CPS630: Project Iterations I and II
TEAM 14

Erica Colaco 500822609
Kylah Santos 500812672
Ting Ip 500776194

Project Objectives

Pages

We implemented a home page that gives users the option to select a specific country's attraction from a variety of dropdown menus. In addition, the search function allows them to easily search for a location to see if it is available on our database. Clicking on the read more button will redirect the user to a new page that will give further information/images on the selected location, as well as suggested locations they could visit. Our shopping cart page will provide guests with the ability to view and select travel packages for purchase.

Languages and Tools

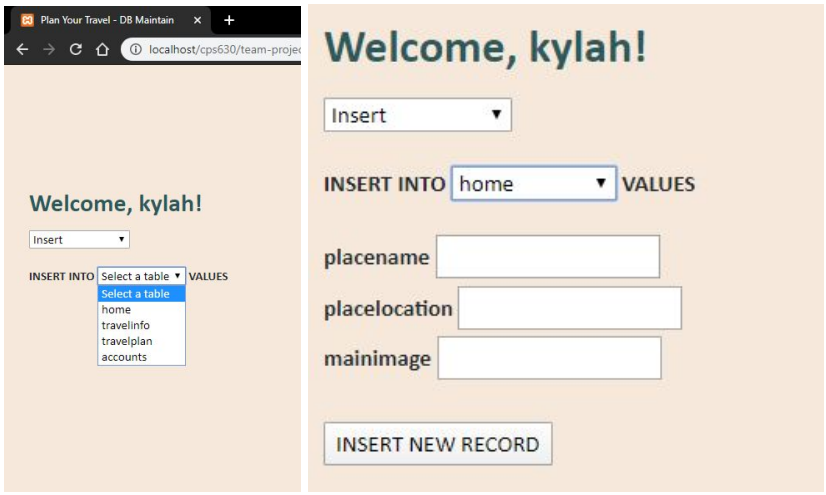
We used a variety of languages/frameworks including:

- HTML
- Javascript
- CSS
- Bootstrap
- phpMyAdmin

We also used XAMPP as our tool for hosting our site and organizing our database.

DB Maintain Code: dbmaintain.php

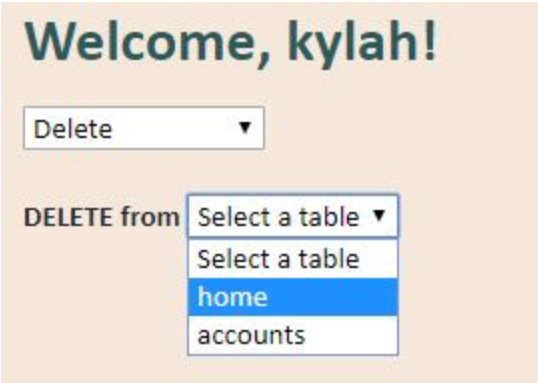
INSERT into TABLE



SQL Code (PHP):

"**INSERT** INTO home(placename, placelocation, mainimage)VALUES ('".$_POST["placename"]."','".$_POST["placelocation"]."','".$_POST["mainimage"]."')";

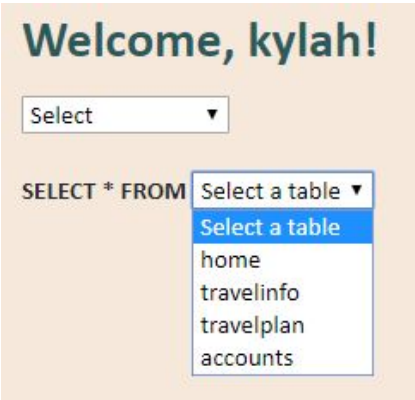
DELETE from TABLE

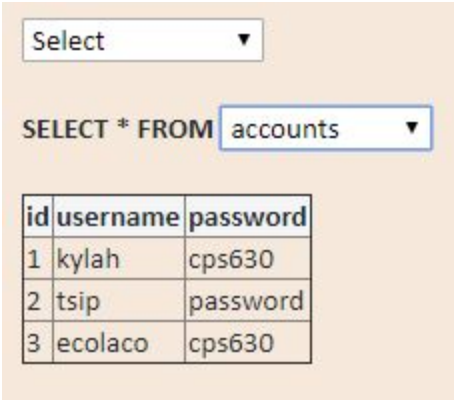


SQL Code (PHP):

"**DELETE** from home WHERE place_id=".$deleteid;

