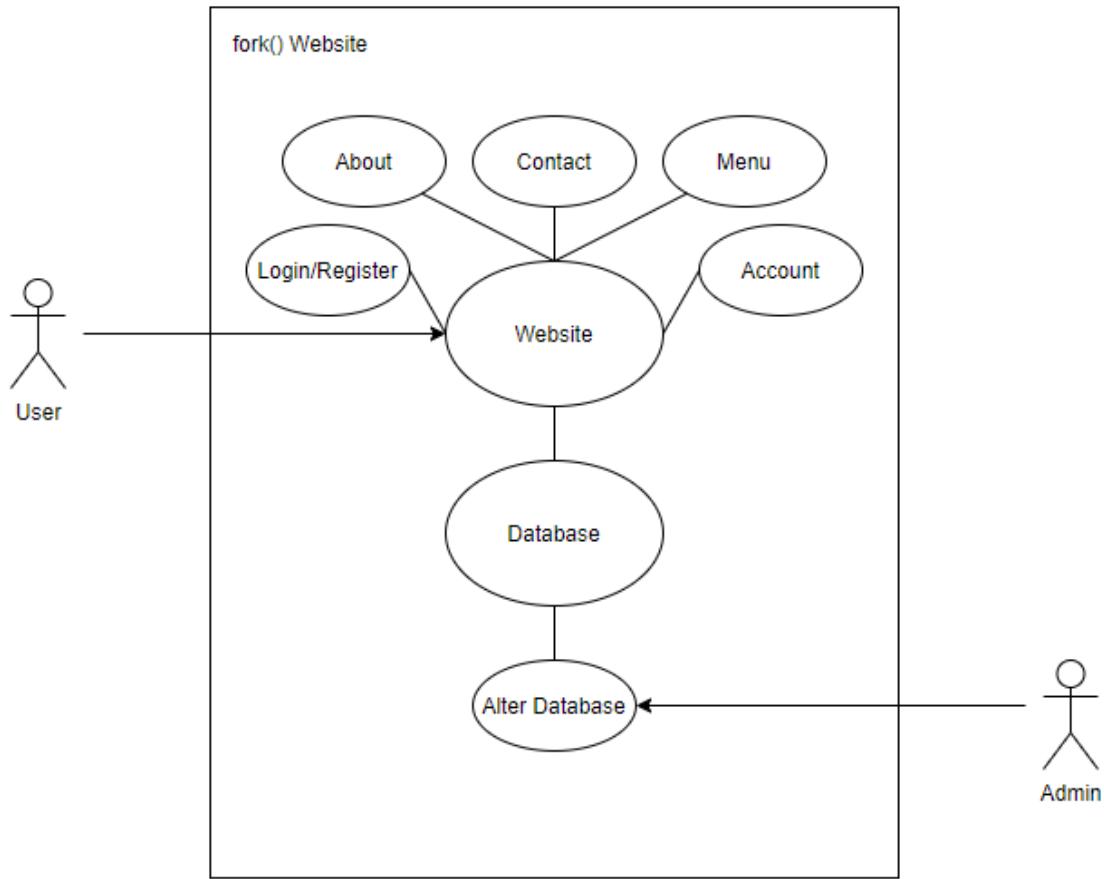


Figure 4: A Use-Case-Diagram for the fork() website.

3.1.2 Structure of Data

The following Data-Structure-Diagrams depicts how the data in the database was modelled prior to implementation and the result after implementation.

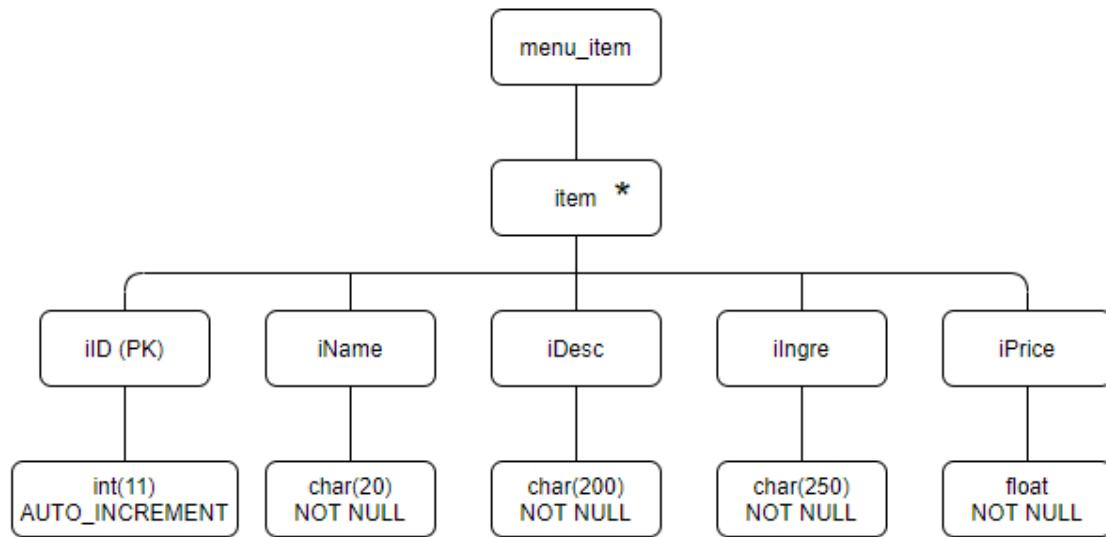


Figure 5: The DSD for the `menu_item` table.

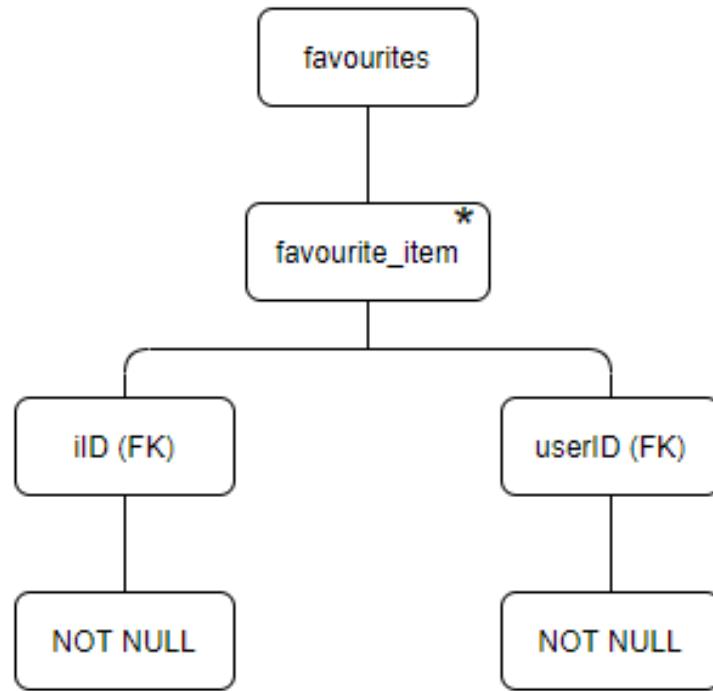


Figure 6: The DSD for the `favourites` table.

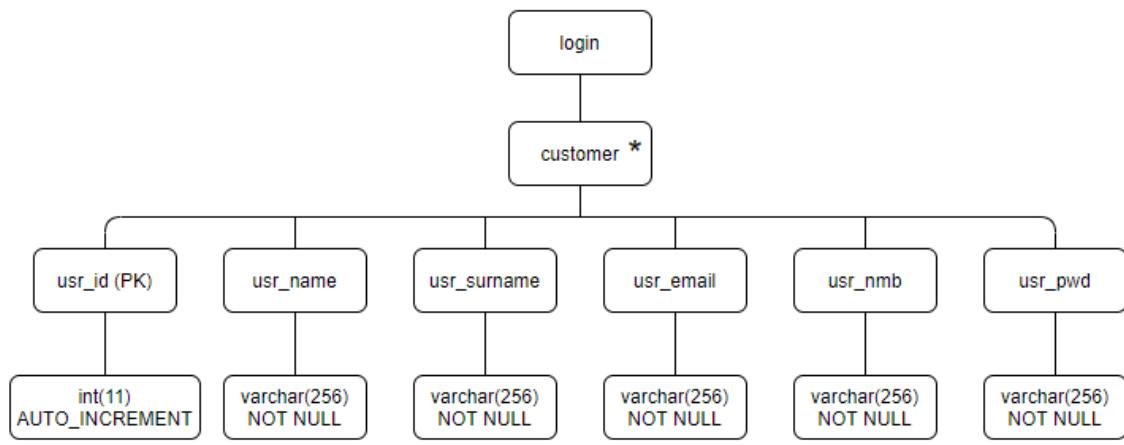


Figure 7: The DSD for the login table.

