

### Part 1:

The website is a Wine website which attempts to create an informative experience while buying wine. The users in the website will need to be above the age of 21, as this website sells alcoholic products. The website turned out working perfectly with the integration to the backend with no major problems.

On the home page, there is a section where there are products shown in order based on their ratings. This section is the top-rated wines, and it is all pulled from the database. The browse page lists all the wines on the database with the newest first. Therefore, if you were to make any changes through the “edit” interface, it will appear first on the browse page. The browse page organizes the products into boxes. Each box has a picture of the product, title, description and price. The About Us page is the only page without any admin actions when an admin logs in. This is because it is a simple page that just informs users about the company.

When the administrator of the website logs in, the website reveals all the admin privileged options. These options are in every page besides the about us, as mentioned above. The admin is a user with ID 1, in the user table of the SQL database. The credentials currently are **Username: sushant94, Password: 123**. This can be changed anytime on the SQL database through the “Users” table. Currently there is only the option to login as administrator, however this can easily be expanded later since it is connected to the database. There is also form detection on the login inputs and the newsletter inputs. If users do not complete both inputs, the input turns red and does not submit. The newsletter form also has detection for emails. If the user does not put in the proper format for an email, they are reminded. This provides the user with a better experience in that if they accidentally submit without finishing, they won’t have to restart.

Everywhere a product is seen after the admin logs in, they can do two options. The edit and delete option come up. Also, an add item tab appears on the top navigation bar allowing the admin to add an item. After the admin logs out, it disappears. The delete button requires a confirmation through a pop up window for it to go through. The edit button causes a redirection to another page where the forms with information show. The forms are completed with information currently on the database. The add page is exactly like the edit page except the forms are not filled in and are left empty for the admin to fill out. The view product page is loaded with database data. For example, the rating, price, and volumes. The reviews are static, but can easily be integrated to be pulled from the database. Also, as mentioned above when the admin logs in they will see the operational edit and delete buttons.

Everything is done with the MVC format in mind. There’s a site controller that handles the home, parts of the browse, and aboutUs pages. It also handles the logging in/out and newsletter operations. The product controller does everything related to managing and displaying the products based on their view. Both controllers rely on the .htaccess to give them

actions based on user interactions on the html. The controllers also make good use of the classes inside the models directory.

## **Part 2:**

- First step is to unzip the cs3744.zip
- Move the folder into the htdocs folder inside the XAMPP directory
- Create SQL database named bottleup through PHPMYAdmin
- Import bottleup.sql
- Go to <http://localhost/cs3744/project3/sushantb/> on your browser

## **Above and Beyond:**

I created a separate class to manage users from the newsletter form located at the bottom. This class handles the information of a user who would like to sign up. It uses the newsletter\_signup table on the SQL database. Also, when a user inputs an incorrectly formatted email address, they are informed that their input is incorrectly formatted and are unable to submit the form. They also can't submit anything that is empty. The forms turn red until they properly fill it out. The products on the front page are ordered from the top rated to the least rated 5 products. They are all retrieved from the database. If someone tries to go onto a page through the URL that they shouldn't access (localhost/addItem) they are greeted with an "Admin Only" message on the page.

When the administrator logs in, the login bar disappears and the submit button becomes a logout button. A message is always displayed on the top notifying that the admin user is logged in. Lastly, the logout button at the top is fully functional and causes the admin to logout when it is pressed. This causes the website to return to a state where all the admin privileges are gone.