|  |
| --- |
|  |
| Project |
| SOFT7008 – Server Side Web Development |
|  |
|  |
|  |

### Specification

Using as a foundation the Pizza Ordering Web Page (PizzaWebsite.zip) contained in the Assignment folder in Dropbox, create a web application to the following specification:

**DONE!!**

1. Create a database called pizza.
2. Create a database table called orders, with the following table structure:
   * id: primary auto\_index int
   * order\_id: varchar(20)
   * student: char(1) Y/N

**DONE!!**

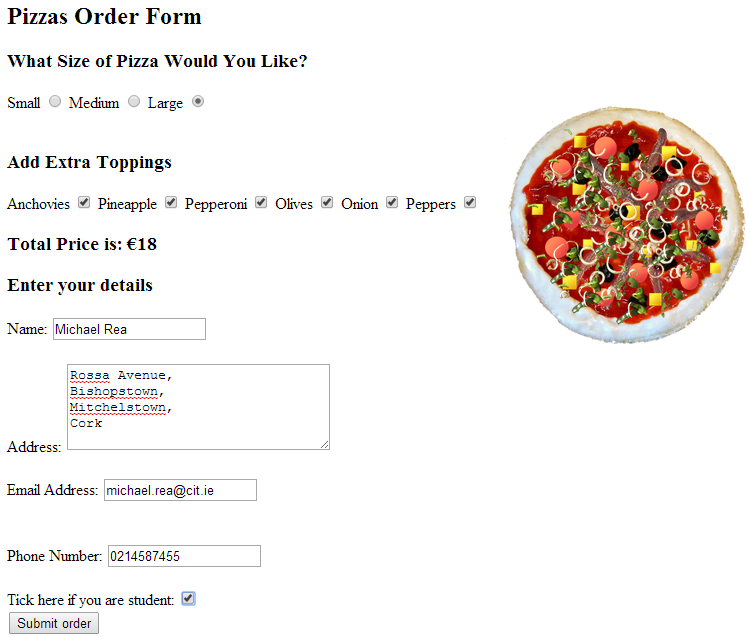
* + firstname: varchar(50)
  + lastname: varchar(50)
  + email: varchar(50)
  + address: varchar(250)
  + phone: varchar(15)
  + price: demical
  + size: varchar(20)
  + anchovies: char(1) Y/N
  + pineapples: char(1) Y/N
  + pepperoni: char(1) Y/N
  + olives: char(1) Y/N
  + onions: char(1) Y/N
  + createddatetime: datetime

**DONE!!**

* + views: int(4)

1. Put database connection and database selection code into a separate database includes file.
2. Create a PHP file called order.php which is based on the Pizza Order Web Page that you downloaded from the link above.

**DONE!! Ish might need to change**



**DONE!!**

1. Set the form action to vieworder.php.
2. Create the PHP file vieworder.php which accepts the contents of the form submission above (as $\_POST) and inserts the details entered to the database table orders as a new record.

**DONE!!**

1. Save the Pizza Order form contents to the database table, once the user has completed the form, such that none of the database table columns listed above is empty, including the createddatetime column, which should be included as part of the database insert query as the server current date and time.

**DONE!!**

1. On submission of the form, show the customer an appropriately designed/formatted receipt of their order that includes on it:
   * A unique order\_id, generated by using the php function uniqid();

**DONE!!**

* + The details of the order saved to the database, including the created date time.
  + A hyperlink to vieworder.php?order\_id=<the\_order\_id>

Eg, <http://localhost/vieworder.php?order_id=54ff645d4506a>

1. The vieworder.php page must also be able to accept, using the $\_GET[] superglobal, the order\_id as part of a parameter in the URL used to call the page. The vieworder.php page should

**DONE!!**

* + Select the orders database table record where it finds the orders.order\_id matching the $\_GET[] order\_id information.
  + Present the same type of receipt/order details as required above.

1. If vieworder.php page finds the order\_id and returns the details, provide the customer with an option to update their order. Clicking on the option to **update** their order should show the customer a “sticky form” on the vieworder.php page, whereby the same form seen in order.php is shown in vieworder.php, with all the relevant form fields populated.

**DONE!!**

For example, if the order is found, and the order was saved with “Student” checkbox ticked in the order, then show the “Student” checkbox as ticked.

1. With the action of this form set to itself (vieworder.php), submitting the form (as $\_POST) should update that same record on the database table where the order\_id is found to match. Any changes should be reflected in the database table row after the database update action, such that the updated order details would display if the order link was opened once more eg <http://localhost/vieworder.php?order_id=54ff645d4506a>.

**DONE!!**

1. Finally, the vieworder.php page with existing order information loaded (eg <http://localhost/vieworder.php?order_id=54ff645d4506a>) must give the customer the ability to delete their order. Clicking on this delete option should submit via a hidden form field (as $\_POST) the order\_id to vieworder.php, whereby the database table record the matches the order\_id is deleted. Ask user to confirm that they wish to delete order.

**DONE!!ish**

1. If, for example by calling <http://localhost/vieworder.php?order_id=54ff645d4506a> web page the order\_id cannot be found, the customer should be presented with a “cannot find order” style of error message.

**DONE!!ish**

### Instructions

* Project to be implemented in PHP only – no client side browser code.
* Use coding best practices: [http://thisinterestsme.com/php-best-practises/.](http://thisinterestsme.com/php-best-practises/)
* **Zip** your application folder containing your code and database **data**.
* Name the compressed file using the following convention: joe\_bloggs.zip
* Email your project file to <mailto:michael.rea@cit.ie>.
* It is your responsibility to ensure that all the files required to run your program are included in the compressed file and that no alteration is necessary to run the program.
* Submissions to be received by 23:59 on Sunday 10th May 2015.

### Marking Scheme

**DONE!!**

**DONE!!**

**DONE!!**

**DONE!!**

|  |  |
| --- | --- |
| Description | Mark |
| New Order   * order.php: Order created in database with all data correctly saved. * vieworder.php: “Receipt” page showing order details and URL to order details, displayed after new order is saved to database. | 15  10 |
| Update Order (vieworder.php)   * “Receipt” page showing order details if found by URL. * “Sticky Form” showing all order details populated in form, and where the form is “hidden & disabled” until the customer chooses the “update” option following display of the order receipt details. * Functioning “update” of “enabled” sticky form order details to same database row. | 10  25  15 |
| Delete Order (vieworder.php)   * “Receipt” page showing order details if found. * Functioning “delete” of order details. | 0  15 |
| Order Not Found (vieworder.php)   * “Receipt” page showing “error message” if order not found. | 10 |

### Late Submission Guidelines & Penalties

* Up to 7 calendar day delay: 50% of the marks available.
* Over 7 calendar day delay: 100% of the marks available.