3.1.3 Entity Relationships

For the website's data to flow well and work efficiently, relationships between the database tables were required. The *login* table utilises the *usr_id* primary key to uniquely identify each registered user. The *menu_item* table utilises the *iID* primary key to uniquely identify each dish for sale. These two primary keys are used as foreign keys in the *favourites* table to identify which users have which dishes saved in their favourites list. The following two relationships occur in the database:

1. favourites to menu_item: 1 to N.
2. login to favourites: 1 to N.

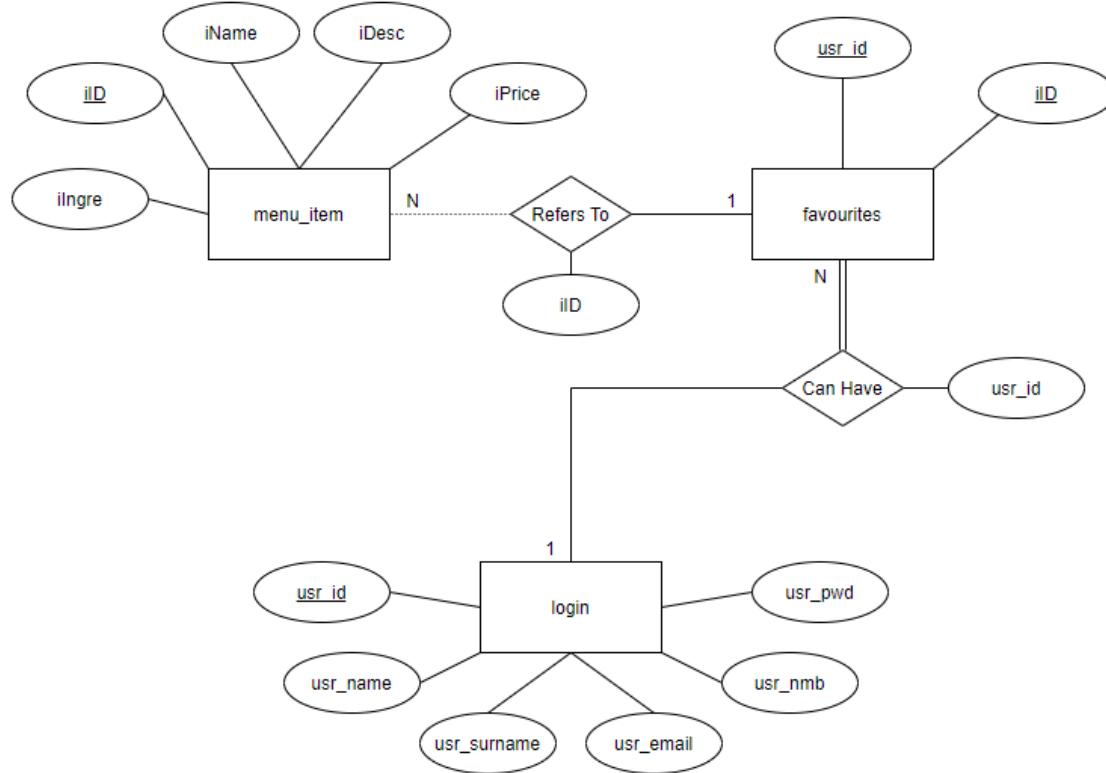


Figure 8: A detailed Entity Relationship Diagram depicting the database.

3.1.4 Flow of Data

The following Data-Flow-Diagrams outline the flow of data between all the functions implemented. The Context Diagram, a Level 0 DFD, depicts the data flow between the external entities and functions. A more detailed and internal depiction of the data flow between the entities and functions is shown through a Level 1 DFD below.

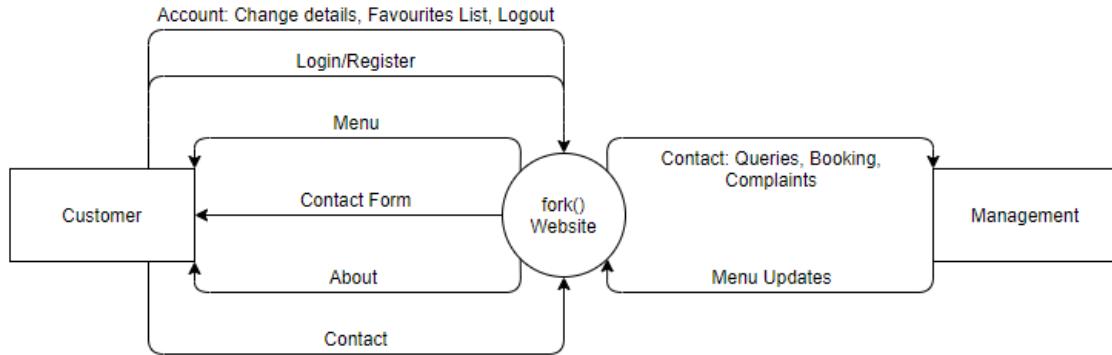


Figure 9: A Context Diagram, Level 0 DFD, depicting the external flow of the website.

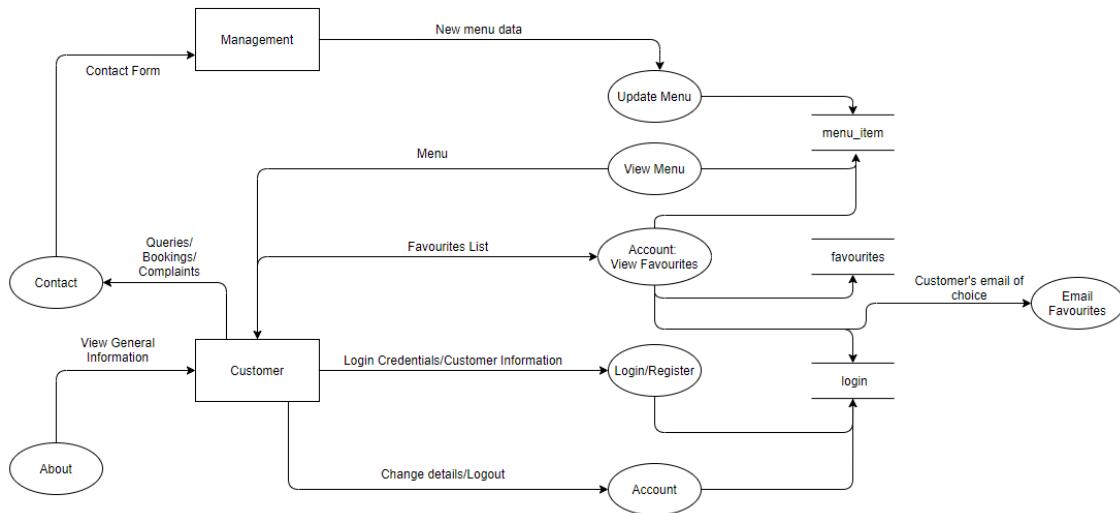


Figure 10: A Level 1 DFD depicting the internal data flow of the website.

3.2 Set Up

3.2.1 Server to run PHP

An Apache server was setup for the php scripts provided, this was set up using the XAMPP software (version 3.2.4). For the database to run, XAMPP's MySQL service was used. The folder housing the website was placed inside of the xampp/htdocs file and could be run locally by going into any browser typing 'http://localhost/CIS1054-Part2-main/' and finding the script 'About.php' which is the front page of the website.

