**Task 3 Part B**

**Francesco, Julian, Kaia**

1 – Kaia

The database for the website was built using XAMP (for Windows devices) or MAMP (for Mac). A database is made up of tables containing multiple records. Each table contains a different category of data such as users’ personal data, order history, products in store, comments on products, nationalities from where customers are buying and so forth. To have a fully functional database it is important to create the relationships between the tables. A relationship can be defined as a state between two connected database tables when one table has a foreign key that creates a link with the primary key of another table.

A relationship can take multiple forms; one-to-one, one-to-many, or many-to-many. In a one-to-one relationship, once record in Table ‘A’ is linked to only one record in Table ‘B’. A. many-to-many relationship is used when one field from Table ‘A’ can be linked to one or more records in Table ‘B’, ‘C’, ‘D’ and so forth. Lastly, many-to-many occurs when multiple records in a table can be connected to multiple records in another table. therefore, we create another table to act as a bridge between the two tables where it would look like the two main tables have a one-to-many relationship with the linking table.

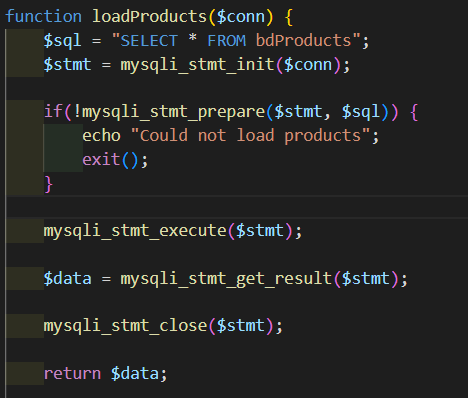
The Primary Key is a single field in a database that is unique and can therefore be used to identify a particular record. For example, the Primary Key of a table containing customer’s personal information is typically the ID card number as it is unique to every individual. A Foreign Key is a column or group of columns in a table that provides a link between data in two separate tables. It refers to the field in Table ‘B’ which is the Primary Key of the Table ‘A’. So, to build on the example given earlier, the table containing customer’s personal information would have the ID card number field as the Primary Key which would be used in the Orders table as the Foreign Key to show which customer ordered which product.

When creating a column in the database one has to decide what datatype to set it as. This depends on the type of data that will be stored in the particular column. There are three main types of data: string, numeric, and date and time. These are some examples of datatypes and how they can be used: VARCHAR is a form of string data which can contain letters, numbers, and special characters, INT refers to a medium integer ranging from -2147483648 to 2147483647, and DATE refers to a date using the YYYY-MM-DD format.

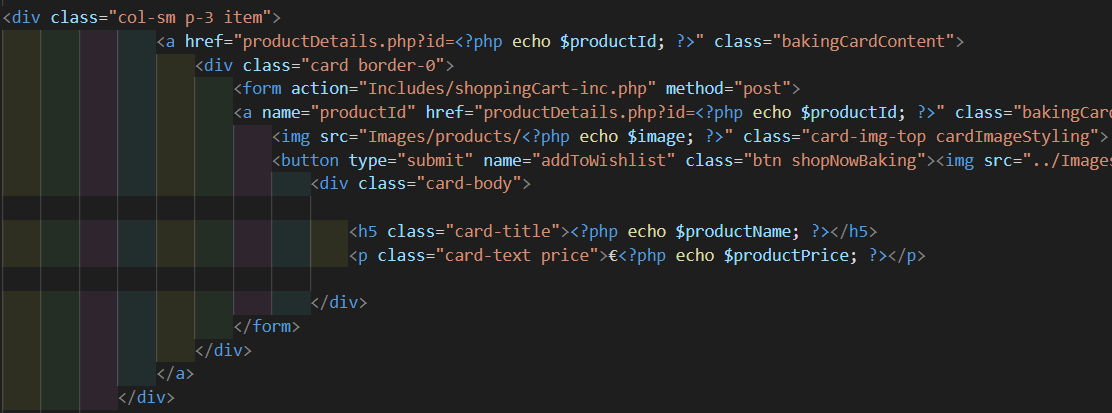
All these elements mentioned are crucial to the structure of a strong database, giving the company the ability to properly store and organize the tonnes of data they may have.

