Challenge

Develop Interactive Web Pages Using DOM And DOM Events

Challenge

- Create an order form and save the order details for the Veggie Pizza outlet

Points to Remember

- Use arrays as in-memory storage for menu, customers and orders.
- Use the `parseInt()` global function to convert string value obtained from form field inputs to integer type value, so that integer manipulation can be done.
- Apply form validation using HTML5 built-in validations for static as well as dynamically added form field elements.
- The JavaScript code should not be embedded within **index.html** file, but instead should be located externally in a separate **.js** file.
- Use `addEventListener()` and `removeEventListener()` methods to add events and disable events.

Instructions for Challenge

- Click here for the boilerplate.
- Read the README.md file available in the boilerplate for further instructions about the challenge.
- Fork the boilerplate into your own workspace.
- Clone the boilerplate into your local System and open the folder having the boilerplate in VS Code.
- Open command terminal and run the command **npm install** to install Mocha and Chai dependencies.
- Write code inside **index.html** as per the requirements specified in the upcoming slides.
- Write CSS code to style the web page located inside the **css/styles.css** of the boilerplate.
- Write JavaScript code as per the requirements stated in the upcoming slides in **js/script.js** file of the boilerplate.
- Use the <script> tag in **index.html** file to refer to the **script.js** file.
- Open the **index.html** file using Live Server and test the output.
- Submit the file for evaluation.

Create an Order Form and Save the Order Details for the Veggie Pizza Outlet

Veggie Pizza is a pizza delivery outlet that became popular for its custom-made vegetarian pizzas. A customer can order pizzas and other items by calling the outlet. The outlet delivers the order to the address specified by each customer.

The outlet now has an app that allows the call operator to enter the order details and store them.

Write a JS program to create an order form with validations, to add order items dynamically, to save the order details and calculate the total bill amount payable.

CHALLENGE

Tasks

- The challenge can be performed by the following steps:
  - Step 1: Create Order Form With Validations
  - Step 2: Add Order Items Dynamically
  - Step 3: Calculate Total Bill Amount
  - Step 4: Save the Order with a Successful Message

Note: Details about these steps are given in the upcoming slide.