SELECT * FROM TABLE

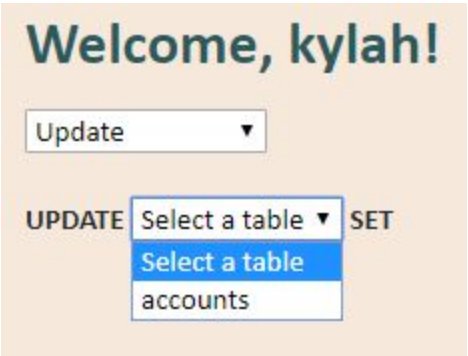


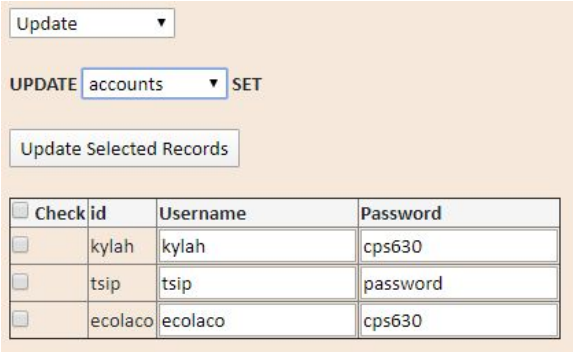


SQL Code (PHP):

"**SELECT** * FROM home"

UPDATE TABLE SET





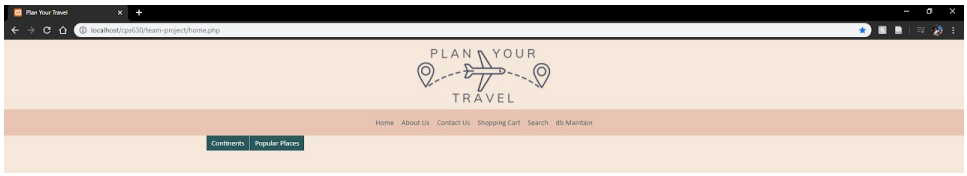
SQL Code (PHP):

"UPDATE accounts SET username='".$username."',password='".$password."'
WHERE id=".$updateid;

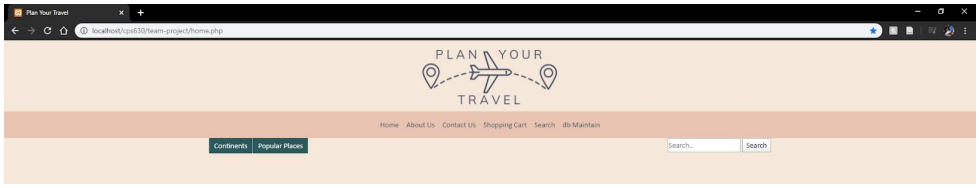
Design and Layout

All our pages use the Bootstrap framework to create a responsive system that can be accessed from many browsers/platforms. Each page that the viewer interacts with has a navigation bar at the top that allows them to go to the home page, the about us page, the contact us page, the shopping cart and the database maintain page.