3.2.2 Database

As stated above the XAMPP's MySQL service was used for the setup for the database, to access the database 'http://localhost/phpmyadmin/' will need to be typed in any browser. Create a new database from the lefthand side and call it 'phplogin' (Note: A different database name will result in the 'dbh.php' file not being able to connect to the database. After this, click on the empty database select 'import' at the top of the screen and choose 'phplogin.sql' found within the file 'DATABASE SETUP'. Alternatively within the same folder three text files have been provided in order and from the empty phplogin database the sql statements found within the textfile's can be put inside sql and run (must be run in order) for a similar result.

3.2.3 Email

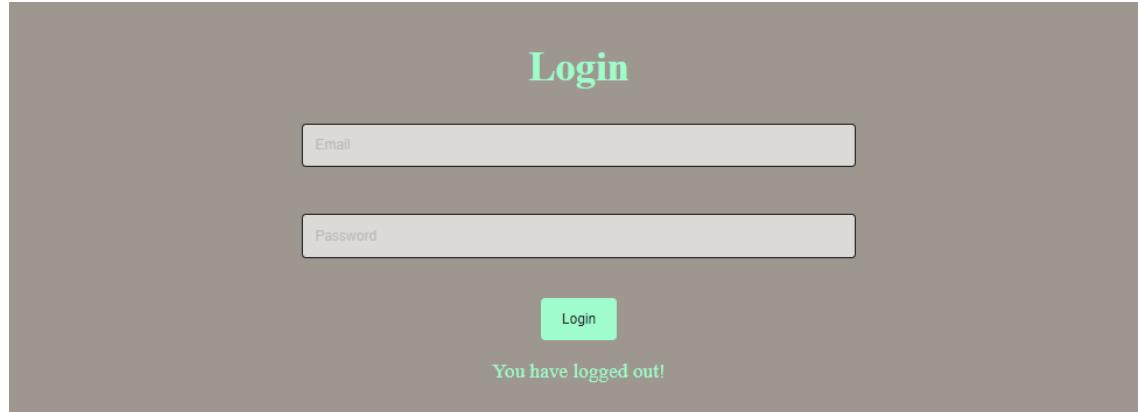
Firstly a unique gmail account was created solely for the use of mailing clients favorites list and booking. The username and password will be provided here for the use of testing Email: 'inforkrestaurant@gmail.com' Password: 'assignment'. For the actual mailing setup the article [1] was used for as a guide for the configuration. The step's of this configuration are as follows:

1. Go to \xampp\php and open php.ini
2. Using 'ctrl + f' find 'mail function'.
3. Change the following (all lines edited need to be uncommentated):
 - (a) SMTP=smtp.gmail.com
 - (b) smtp_port=587
 - (c) sendmail_from = inforkrestaurant@gmail.com
 - (d) sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"
4. Next find the file \xampp\send mail and open sendmail.ini
5. Using 'ctrl + f' find 'sendmail'.
6. Change the following (all lines edited need to be uncommentated):
 - (a) smtp_server=smtp.gmail.com
 - (b) smtp_port=587
 - (c) error_logfile=error.log
 - (d) debug_logfile=debug.log
 - (e) auth_username=inforkrestaurant@gmail.com
 - (f) auth_password=assignment
 - (g) force_sender=inforkrestaurant@gmail.com
7. Finally restart Apache server

3.3 Testing

3.3.1 Testing the Navigation Bar

When the user is logged out:



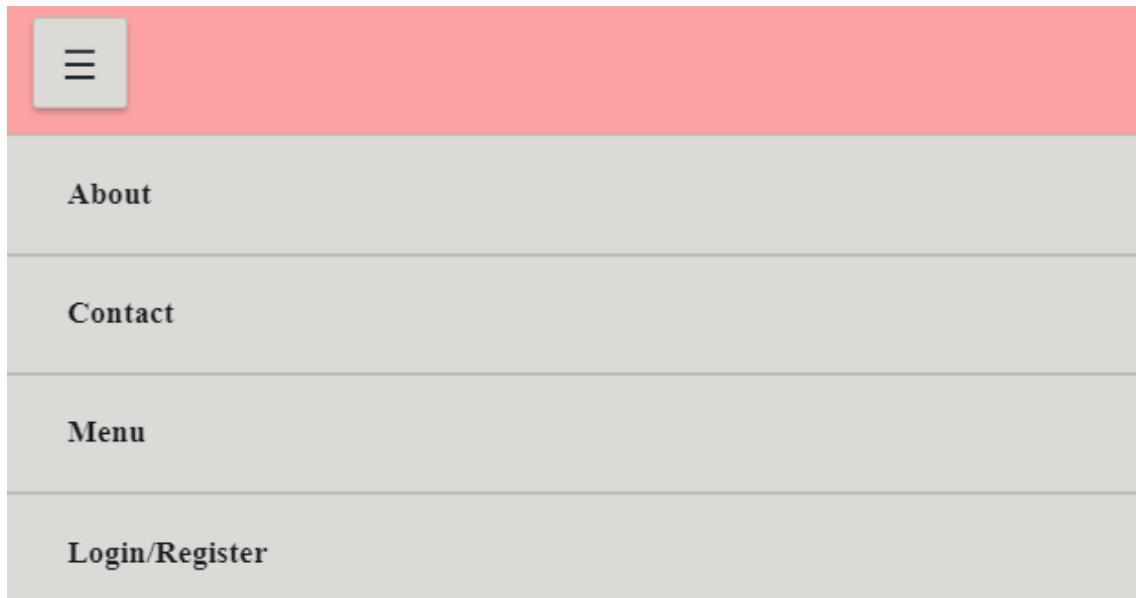
When the user logs in:



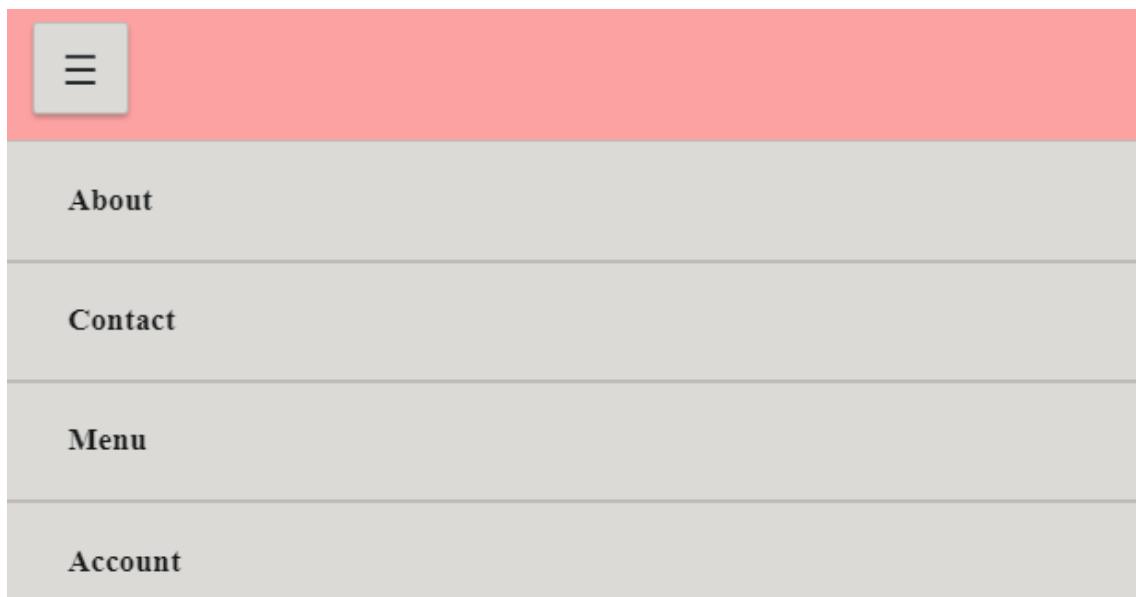
When the webpage is minimized:



When the user is logged out and clicks the minimised bar:



When the user is logged in and clicks the minimised bar:



3.3.2 Testing Each Page Link

When the user clicks 'About':

The screenshot shows the 'About' page of the fork() website. At the top, there is a navigation bar with four items: 'About', 'Contact', 'Menu', and 'Login/Register'. Below the navigation bar, the page title 'fork()' is displayed in a large, dark button. The main content area is divided into two sections: 'LOCATION' and 'ABOUT'. The 'LOCATION' section contains the address '2142 Cedar Street Little Rock Arkansas 72210' and contact information 'tempemailaddress@temp.com' and '870-847-4945'. The 'ABOUT' section contains the following text:

fork() was founded by Kian Parnis and Evangeline Azzopardi

Start of mid-March in 2021 after being plagued with this assignment. Our name was inspired from one of our Operating Systems and Systems Programming lectures, specifically on processes, hence, *fork()*. It is far from the most creative restaurant name on the Internet, but you will have to bear with us. Now on to the serious stuff, all the goodies offered by our restaurant, *fork()*, can be consumed easily using a fork.

