**Programming Dynamic Websites**

**Final Exam (Set 1)**

**Marks: 30 Time: 120 minutes**

Create an application for a Juice Shop that will connect to a MongoDB database for storing all the orders. This application should be built using **HTML, CSS, JS, EJS, Node.js and MongoDB** only. All the business logic/handling form submissions should be on the server side only.

The Juice Shop sells **Lichi Juices ($3.99 each), Beet Juices ($2.99 each), Peach Juices ($3.49 each)** only.

To place an order, a customer is required to provide their **Name and Phone** number. The phone number should be a 10 digit number. **Use only server side validation for Name and Phone number. Do not use input type number or min/max/required attributes from HTML.**

Once an order is placed, the user should be shown an order confirmation with all the order details along with **Sub Total (total cost of all juices), Quebec Tax(@** 14.975%**) and Total Cost (Sub Total + Tax)**.

Create a separate page to **view all the orders** in a tabular format. The table should show all the information from the orders along with Sub Total (total cost of all juices), Tax (@ 14.975% and Total Cost (Sub Total + Tax).

Please refer to the wireframes attached to get an idea on what the output may look like. The calculations on the wireframes may or may not be correct. You are free to choose the design, fonts and colors as long as you fulfill the minimum requirements mentioned. Use of some basic CSS is expected to make the application user friendly.

**Submission Checklist:**

1. All the code files and image files along with any dependencies zipped properly.
2. Word doc with screenshots for:
   1. Order form page
   2. Order form page showing validation errors
   3. Order success/thank you page
   4. View orders page
3. Git commits for your project that are spread over the time allotted for the exam (not all in the last 5 minutes). The exam will not be marked without at least 1 commit.

**Note:** If you are unable to take the screenshot due to an incomplete task, just write “Task not done” under the appropriate heading.

***Please turn over***

**Submission Guidelines:**

To submit this evaluation, place all of your work as per the checklist into a single folder. Zip this folder and submit the zipped file to eConestoga. No cover sheet is required. A good way to make sure everything is correct; is to unzip the file you are submitting on a different location on your computer and making sure everything works. Before leaving the exam, student should check that the submission was successful. Incorrect submissions cannot be graded as there is no way to resubmit afterwards. It is the responsibility of the student to ensure the submission was successful. **Uploading a broken archive(zip) file or a blank archive(zip) file/blank files will receive 0 marks. Submissions not done before the dropbox closes will also receive 0 marks. Submissions not including at least 1 git commit will also receive 0 marks.**