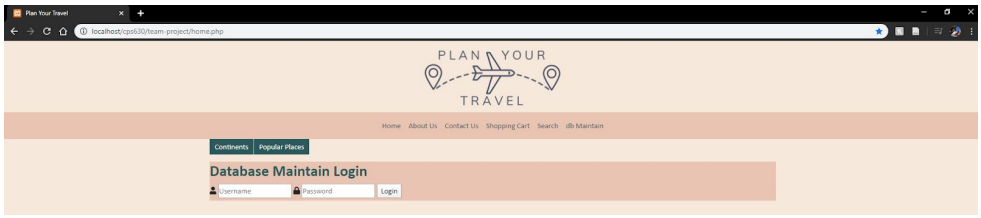
MAIN PAGE



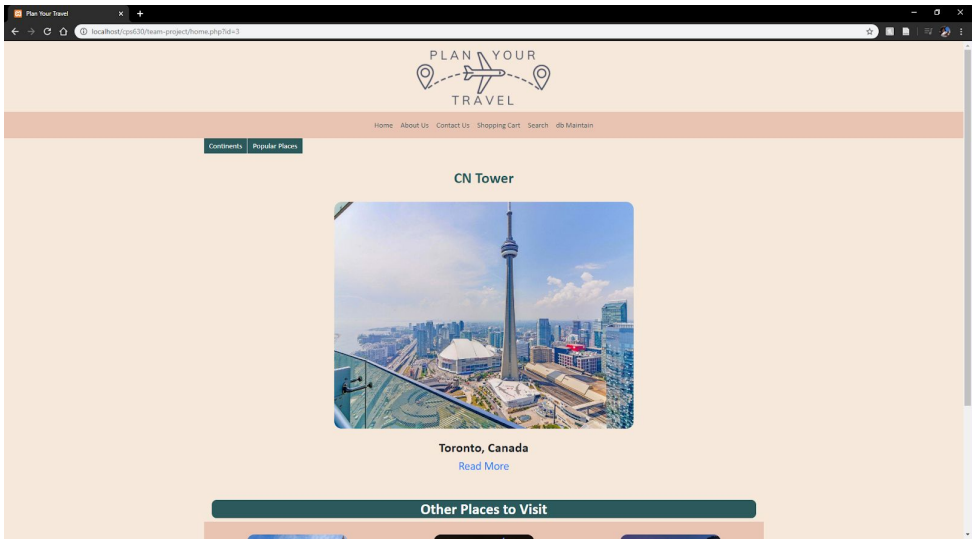
MAIN PAGE: search function shown



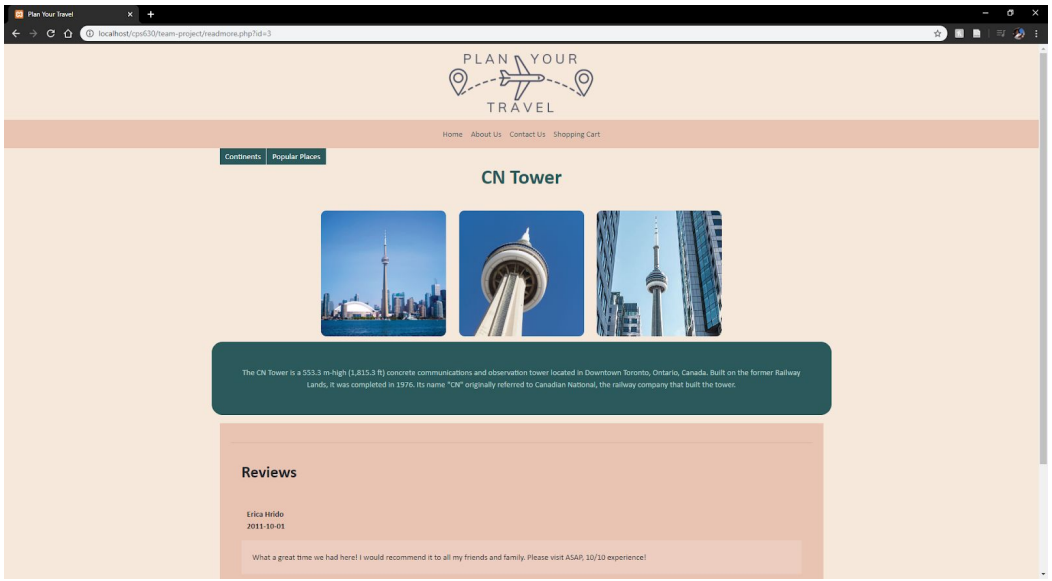
MAIN PAGE: db maintain login shown



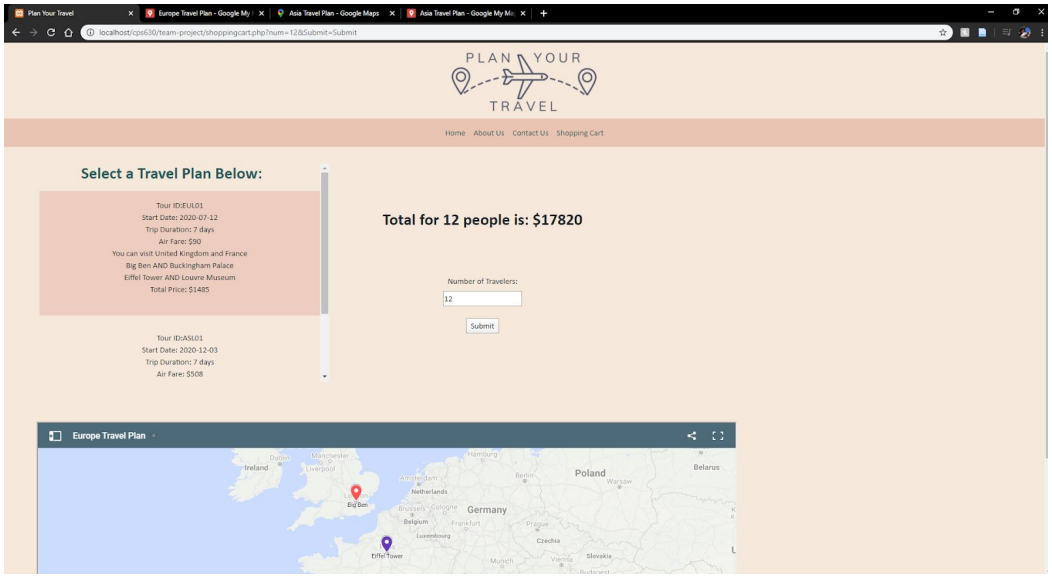
THE HOME PAGE WHEN A USER SELECTS AN ATTRACTION:



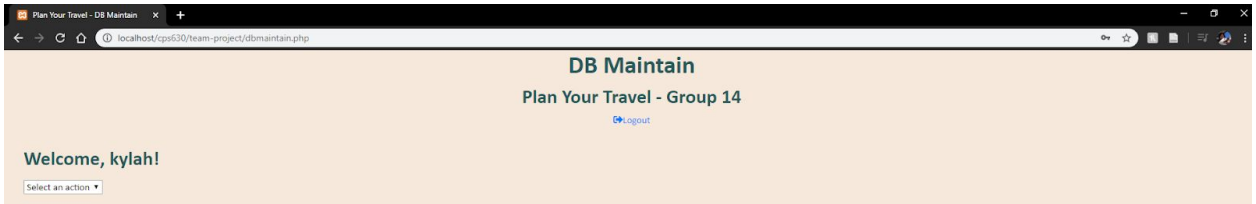
THE READ MORE PAGE FOR THE ATTRACTION USER SELECTED



SHOPPING CART PAGE



DB MAINTAIN PAGE



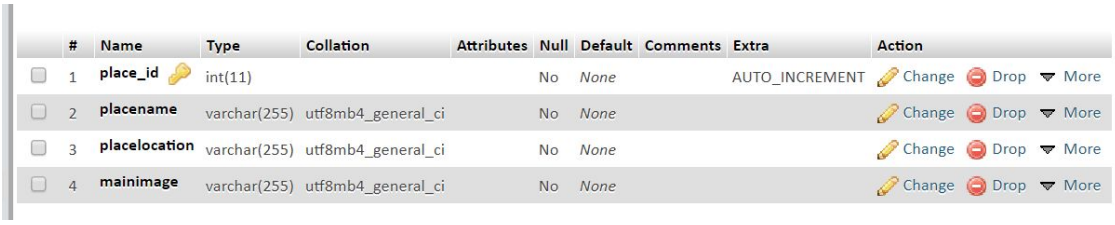
Database Design and Structure

Name: home

Fields: place_id, placename, placelocation, mainimage

Primary Key: place_id

This table provides the foreign key for the closedistances, landmarks, nature and museums tables.



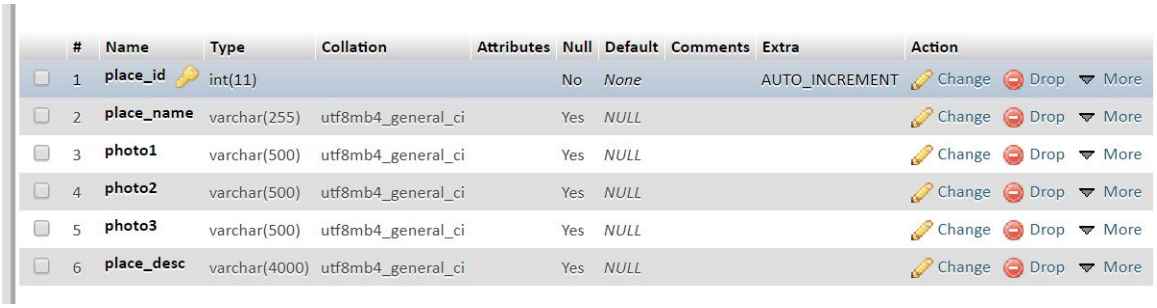
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	place_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	placename	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	placelocation	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	mainimage	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Name: travelinfo

Fields: place_id, placename, photo1, photo2, photo3, place_desc

Primary Key: place_id

This table is connected to the home table, review table and



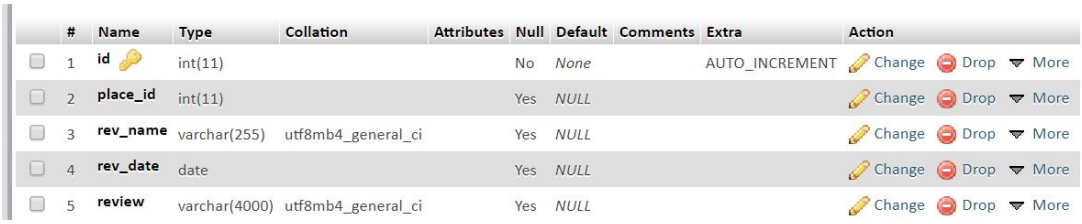
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	place_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	place_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	photo1	varchar(500)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	photo2	varchar(500)	utf8mb4_general_ci		Yes	NULL			Change Drop More
5	photo3	varchar(500)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	place_desc	varchar(4000)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Name: review