Though our menu is quite generic, courtesy of the idiot who created the menu's database table and also happens to be the writer of the content you are currently reading, our food will have you coming back for more. We believe that you, the consumer, should choose the ingredients, that we can offer, to create a customised mouth-watering meal. We also offer award-worthy, at least in our opinion, deserts.

Click 'Menu' at the top of the page to find out more.

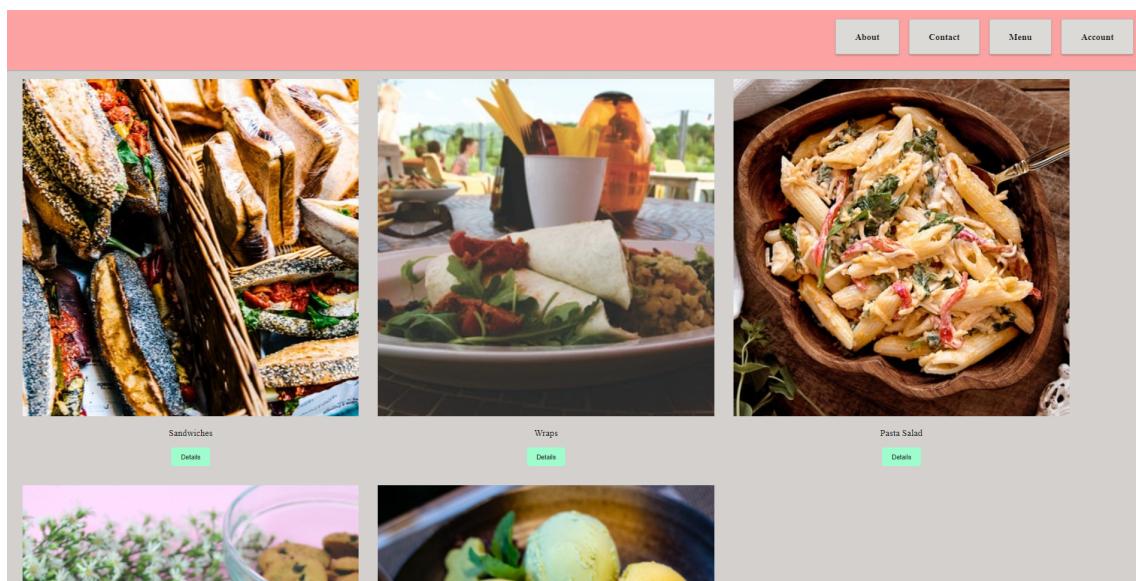
On the left side of the page, there is a map of a coastal city area with a red circle highlighting a specific location labeled 'Location of Park'.

When the user clicks 'Contact':



A screenshot of a contact form titled "Contact Us". The form includes fields for First Name, Last Name, Email, Contact Number, and Address/Description. There is also a large text area for a message and a "Send" button.

When the user clicks 'Menu':



When the user is logged in and clicks 'Account':

The screenshot shows a web application interface. At the top, there is a red header bar with four buttons: 'About', 'Contact', 'Menu', and 'Account'. Below the header, there are two main sections. The first section is titled 'Change Personal Details' and contains four input fields: 'First Name', 'Last Name', 'Email', and 'Contact Number', followed by a 'Save Details' button. The second section is titled 'Change Password' and contains two input fields: 'Enter Password' and 'New Password', also followed by a 'Save Details' button. The background of the main content area is grey.

When the user is logged out and clicks 'Login/Register':

The screenshot shows a web application interface. At the top, there is a red header bar with four buttons: 'About', 'Contact', 'Menu', and 'Login/Register'. Below the header, there are two main sections. The first section is titled 'Login' and contains two input fields: 'Email' and 'Password', followed by a 'Login' button. The second section is titled 'Register' and contains six input fields: 'First Name', 'Last Name', 'Email', 'Contact Number', 'Password', and 'Confirm Password', followed by a 'Sign up' button. The background of the main content area is grey.

3.3.3 Testing the Contact form

If the user correctly fills in the 'Contact' form:

Contact Us

Kian

Parnis

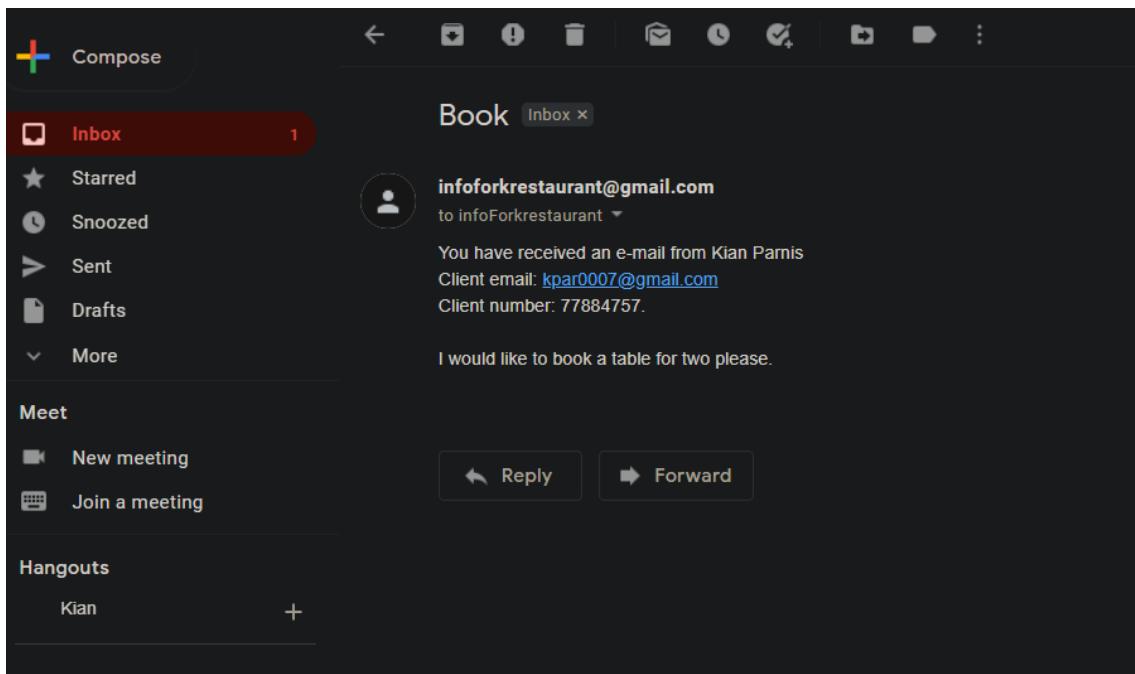
kpar0007@gmail.com

77884757

Book

I would like to book a table for two please.

Send



Contact Us

First Name

Last Name

Email

Contact Number

Book/Query/Complaint

Enter your message here....

Send

You have successfully contacted us!

The following is the validation checks done on the Contact Us form:

User leaves any entry empty:

The screenshot shows a 'Contact Us' form with five input fields: First Name, Last Name, Email, Contact Number, and Book/Query/Complaint. Below these is a large text area labeled 'Enter your message here....'. A red error message 'Not all the fields were filled!' is displayed at the bottom. A green 'Send' button is located below the message area.

Contact Us

First Name

Last Name

Email

Contact Number

Book/Query/Complaint

Enter your message here....

Send

Not all the fields were filled!

User enters an invalid number:

Contact Us

Kian

Parnis

kpar0007@gmail.com

778847572321321

Book

Testing

Send

Error invalid number!