To be able to display and manipulate the data from the database, a function was created to get all the data from the table and initialise the connection to the database.

2 – Julian



A card was created to display the main details of the products, cards will keep on being created until all products have been displayed



The site will fetch the data from the database and will loop until all products have been displayed.

Text

Description automatically generated

The same is exactly repeated on UtensilsShop.php

If data is altered from the database, it will automatically update on the website.

3 - Kaia

Explain how a virtual server was set up locally on your device to mimic a live server

XAMPP/MAMPP were used to create a local server on our devices which imitates a live host server. This server software is used to test websites before they are published. They are appropriate for testing MYSQL, and PHP projects, as was done for this project. To initially set up server software, you first have to change the document root from the config file to redirect it to the repository on GitHub. The server has to be switched on every time you need to work on the SQL part of the site or see changes made in the code. To open the PHP project on the browser one would need to type ‘http://localhost:8888/’ before the rest of the URL, phpMyAdmin5/ to access the local server.

4 – Francesco

Baker’s Dozen, although it features nice design, doesn’t have all the functionalities up and running, something that we would need to continue working on to improve it. The PHP behind the website is what we struggled the most with and the best we could.

When looking forward and seeing what could be added to improve this website, there are a few functionalities that come to mind. Firstly, a dark colored theme or any type of accessibility mode to aid people with visual impairments to be able to still view and use the site.

Secondly, a feature that could be used to facilitate the contact us section could be a chatbot using the messenger API to be able to have real-time communication with a member of the Baker’s Dozen team.

Lastly and more importantly, a possible life saving feature would be to provide people with versions of the deserts that cater to their specific intolerance as a mistake in this regard could send a customer to the hospital or even worse. Providing, for example, lactose free versions or gluten free could appeal to more customers as no one would be left out of using the site due to their condition.

5 – IPO Chart - Julian

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| User creates a valid account | Adds user details ‘’bdUsers | Directs user to account.php  ‘error=none’ |
| User leaves a field empty | Checks for validation | ‘Please fill out this field’ pop up |
| User creates an account but passwords do not match | Checks for validation | ‘error=passwords-doesnt-match’ |
| User creates a valid password | Hashed password | Encrypted password on database |
| Navigation on header | Goes to link | All buttons of all pages go to correct page |
| Logo on header is clicked | Goes to link | Takes user to index.html |
| User loads Baking Packages page | Gets product information from ‘bdProducts’ | All products have an individual card with a title, price, and image |
| User loads Utensils Shop page | Gets product information from ‘bdUtensils’ | All products have an individual card with a title, price, and image |
| User clicks on a recipe and taken to its information page | Gets product information from ‘bdProducts’ | More information of the product is given such as servings, preparation time, cook time and a description. |
| User clicks on a recipe and taken to its information page | Gets product information from ‘bdUtensils’ | More information of the product is shown as well as adding the product to cart. |
| User clicks on sitemap on the footer | Goes to link | User can easily navigate through the website here |
| User clicks on the ‘Talk to us here’ button | Goes to link | Button takes user to ‘contactUs.php’ |
| User clicks on the ‘Lets fix it here button’ | Goes to link | Button takes user to ‘dispute.php’ |