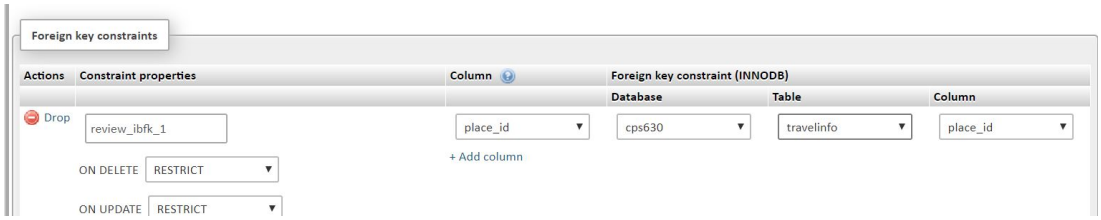
Fields: place_id, rev_name, rev_date

Primary Key: place_id

This table is connected to the travelinfo table, they share place_id's.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	place_id	int(11)			Yes	NULL			Change Drop More
3	rev_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	rev_date	date			Yes	NULL			Change Drop More
5	review	varchar(4000)	utf8mb4_general_ci		Yes	NULL			Change Drop More

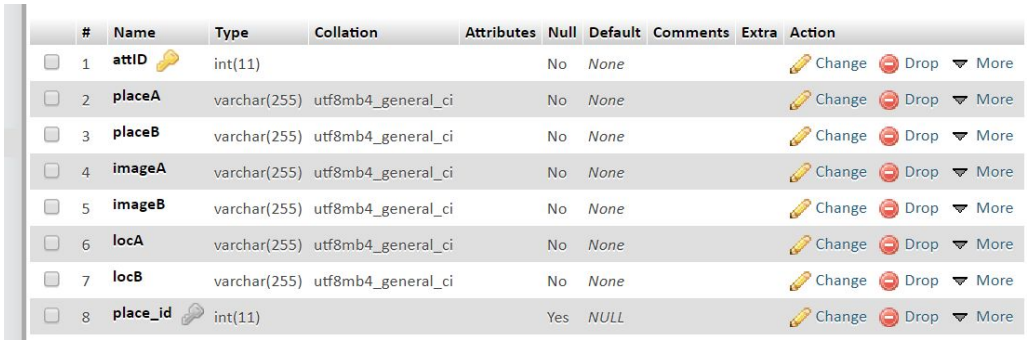


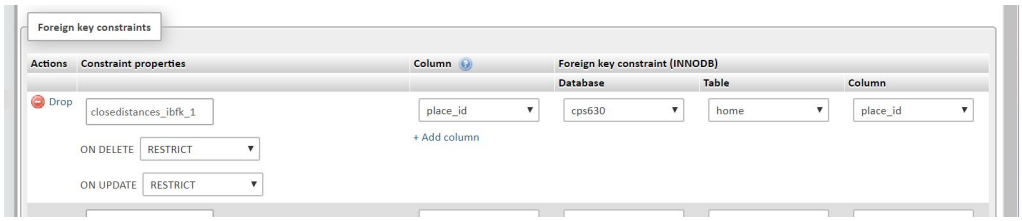
Name: closedistances

Fields: attID, placeA, placeB, imageA, imageB, locA, locB, place_id

Primary Key: attID

This table references the home table to get the foreign key “place_id”.



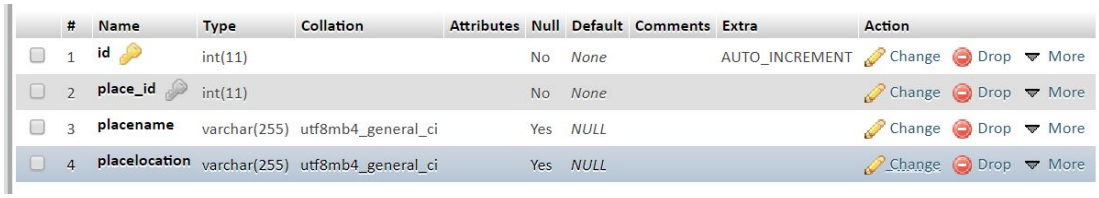


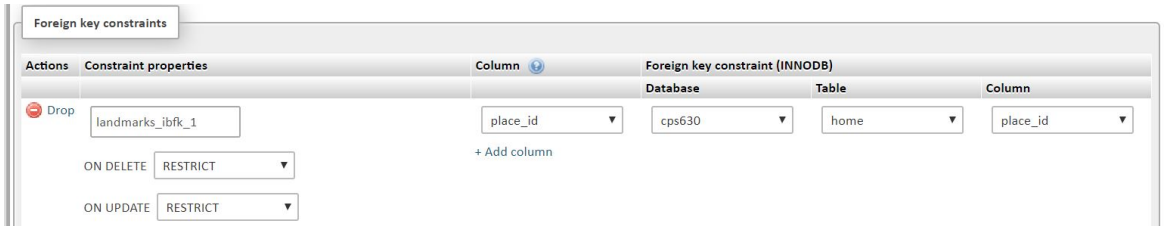
Name: landmarks

Fields: place_id, placename, placelocation, mainimage

Primary Key: place_id

This table references the home table to get the foreign key “place_id”.