The image shows a contact form titled "Contact Us". It consists of several input fields and a message area. The input fields contain the following text:

- Kian
- Parnis
- kpar0007@gmail.com
- 778847572321321
- Book

The message area contains the text "Testing". Below the form is a button labeled "Send". At the bottom of the page, there is an error message: "Error invalid number!".

User enters an invalid email:

Contact Us

Kian

Parnis

Not a valid email

77884757

Book

testing

Send

Error invalid email!

The form consists of the following fields:

- Name: Kian
- Surname: Parnis
- Email: Not a valid email
- Phone: 77884757
- Book
- testing

At the bottom, there is a "Send" button and an error message: "Error invalid email!".

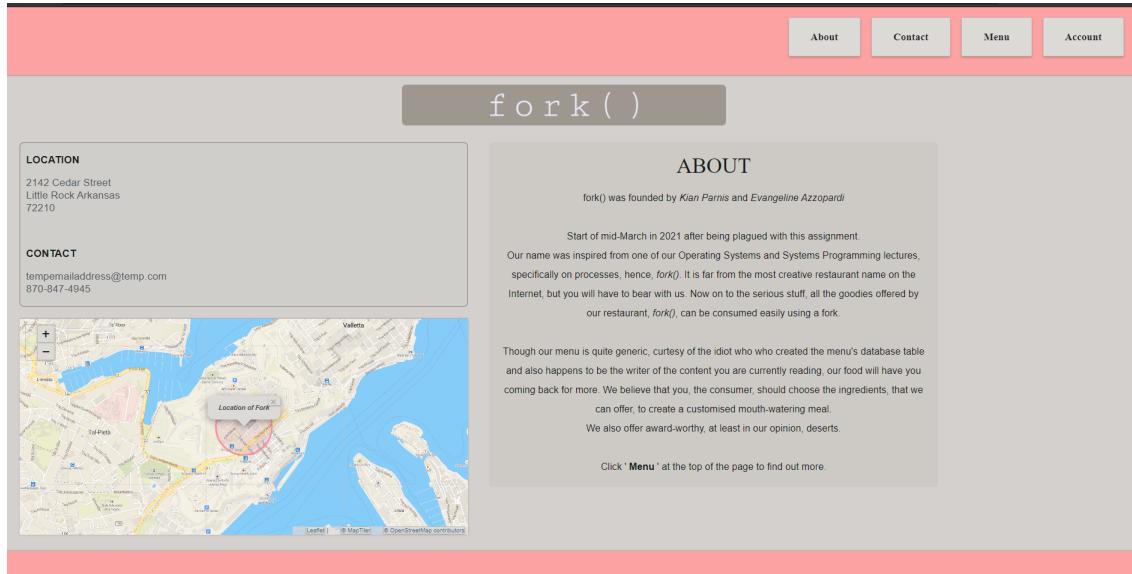
User enters an invalid name or surname:

Contact Us

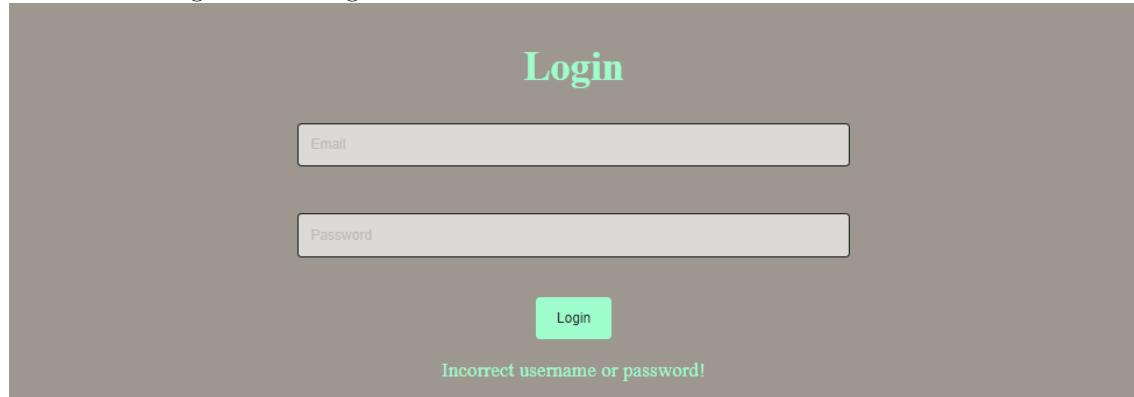
Name is unvalid!

3.3.4 Testing the Login System and its Functions

If the user successfully logs in, they are redirected to the about page and the navbar changes to logged in mode



On the other hand if the user either leaves a field empty or types an incorrect email or password they will get the the following error message:



Once the user is logged in, they can choose to log out by either totally exiting the session or by pressing 'Logout' and this redirects the user with the following message:

The image shows a 'Login' page with a light brown background. At the top center is the word 'Login' in a large, bold, black font. Below it are two input fields: the first is labeled 'Email' and the second is labeled 'Password'. At the bottom center is a green rectangular button labeled 'Login'. Below the 'Login' button, there is a small, centered message in a smaller black font: 'You have logged out!'

If the user successfully registers they are redirected to a thank you page stating the the credentials filled and this information is saved inside the database.

The image shows a 'Register' page with a light brown background. At the top center is the word 'Register' in a large, bold, black font. Below it are five input fields: the first contains 'Sharon', the second contains 'Parnis', the third contains 'sharon.parnis@gmail.com', the fourth contains '79058967', and the fifth contains two dots ('..'). At the bottom center is a green rectangular button labeled 'Sign up'.

The image shows a 'Thank You for registering' page with a light brown background. At the top center is the text 'Thank You for registering' in a large, bold, black font. Below it is a smaller text: 'Here is the information you have submitted:'. Underneath this, the submitted information is listed in a smaller black font: 'Name: Sharon', 'Surname: Parnis', 'Email: sharon.parnis@gmail.com', and 'Contact No: 79058967'.

			▼ usr_id	usr_name	usr_surname	usr_email	usr_mmb	usr_pwd
<input type="checkbox"/>				1	Kian	Paris	1234	77884757
<input type="checkbox"/>				2	Evangeline	Azzopardi	12345	79049526
<input type="checkbox"/>				3	Sharon	Parnis	77	sharon.parnis@gmail.com

Similar to the forms described above, the Register Form validates the name, email, number and checks if all the fields are filled. Additionally, the database checks if the number and email are already registered and if so the following errors are shown respectively:

Register

Kian

Parnis

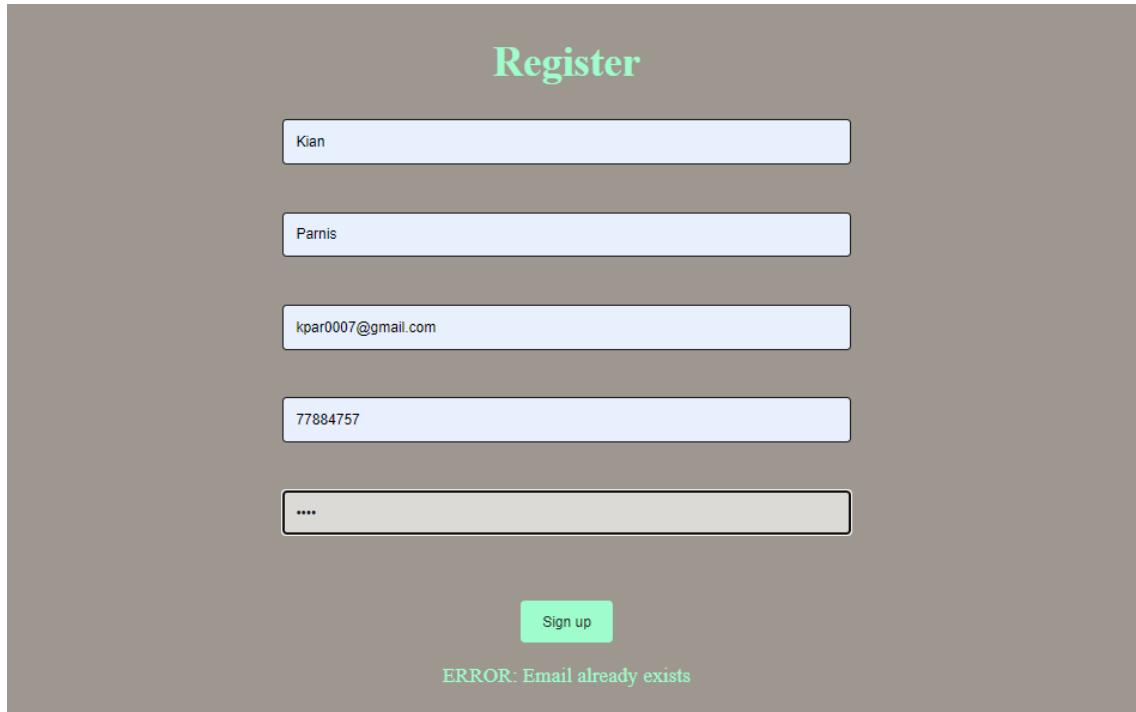
kpar0007@gmail.com

77884757

....