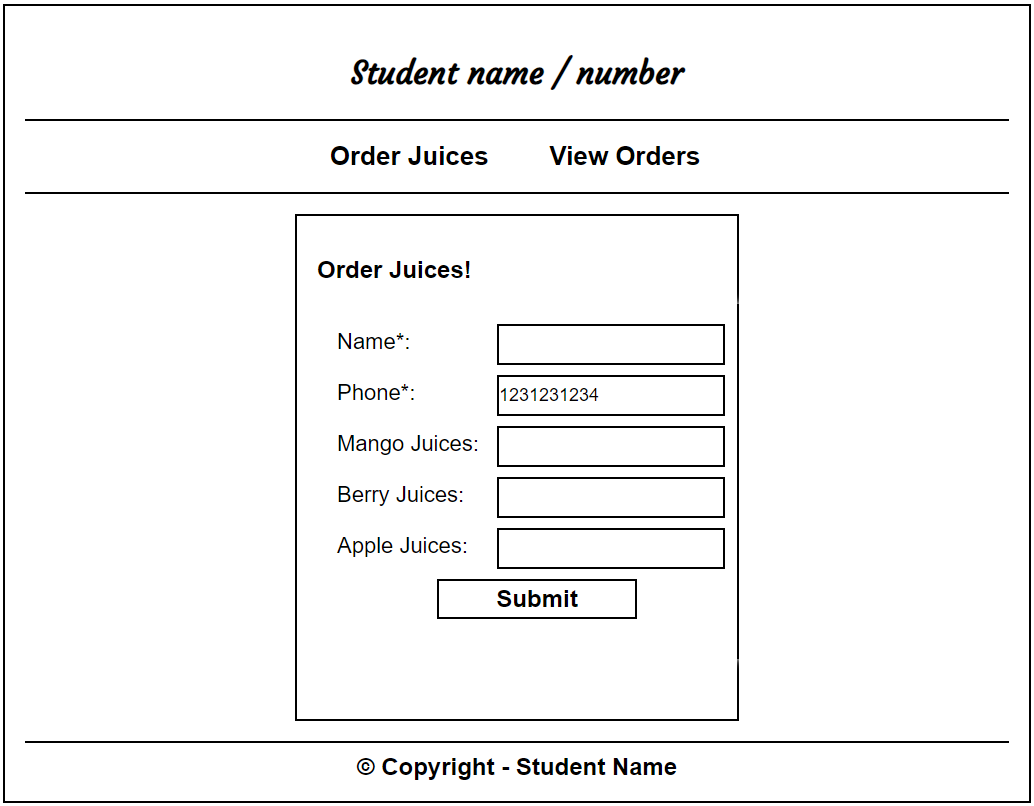
**Rubrics:**

|  |  |
| --- | --- |
| **Item** | **Marks** |
| Basic HTML and CSS for the 2 pages | 4 |
| View Orders | 8 |
| Create Orders | 8 |
| Validation for Name | 2 |
| Validation for Phone | 4 |
| Programming logic, standards, application completeness | 2 |
| Git Repo with commits in your name | 2 |
| **Total** | **30** |

