The website is split into 3 parts and this document will explain the design of each page within each part. The fourth part explains the utility functions.

A small note on the database design, the type mediumtext was used for the “foods” column in the table pending_order and completed_order to accommodate for large serialized arrays. Otherwise, the primary keys and unique keys were those chosen logically for what worked as primary and unique keys in the specific tables.

PART 1: USER FRONT END (www.instanomz.com) — This is the part of the site where the user (the customer) interacts with InstaNomz to make orders.

logout_user.php

If a guest logs out, they lose access to their pending/completed transactions. They will still receive email notifications. This is done to incentivize users to sign up and also because as a guest, this data is not associated with any specific id rather is associated to a $_SESSION. In a future iteration, a unique id can be associated with a guest so a guest can resume their session if they know their unique id (which can be emailed to them).

index.php

Queries the restaurants_data table to get the name of the opening image to display for the restaurant in addition to the restaurant name. If the user is logged in with an account, I query the database for the user’s name to say hello to the user. If the user is signed in as a guest, I access the $_SESSION information for their name. In future iterations (applicable to multiple files below), validation and data retrieval could be done and improved with javascript (ajax & jquery) to prevent excessive querying when the same data may have already been queried a page earlier. Time prevented this from happening in this iteration.

register_user.php

Validates all inputs to make sure they are legitimate (i.e. non-empty, real e-mail, real mobile-number, already exists). Then, this is inserted into the table called users. For future iterations, before registering an email, the user needs to confirm their email to prevent users from stealing others emails.

pending_transactions_user.php

Pending transactions are received by querying the table pending_order using the user’s id or by looking at the orders the guest has made during the session and retrieving those orders from pending_order if they are still there.

menu_user.php

Data is sent by GET so the user can bookmark certain restaurants they regularly enjoy ordering from. The menu is retrieved from the menu table by querying it for the given restaurant id sent via GET. The current order is saved in the $_SESSION so it can be easily referenced across PHP files because many different pages need this information. If the user is not logged in, they must log in either as a guest or in their account and then they can continue to confirm their order. To achieve this, $_SESSION remembers the next page to go to after completing a certain task.

login_user.php

Validates log-in by querying database.

guest_user.php

Validates guest registering, but allows the guest to put in duplicate emails/phone numbers that may already exist for another user. This is allowed because it is temporary, allows the user to quickly make an order, and there is no risk of fraudulence as the guest user's order must be confirmed. There is a potential for spamming an email so email confirmation is likely to be involved in a future iteration.

confirm_order_user.php

Populates the order to confirm using the current order stored in $_SESSION. When pressing submit it is added to pending_order table and the user and deliverer is sent a email notification. The latter is also sent a text notification so that the deliverer can begin the confirmation process. Otherwise, if cancel is pressed, the user is redirected to index.php and the current order is deleted from $_SESSION as the user is not interested in it. If edit is pressed, the user returns to modify their order with it being pre-populated with the data from $_SESSION. If the user is a guest, the uniqueid of this submission must be saved in $_SESSION if the order is confirmed so the guest can reference it again in pending or completed transactions as the guest themselves has no identifying id as this would be a waste of space as it is only a temporary account.

completed_transactions_user.php

Queries the completed_order table for completed transactions given a user's id or from the orders given by a guest during his or her session.

PART 2: ADMIN BACK END (www.instanomz.com/login_admin.php) — This is the part of the site where the admin (the deliverer) changes the status of pending transactions (deletes them, sets them as completed, or changes their status in the stages of delivery)

GENERAL DESIGN NOTE: Admin and Restaurant backend does not validate inputs. There is an assumption of rationality and education on their parts. For future iterations when there is more time, validation will be put in to avoid potential mistakes on part of the restaurants and administrators.

USERNAME: akshar

PASSWORD: akshar

completed_transactions_admin.php

Queries completed_order table for all the completed transactions.

index_admin.php

Retrieves all pending orders from pending_order and categorizes them according to their point in delivery. This allows the deliverer to address unconfirmed orders easily and confirm them one by one. There is an edit button per menu item to allow the admin to edit each menu item as they confirm it or as they receive notification that it has been delivered/picked up etc. To edit a pending_order, the database is queried again just for the specific pending_order. In future iterations, the website will attempt to store this data offline or cache it with the user so the data does not have to be queried each time. Time was a limitation in this case.

login_admin.php

Validates by checking the values in the admins table.

update_admin.php

Sends email if status changes keeping users and deliverers up to date with the status of the orders. If deleted, it is deleted from pending_order, if completed, it is deleted from pending_order and added to completed_order, and if anything else is updated, it is updated in pending_order.

PART 3: COMPANY BACK END (www.instanomz.com/login_company.php) — This is the part of the site where the company (the restaurant) can update their menu offerings, view their pending orders, and view their completed orders.

USERNAME: boloco
PASSWORD: boloco

update_company.php

The specific food item in the menu table is updated or deleted using a query.

pending_transactions_company.php

The pending transactions using the restaurant's id is queried.

new_item_company.php

A new menu item is inserted into the menu table with the restaurant's id.

login_company.php

Validates by checking the values in the restaurants table.

index_company.php

Retrieves the menu items for the specific restaurant from the menu table and allows them to edit each specific one. Upon editing, an update query is sent for the menu item in the menu table.

completed_transactions_company.php

This page retrieves the completed transactions from completed_table for the given restaurant and displays it.

PART 4: UTILITY FILES

PHPMailerAutoLoad.php, class.smtp.php, class.pop3.php, class.phpmailer.php

These files handle PHPMailer.

config.php

Config is set up to list the pages that are accessible if the user is logged in and if not, they will be redirected to the respective log-in page or index page for the user front end.

function.php

Render and apologize has been modified to accommodate for different headers for the 3 different parts of the website. This is so that the link of the header file redirects the user to appropriate page i.e. index.php, index_admin.php or index_company.php.