Sign up

ERROR: Email already exists



Register

Kian

Parnis

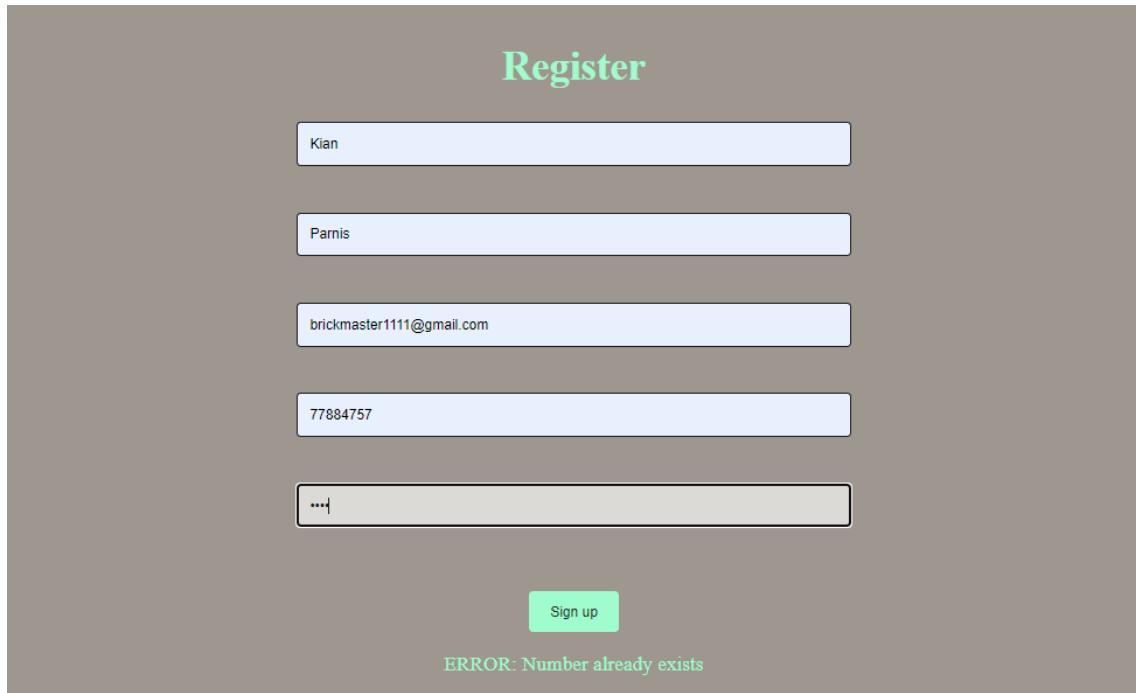
brickmaster1111@gmail.com

77884757

....

Sign up

ERROR: Number already exists



If the user chooses to change either the Name/Surname/Email/Number/password two respective forms are proved. In the first form 'Change Personal Details' if the user provides the valid credentials:

Change Personal Details

David

Cauchi

david.cauchi@gmail.com

77627829

Save Details

Change Personal Details

First Name

Last Name

Email

Contact Number

Save Details

You have successfully changed your Personal Details!

			▼ usr_id	usr_name	usr_surname	usr_email	usr_nmb	usr_pwd
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	David	Cauchi	david.cauchi@gmail.com	77627829
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Evangeline	Azzopardi	evieazzopardi@gmail.com	79045526
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	Sharon	Parmis	sharon.parmis@gmail.com	79059967

For validation, this works the exact same as the Register Form.

If the user changes the password successfully they will get the following message and the database will change the old password to the new one:

Change Password

Current Password

New Password

Save Details

Incorrect password!



Change Password

Current Password

New Password

Save Details

You have successfully changed your password!



			usr_id	usr_name	usr_surname	usr_email	usr_rmb	usr_pwd
<input type="checkbox"/>			1	David	Cauchi	david.cauchi@gmail.com	77827829	1234567
<input type="checkbox"/>			2	Evangeline	Azzopardi	evieazzopardi@gmail.com	79045826	12345
<input type="checkbox"/>			3	Sharon	Parris	sharon.parris@gmail.com	79058967	77

The validation done on 'change password' is checking if the password matches the accounts already existing one.

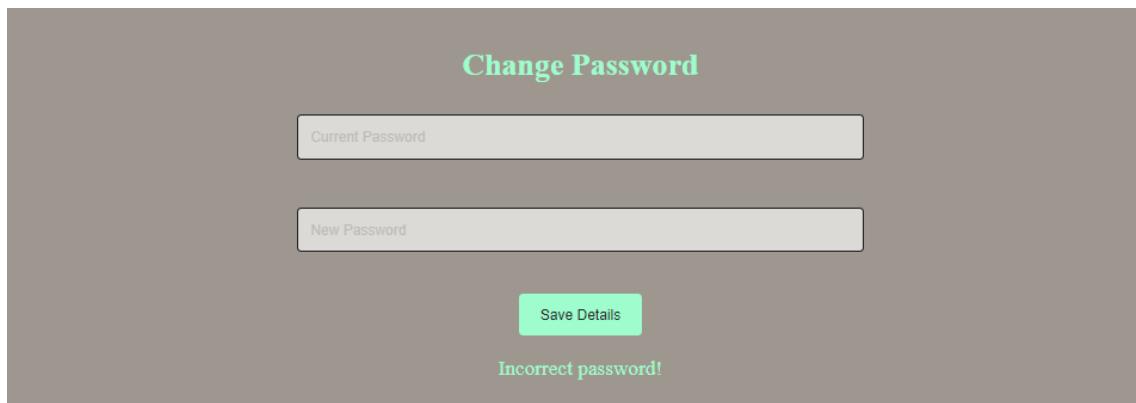
Change Password

Current Password

New Password

Save Details

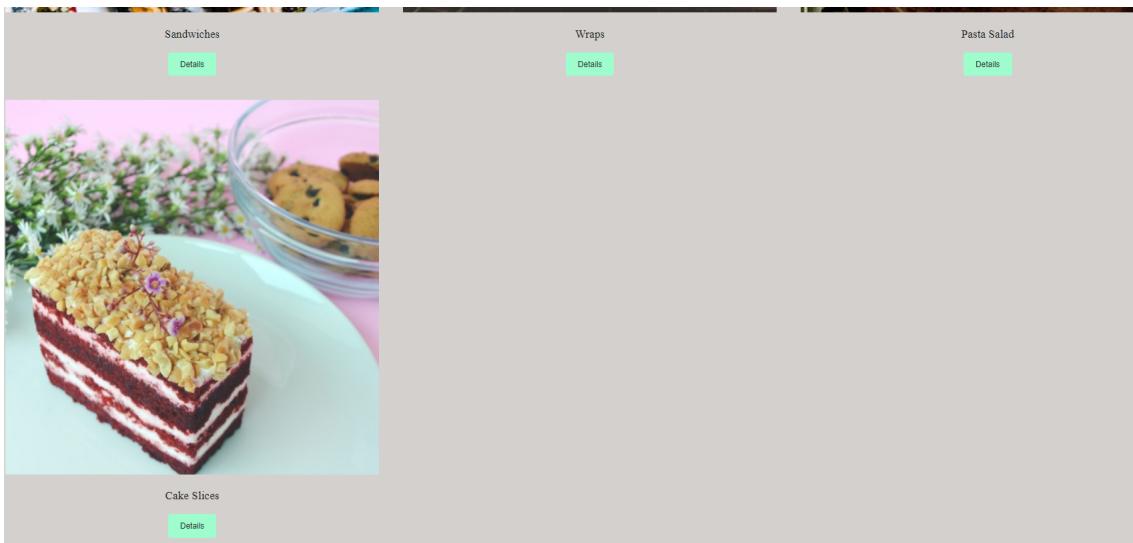
Incorrect password!