Step 1 – Create Order Form With Validations

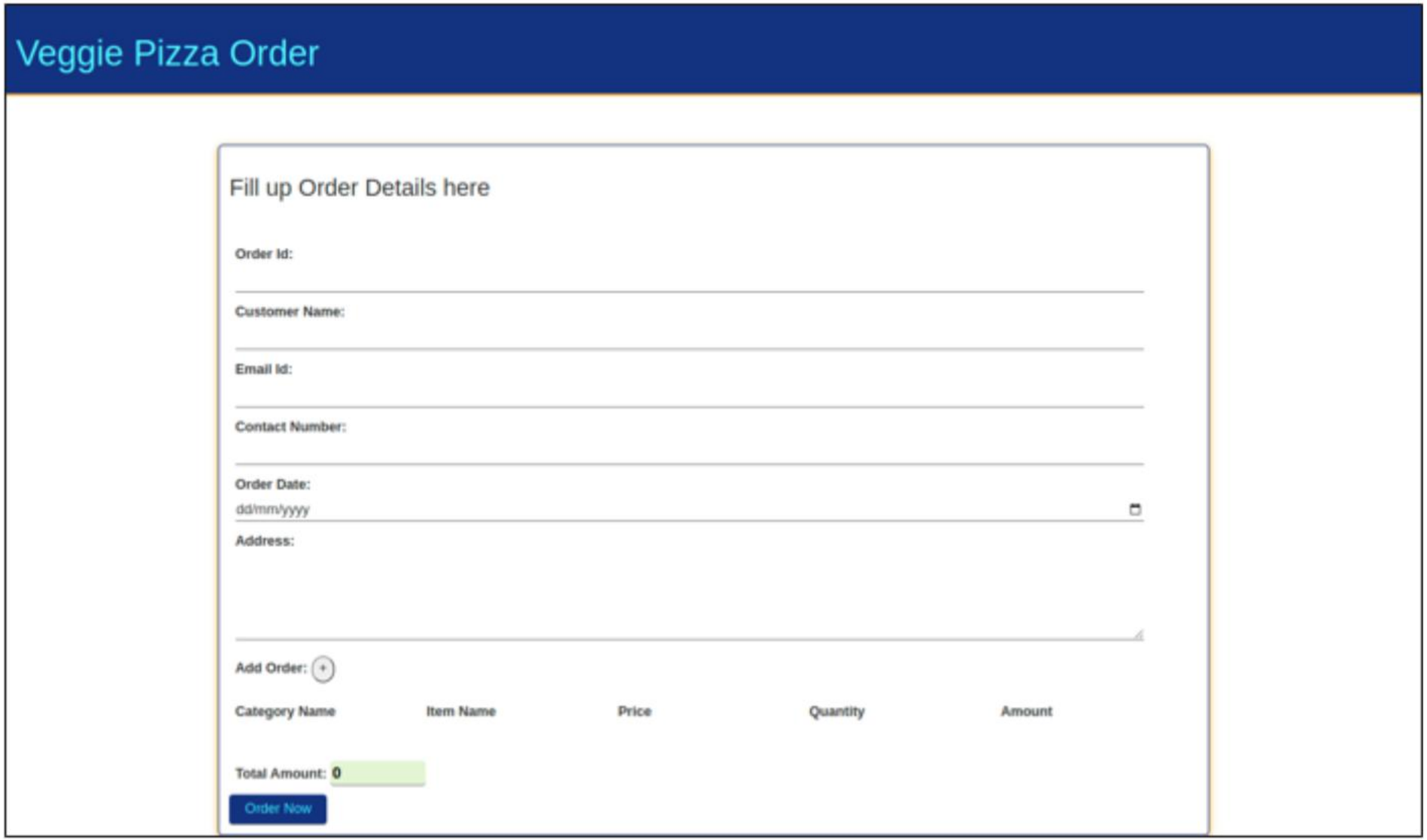
- Open **index.html** file inside **solution** folder to create an HTML form.
- An order form should be created with the fields and validations as listed in the table below.
- Error messages should be displayed for the field inputs that do not fulfil the validation criteria. Forms with invalid inputs should not get submitted.

Input Field	Validation
Order Id	Should not be left blank and should be of type number
Order Date	Should be in correct date format
Name	Should not be left blank
Email Id	Should not be left blank and accept valid email value
Contact Number	Should not be left blank and takes only 10 digitted number
Address	Should not be left blank and should have minimum 10 characters
Order Amount	Should be non-editable and get updated as order items are added

Dynamically Added Form Fields

Input Field	Validation
Category	Should not be left blank
Item Name	Should not be left blank
Price	Should not be left blank and should allow only numbers. Default value should be 0.
Quantity	Should not be left blank and should allow only numbers. Default value should be 0.
Amount	Should be non-editable and get updated as price or quantity are updated. Default value should be 0.