5 – Test Cases - Julian

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Action | Inputs | Expected Output | Actual Output | Result | Comments |
| 1 | User Creates valid account | User fills sign up form | New user in database | New user in database with a hashed password | Pass | N/A |
| 2 | User adds items to cart | Button input | ‘Item added successfully’ Item added to database | No data outputted | Fail | Could not get function to work |
| 3 | Load all users from database | N/A | Users will be loaded from database on load | Users loaded from database on load | Pass | N/A |
| 4 | Check if passwords match on sign up | Inputs a new password | Gives error if passwords do not match | Gives error if passwords do not match | Pass | N/A |
| 5 | Checks for empty inputs | User fills sign up form | Gives pop-up message to fill the field | Gives pop-up message to fill the field | Pass | N/A |
| 6 | Checks for similar email input | User inputs email from sign up | Gives an error message if 2 emails match | Gives an error message if 2 emails match | Pass | N/A |
| 7 | Checks for empty input on log in form | Clicks log in when there are empty fields | Gives pop-up message to fill the field | Gives pop-up message to fill the field | Pass | N/A |
| 8 | Checks if the email inputted by the user matches one of the emails in the database | User fills log in form | Gives error message id emails match | Gives error message id emails match | Pass | N/A |
| 9 | User can update their profile details | User inputs new details | Details from database and website will be updated | No changes occur | Failed | Did not have enough time to fix |
| 10 | Users can delete their own account | Clicks on delete account | User will be deleted from database | No changes occur | Fail | Did not have enough time to fix |
| 11 | Utensils page will show the information of all utensils when loaded | User clicks on UtensilsShop.php | All utensils will load with their respective details | All utensils have loaded on the page | Pass | N/A |
| 12 | Products page will show the information of all products when loaded | User clicks on bakingKit.php | All products will load with their respective details | All products have loaded on the page | Pass | N/A |
| 13 | Product details page will load for each product when clicked | User clicks on one of the products | Site will give the details of the products | Site gives details of the product | Pass | N/A |
| 14 | Utensil details page will load for each product when clicked | User clicks on one of the utensils | Site will give the details of the products | Site gives details of the product | Pass | N/A |
| 15 | Site shows a few similar products in the product details | / | Site shows similar products to the user | Site shows similar products to the user | Pass | N/A |
| 16 | Site shows a few similar products in the utensil details | / | Site shows similar products to the user | Site shows similar products to the user | Pass | N/A |
| 16 | User can post a comment on the product page | User inputs their name and inserts a comment | Comment will be displayed on the database | Comment is displayed on the database | Pass | N/A |

Bibliography (for parts 1 and 3):

*Using database indexes tutorial example* (no date) *Progress.com*. Available at: https://www.progress.com/tutorials/odbc/using-indexes#:~:text=An%20index%20is%20a%20database,single%20field%20name%2C%20like%20EMP\_ID. (Accessed: January 25, 2023).

*What is a relationship? - definition from Techopedia* (2014) *Techopedia.com*. Available at: https://www.techopedia.com/definition/24438/relationship-databases#:~:text=A%20relationship%2C%20in%20the%20context,while%20linking%20disparate%20data%20items. (Accessed: January 25, 2023).

*One-to-one relationships* (n.d.) *FileMaker Pro 18 advanced help*. Available at: https://fmhelp.filemaker.com/help/18/fmp/en/index.html#page/FMP\_Help%2Fone-to-one-relationships.html%23 (Accessed: January 25, 2023).

*One-to-many relationships* (n.d.) *FileMaker Pro 18 advanced help*. Available at: https://fmhelp.filemaker.com/help/18/fmp/en/index.html#page/FMP\_Help%2Fone-to-many-relationships.html%23 (Accessed: January 25, 2023).

*Many-to-many relationships* (n.d.) *FileMaker Pro 18 advanced help*. Available at: https://fmhelp.filemaker.com/help/18/fmp/en/index.html#page/FMP\_Help%2Fmany-to-many-relationships.html%23 (Accessed: January 25, 2023).

Yasar, K. (2022) *What is a primary key? - definition from whatis.com*, *Data Management*. TechTarget. Available at: https://www.techtarget.com/searchdatamanagement/definition/primary-key#:~:text=A%20primary%20key%2C%20also%20called,have%20only%20one%20primary%20key. (Accessed: January 25, 2023).

*Difference between primary key and foreign key* (2022) *GeeksforGeeks*. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/difference-between-primary-key-and-foreign-key/ (Accessed: January 25, 2023).

*SQL Data Types for MySQL, SQL Server, and MS Access* (n.d.) *SQL data types for mysql, SQL Server, and MS access*. Available at: https://www.w3schools.com/sql/sql\_datatypes.asp (Accessed: January 25, 2023).