3.3.5 Testing the Menu

The Menu allows an employee to dynamically update the contents of the menu directly from the database. This is done by first placing the appropriate image inside the Images folder and naming it the same as the menu items iName in the database. The following is testing of inserting a new menu item:

Before adding a new item:



Database before adding a new item:

ID	Name	Description	Ingr	iPrice
1	Water	It is just water, there is nothing to explain or d...	100% Water	2
2	Iced Tea	Iced tea, peach or lemon flavoured	Water, fructose, citric acid, lemon juice concentr...	2.95
3	Tea and Coffee	White, Green and Black available. Black coffee...	Tea leaves, Coffee Beans, Plant-based and regular...	3.5
4	Sandwiches	Customise your sandwich with the choices below.	White or Oatmeal bread, Ham, Chicken, Bacon, Tuna...	3.95
5	Wraps	Customise your wrap with the choices below.	White or Oatmeal wrap, Ham, Chicken, Bacon, Tuna, ...	4.95
6	Pasta Salad	Customise your pasta salad with the choices below.	White or Brown pasta, Ham, Chicken, Bacon, Tuna, A...	5.95
7	Cake Slices	Victoria Sponge, Chocolate, Red Velvet, Coffee and...	Flour, Eggs, Water, Milk, Flax Seeds, Cocoa powder...	2.55

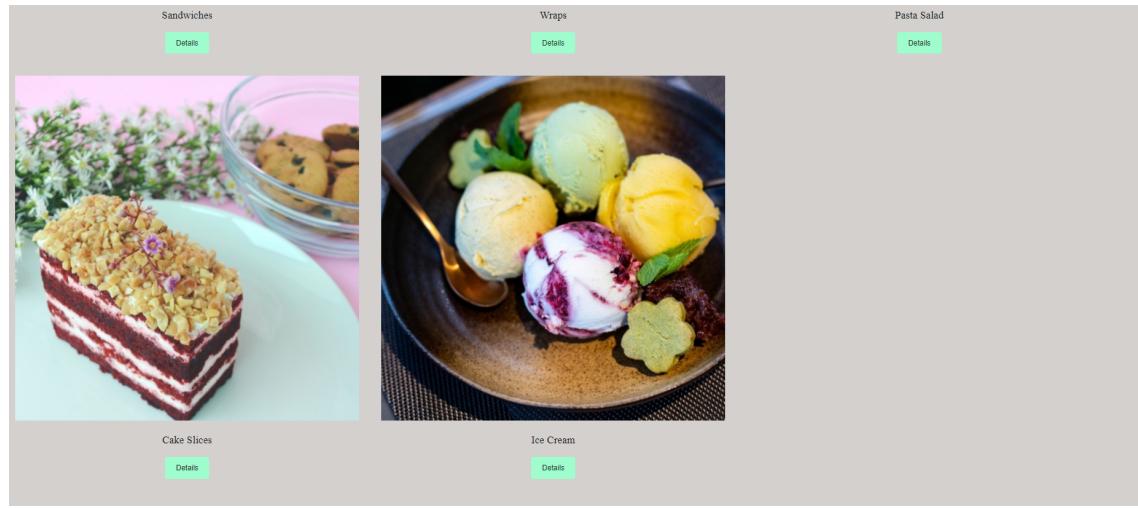
Adding a new item in the database:

ID	Name	Desc	Ingr	iPrice
1	Water	It is just water, there is nothing to explain or d...	100% Water	2
2	Iced Tea	Iced tea, peach or lemon flavoured	Water, fructose, citric acid, lemon juice concentr...	2.95
3	Tea and Coffee	White, Green and Black available. Black coffee...	Tea leaves, Coffee Beans, Plant-based and regular...	3.5
4	Sandwiches	Customise your sandwich with the choices below.	White or Oatmeal bread, Ham, Chicken, Bacon, Tuna...	3.95
5	Wraps	Customise your wrap with the choices below.	White or Oatmeal wrap, Ham, Chicken, Bacon, Tuna, ...	4.95
6	Pasta Salad	Customise your pasta salad with the choices below.	White or Brown pasta, Ham, Chicken, Bacon, Tuna, A...	5.95
7	Cake Slices	Victoria Sponge, Chocolate, Red Velvet, Coffee and...	Flour, Eggs, Water, Milk, Flax Seeds, Cocoa powder...	2.55

Database after adding a new item:

Options		ID	iName	iDesc	ingre	iPrice
<input type="checkbox"/>	Edit Copy Delete	1	Water	It is just water, there is nothing to explain or d...	100% Water	2
<input type="checkbox"/>	Edit Copy Delete	2	Iced Tea	Iced tea, peach or lemon flavoured	Water, fructose, citric acid, lemon juice concentr...	2.95
<input type="checkbox"/>	Edit Copy Delete	3	Tea and Coffee	White, Green and Black Tea available. Black coffee...	Tea leaves, Coffee Beans, Plant-based and regular...	3.5
<input type="checkbox"/>	Edit Copy Delete	4	Sandwiches	Customise your sandwich with the choices below.	White or Oatmeal bread, Ham, Chicken, Bacon, Tuna...	3.95
<input type="checkbox"/>	Edit Copy Delete	5	Wraps	Customise your wrap with the choices below.	White or Oatmeal wrap, Ham, Chicken, Bacon, Tuna, ...	4.95
<input type="checkbox"/>	Edit Copy Delete	6	Pasta Salad	Customise your pasta salad with the choices below.	White or Brown pasta, Ham, Chicken, Bacon, Tuna, A...	5.95
<input type="checkbox"/>	Edit Copy Delete	7	Cake Slices	Victoria Sponge, Chocolate, Red Velvet, Coffee and...	Flour, Eggs, Water, Milk, Flax Seeds, Cocoa powder...	2.55
<input type="checkbox"/>	Edit Copy Delete	9	Ice Cream	Chocolate, Strawberry, Red Velvet, Coffee, Vanilla...	Milk, Flax Seeds, Cocoa powder, Coffee powder, Sta...	0.95

Menu after adding a new item:



The menu item being shown in more detail after clicking 'Details':

Ice Cream

Description: Chocolate, Strawberry, Red Velvet, Coffee, Vanilla, Mint flavours available. Vegan and Gluten Free options available

Ingredients: Milk, Flax Seeds, Cocoa powder, Coffee powder, Stawberry Jam, Vanilla beans, Cream, Mint.

Price: €0.95

[Add To Favourites](#)

3.3.6 Testing Adding to Favorites List and Mailing the List

Each user can uniquely add a selected dish to favorites by pressing the 'Add to favorites' button when checking a menu items details. The user can then access there favorites list from 'account' and then by clicking 'favourites list'. This is achieved via the database by grabbing the keys of the userID and iID as shown here:

iID	userID
5	1
6	1
7	2
3	2
9	1

Logging into user with userID 1 gives this result:

Yours Favourites

Item: Wraps

Ingredients: White or Oatmeal wrap, Ham, Chicken, Bacon, Tuna, Lettuce, Avocado, Spinach, Tomatoes, Peppers, Onions, Hummus, Tofu, Mushrooms, Mayonnaise, Ranch dressing.

Price: €4.95

[Remove](#)

Item: Pasta Salad

Ingredients: White or Brown pasta, Ham, Chicken, Bacon, Tuna, Avocado, Spinach, Tomatoes, Peppers, Onions, Hummus on the side, Tofu, Mushrooms, Pesto, Tomato and Basil Sauce

Price: €5.95

[Remove](#)

Item: Ice Cream

Ingredients: Milk, Flax Seeds, Cocoa powder, Coffee powder, Strawberry Jam, Vanilla beans, Cream, Mint.