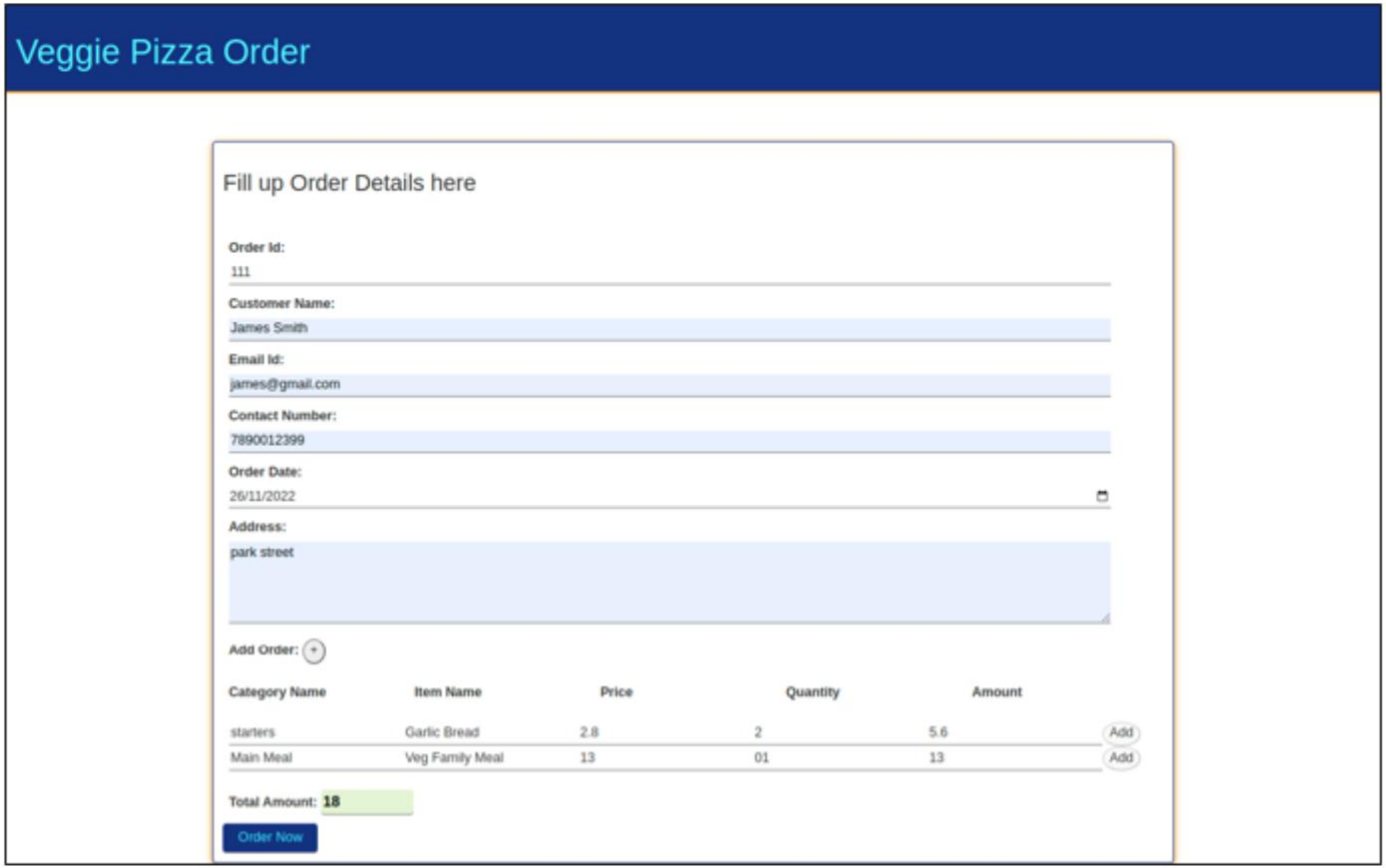
Expected Output: Form



Step 2 – Add Order Items Dynamically

- Inside **index.html file**, add a button with the “+” icon in the order form.
- Open the **script.js** file inside the **solution/js** folder to add JavaScript code.
- Write a JS function such that when the user clicks the "+" button, input fields should get added dynamically, allowing the user to input order item details such as:
  - Category – should be an input field that allows the user to input category.
  - Item name – should be an input field that allows the user to input menu item.
  - Price – should be an input field that allows only numbers to be inputted.
  - Quantity – should be an input field that allows only numbers to be inputted.
  - Amount – should be a read-only field that displays the total amount whenever the price or quantity is updated.
  - Add button - Associate with click event listener programmatically to handle the click event.

Expected Output: Form With Order Item Details



Step 3 – Calculate Total Bill Amount

- As the user enters order items and click the "add" button,
  - The text field for “Total Bill Amount” should get updated with the sum of the amount of all order items.
  - Save the ordered items in an array
  - Make all fields read-only
  - Disable the active state of the "add" button

Note: The total amount field should not be editable and should display the value as "0" initially.

Step 4 – Save Order With a Successful Message

- Create a JavaScript function to save the order details.
- Upon clicking the “Order Now” button, capture the order details and store it in an array.
- Confirm the successful completion of the transaction by returning the following message :

```
Total amount to be paid: <totalAmount>
```

For Reference: Bill Details