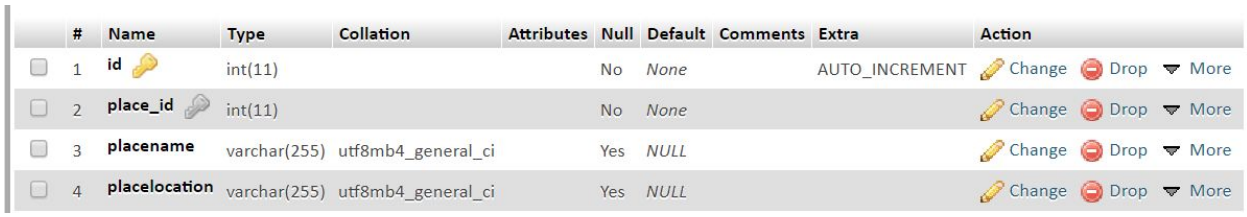


Name: nature

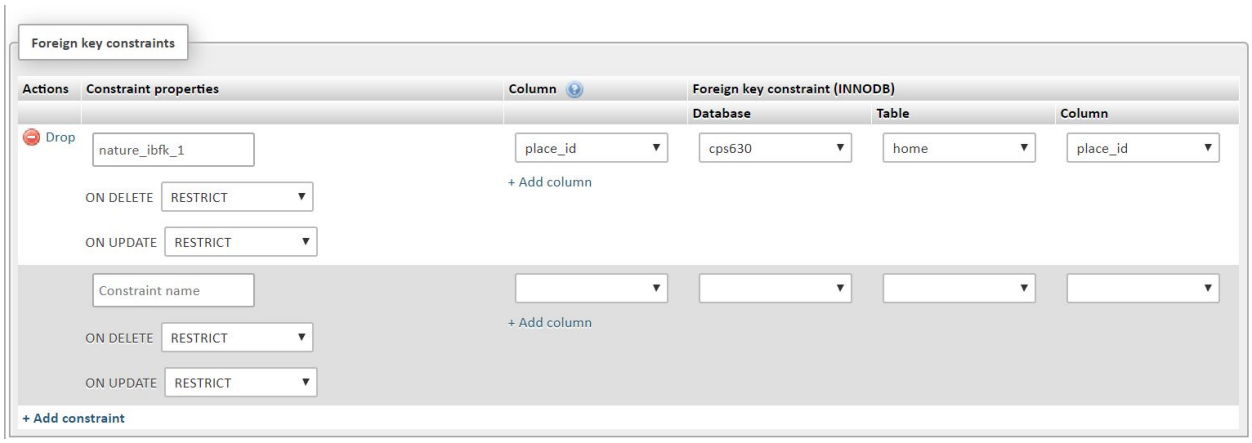
Fields: place_id, placename, placelocation, mainimage

Primary Key: place_id

This table references the home table to get the foreign key “place_id”.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	place_id	int(11)			No	*None*			Change Drop More
☐	3	placename	varchar(255)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
☐	4	placelocation	varchar(255)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More



Foreign key constraints

Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
Drop	nature_ibfk_1	place_id	cps630	home	place_id
	ON DELETE RESTRICT	+ Add column			
	ON UPDATE RESTRICT				
	Constraint name				
	ON DELETE RESTRICT	+ Add column			
	ON UPDATE RESTRICT				

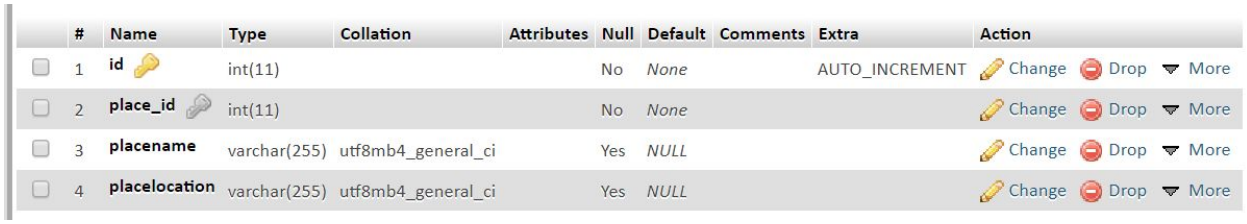
+ Add constraint

Name: museums

Fields: place_id, placename, placelocation, mainimage

Primary Key: place_id

This table references the home table to get the foreign key “place_id”.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	place_id	int(11)			No	*None*			Change Drop More
☐	3	placename	varchar(255)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
☐	4	placelocation	varchar(255)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More