Price: €0.95

[Remove](#)

Mail List

To send the entire list to your email, click the button below.

[Send favourites List](#)

While logging into user with userID 2 gives this:

Yours Favourites

Item: Cake Slices
Ingredients: Flour, Eggs, Water, Milk, Flax Seeds, Cocoa powder, Coffee powder, Jam, Vanilla, Cream.
Price: €2.55
[Remove](#)

Item: Tea and Coffee
Ingredients: Tea leaves, Coffee Beans, Plant-based and regular milk.
Price: €3.5
[Remove](#)

Mail List
To send the entire list to your email, click the button below.

[Send favourites List](#)

Users are also given the option to remove an item from favorites by pressing the 'remove button':

Yours Favourites

Item: Cake Slices
Ingredients: Flour, Eggs, Water, Milk, Flax Seeds, Cocoa powder, Coffee powder, Jam, Vanilla, Cream.
Price: €2.55
[Remove](#)

Mail List
To send the entire list to your email, click the button below.

[Send favourites List](#)

+ Options	
ID	userID
5	1
6	1
7	2
8	1

Finally, an option is given for a user to send there favorites to there email. This is the result of when a user successfully sends there favorites list:

Yours Favourites

Item: Wraps

Ingredients: White or Oatmeal wrap, Ham, Chicken, Bacon, Tuna, Lettuce, Avocado, Spinach, Tomatoes, Peppers, Onions, Hummus, Tofu, Mushrooms, Mayonnaise, Ranch dressing.

Price: €4.95

[Remove](#)

Item: Pasta Salad

Ingredients: White or Brown pasta, Ham, Chicken, Bacon, Tuna, Avocado, Spinach, Tomatoes, Peppers, Onions, Hummus on the side, Tofu, Mushrooms, Pesto, Tomato and Basil Sauce

Price: €5.95

[Remove](#)

Item: Ice Cream

Ingredients: Milk, Flax Seeds, Cocoa powder, Coffee powder, Stawberry Jam, Vanilla beans, Cream, Mint.

Price: €0.95

[Remove](#)

Mail List

To send the entire list to your email, click the button below.

kpar0007@gmail.com

[Send favourites List](#)

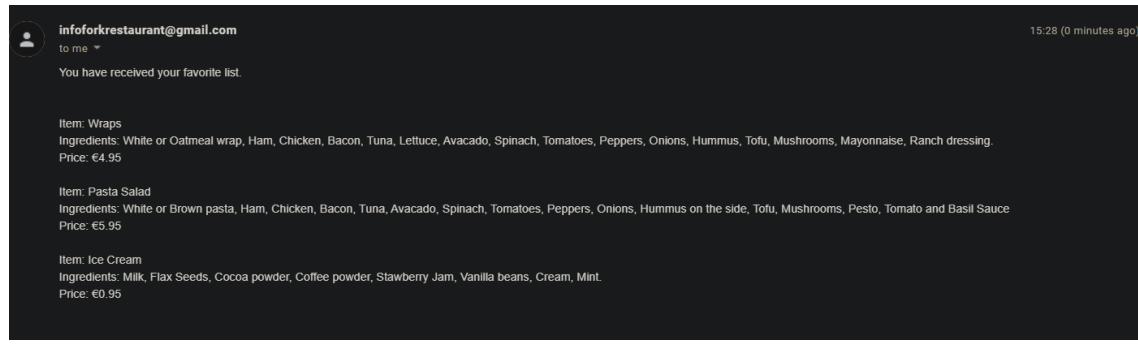
Mail List

To send the entire list to your email, click the button below.

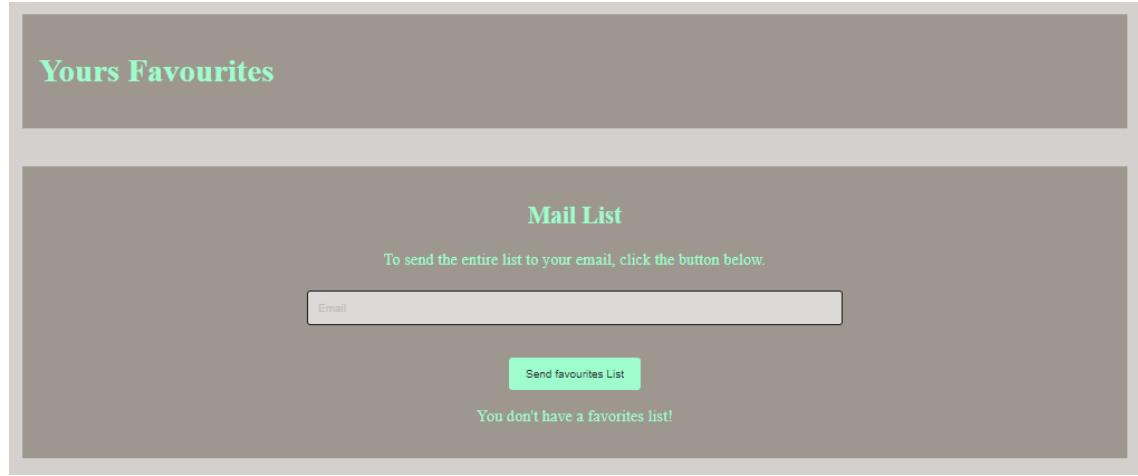
Email

[Send favourites List](#)

You have successfully sent your favorites list!



If the user doesn't have a favorites list the following error message will be displayed:



The user will also get an error message if they leave the email field empty or if the email isn't valid.

3.4 Further Implementation and Improvements

1. **Twig** - Twig is a templating language aimed to separate PHP scripts from HTML code. Incorporating Twig into the fork() website code would improve its robustness by modularising the code further. It was not incorporated due to time constraints.
2. **Default username and password for MySQL database** - The default username and password for the MySQL database were used for this implementation. To improve the database's security these would be changed.
3. **Denying the admin user the rights to drop the table/database as a whole** - As of the current implementation, the administrator does have the rights required to drop whole tables and even the database as a whole. An improvement would be to deny the admin of these rights for database security.
4. **Adding the same item multiple times to the Favourites List, despite only displaying one of each** - The current implementation of the website does allow a user to add the same item to their favourites list more than once. However, when viewing their favourites list only a single instance of an item will be shown due to the SQL DISTINCT constraint in the code. An improvement would include adding a constraint to prevent this, which will also prevent wasting space in the database.
5. **When changing user data, all data must be changed to accept the changes** - Further refinement to this function would allow changes to a particular data field instead of all of them.

4 Group Responsibilities

4.1 Individual Responsibilities

The following are the components of the website which were implemented individually:

Kian Parnis:

1. Part 1: Question 3, 4 and 5.
2. About page design and map implementation.
3. Navigation Bar.
4. Login and Register Systems, including setting up the 'login' table in the database.
5. Setting up the email functionality for the Contact Form.
6. Setting up the email functionality for the Favourites List.

Evangeline Azzopardi:

1. Part 1: Question 1 and 2.
2. Initial basic design of the website in general.
3. Information given on the About page.
4. Design and construction of the Contact and Login/Register pages
5. Menu design, functionality and the 'menu_item' database table setup.
6. Displaying the Favourites List and the 'favourites' database table setup.

Overall the fork() website was constructed and implemented through group effort, even though there were aspects that were worked on individually.

4.2 Group Dynamics

Both group members found the experience of working together enjoyable and beneficial towards the assignment implementation. The process of implementing fork() was well organised, time management was enforced and kept, the work was distributed evenly and whenever any errors were encountered, they were solved as a group.

4.3 Meeting Attendance Log

Meetings were held in an informal manner and whenever necessary. Both parties, Kian Parnis and Evangeline Azzopardi, were always present.

5 Bibliography

References

- [1] Meetanshi Blog *How to Send Mail from Localhost XAMPP Using Gmail.*
<https://meetanshi.com/blog/send-mail-from-localhost-xampp-using-gmail>
- [2] Dani Krossing *How To Create A Login System In PHP For Beginners — Procedural MySQLi — PHP Tutorial.*
<https://www.youtube.com/watch?v=gCo6JqGMi30>
- [3] Leaflet Maps *Leaflet Documentation*
<https://leafletjs.com/reference-1.7.1.html>