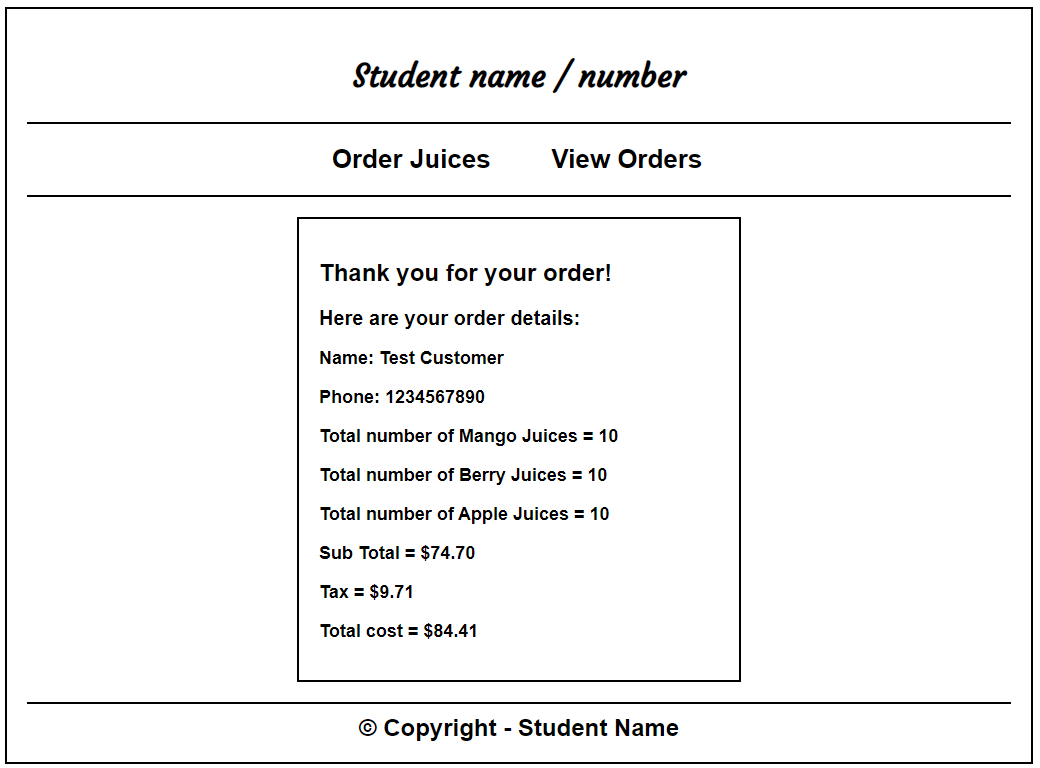
***Please turn over***

**Wireframes:**

1. **Order form page:**

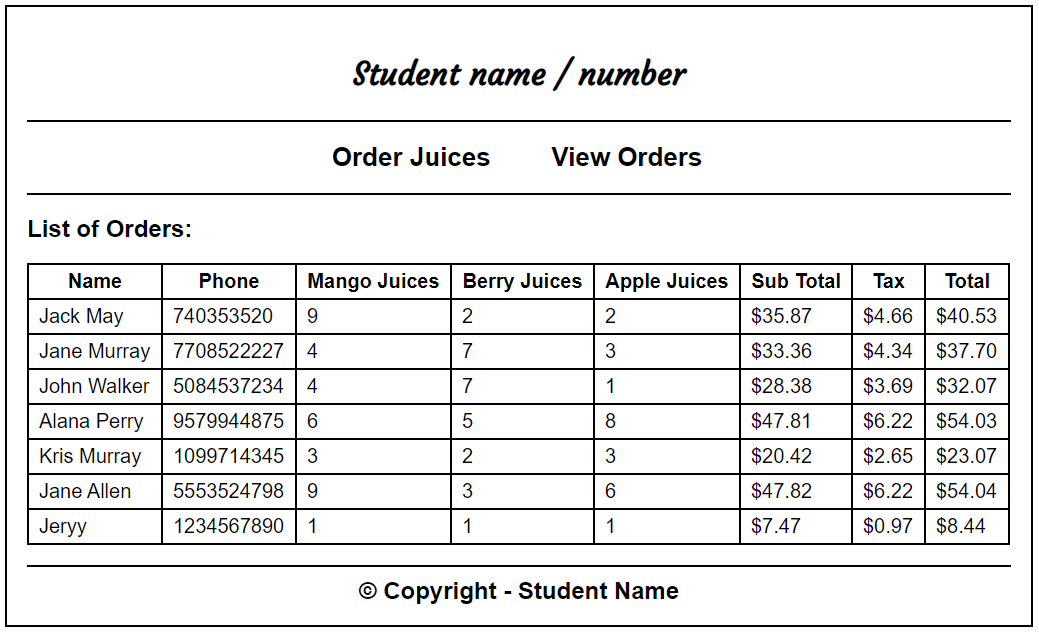


1. **Order Success/Thank you page:**

****

***Please turn over***

1. **View Orders:**

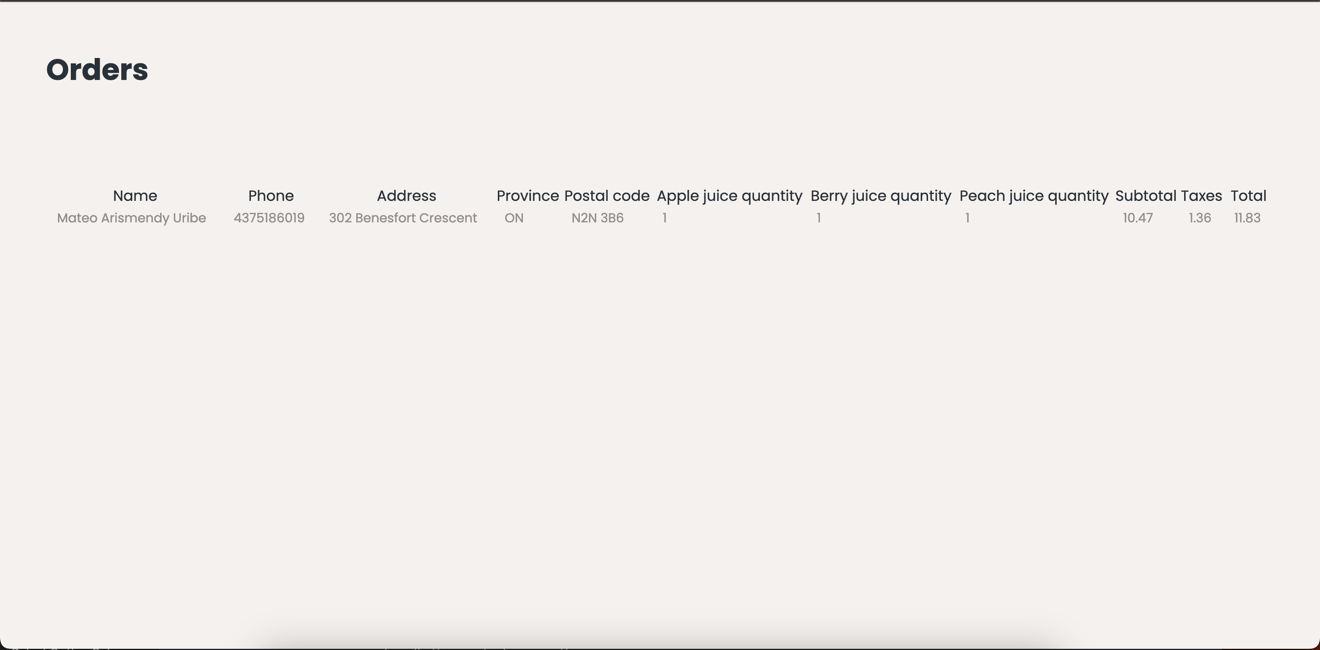
****

**A screenshot of a menu

Description automatically generated**

**A white screen with a couple of squares

Description automatically generated with medium confidence**

****