Foreign key constraints

Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
Drop	museums_ibfk_1	place_id	cps630	home	place_id
	ON DELETE RESTRICT	+ Add column			
	ON UPDATE RESTRICT				

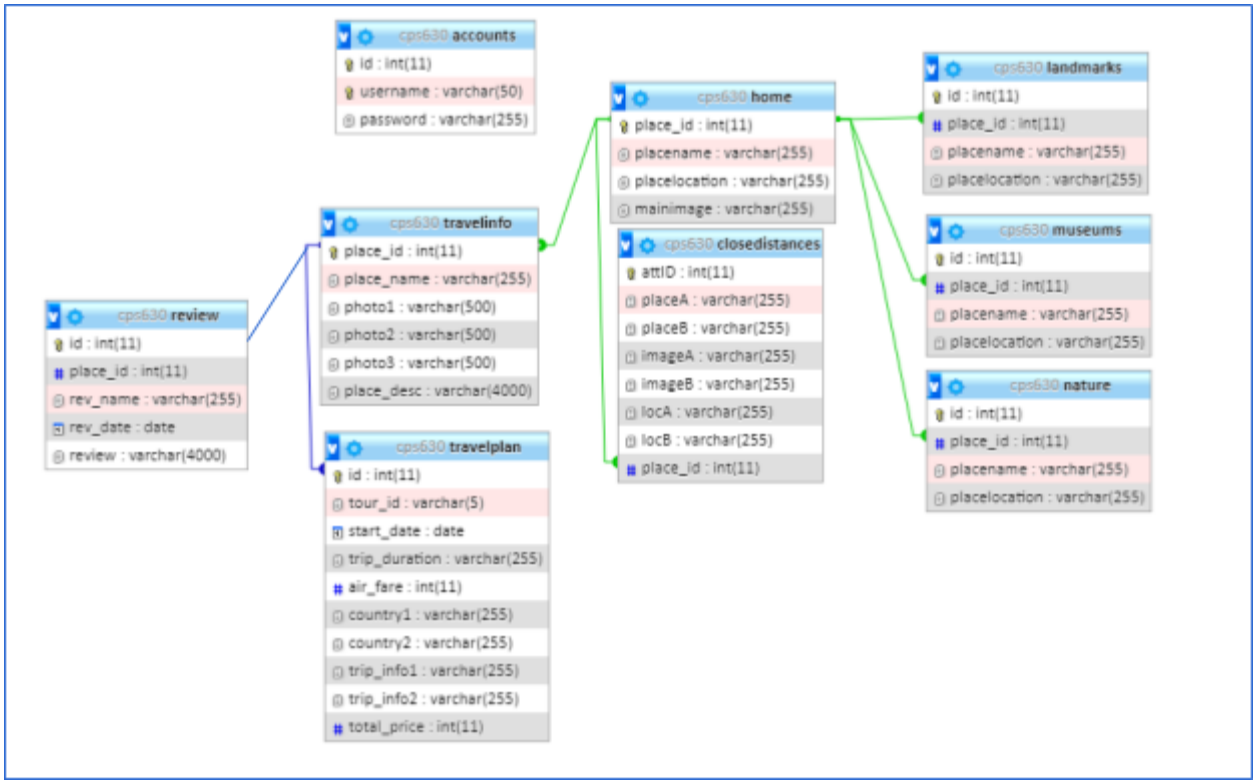
Name: accounts

Fields: id, username, password

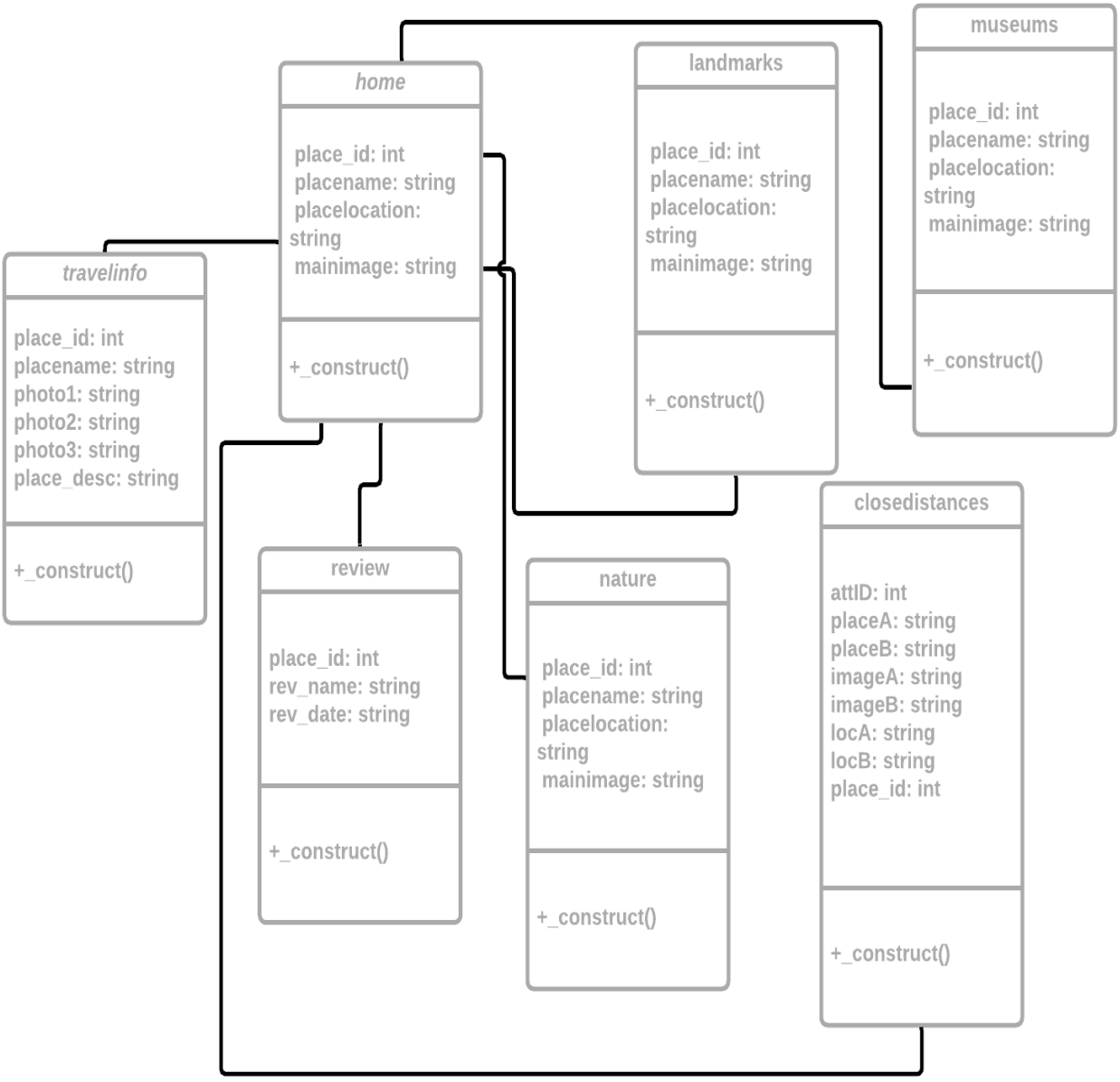
Primary Key: id

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	username	varchar(50)	utf8mb4_general_ci		No	*None*			Change Drop More
3	password	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

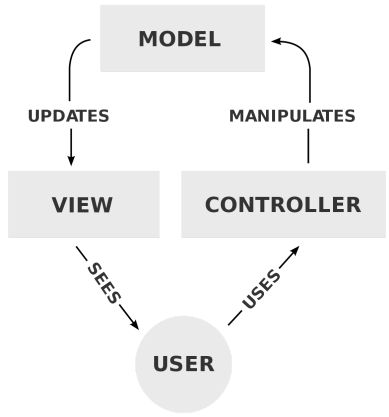
Database Schema Design



Class Diagram



MVC Pattern



Model:
The model in our architecture is our databases and all the data they contain. This includes tables such as the home table, travelinfo table and review table. To access and manipulate our database, we used PHP.

View:
The view in our project is our user interface, which was mainly built using HTML/CSS. We also used the Bootstrap framework to make our design more responsive. Our UI design displayed the table data in an attractive manner.

Controller:
The controller in our architecture is the database maintain page. This page is only accessed by valid database administrators; records only found in our accounts table. The controller allows a database admin to manipulate our database tables by inserting new records, updating records, deleting records and or selecting records. The controller was implemented using HTML, CSS, JavaScript, jQuery and PHP.

Task Contribution

Team Member	Tasks	Percent
Erica Colaco	• Created accounts, travelinfo, and review database tables • Designed both client and server side of Read More and Shopping Cart pages • Provided outline and wording of technical report • Created class diagram • Chose colour scheme and designed logo • Compiled information for each attraction • Created the travel plans	40%
Kylah Santos	• Created the home, closedistances, museums, nature, landmarks database tables • Designed both client and server side of Home and DB Maintain pages • Implemented the dropdowns for the attractions • Implemented the search function in the main page • Implemented the login process for the DB Maintain, so only valid db admins can access it • Contributed to the technical report, providing screenshots and Database Schema	40%

Ting Ip	• Implemented About Us and Contact Us modules • Created outline using PHP classes	20%