Full Stack Development with AI
Lab 5.6 – Manipulating DOM with JavaScript

Lab Overview

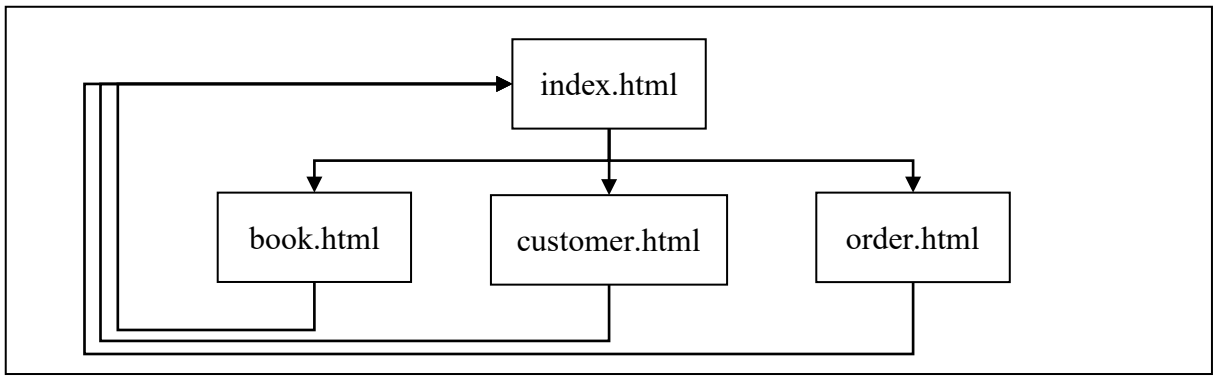
In this lab, you will practise how to manipulate DOM with JavaScript using the Online Bookstore case study. The Online Bookstore case study simulates a typical web application that performs data manipulation operations such as create, retrieve, update and delete (CRUD) using a relational database.

This lab assumes that you are familiar with basic web authoring using HTML and CSS.

Exercise 1 – Understanding Static HTML

Before you can manipulate DOM with JavaScript to create dynamic HTML, it is important to first understand how static HTML works. This exercise walks you through the static HTML web pages for the Online Bookstore case study. Whenever appropriate, HTML5 elements are used to improve the user interface of the web pages.

The sitemap of the web application is shown below:



The actual folder structure of the web pages is listed below for clarity:

- (root folder)
  - images subfolder
    - onlinebookstore.jpg
  - index.html
  - book.html
  - customer.html
  - order.html

All the HTML documents can be found in `static.zip`.

Minimal CSS styling has been applied to the static HTML for simplicity. In other words, the web pages will be rendered in the default black and white.

The `index.html` is the default or home page of the web application. The screenshot of the web page is provided below. The bookstore image can be found inside the `images` subfolder.

Welcome to Online Bookstore

Main Menu

- Book
- Customer
- Order

The `book.html` is the web page that is used to perform CRUD operations for the book records. The screenshot of the web page is provided below.

Observe that the "Create/Update Book" form uses appropriate HTML input elements for the respective attributes. The "View All Books" table shows five rows of records for simplicity. The "Update" and "Delete" hyperlinks are placeholders and link to `href="#"`.

Book 🕮

Create/Update Book

Bid	Enter book id
Bname	Enter book name
Bpd	dd / mm / yyyy
Bcountry	USA
Bprice	

Clear Create/Update

View All Books

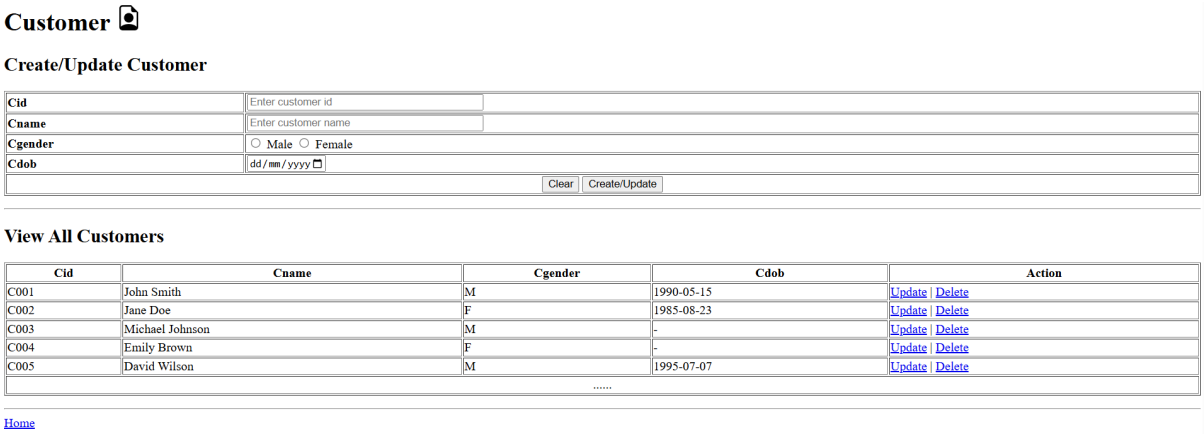
Bid	Bname	Bpd	Bcountry	Bprice	Action
B001	The Great Gatsby	2022-06-15	USA	$15.99	Update | Delete
B002	To Kill a Mockingbird	2022-07-05	USA	$12.50	Update | Delete
B003	Pride and Prejudice	2022-06-30	England	$10.99	Update | Delete
B004	1984	2022-08-10	UK	$14.75	Update | Delete
B005	The Catcher in the Rye	2022-07-20	USA	$13.25	Update | Delete

.....

Home

The `customer.html` is the web page that is used to perform CRUD operations for the customer records. The screenshot of the web page is provided on the next page.

Observe that the "Create/Update Customer" form also uses appropriate HTML input elements for the respective attributes.



Customer

Create/Update Customer

Cid	Enter customer id
Cname	Enter customer name
Cgender	Male Female
Cdob	dd/mm/yyyy

Clear Create/Update

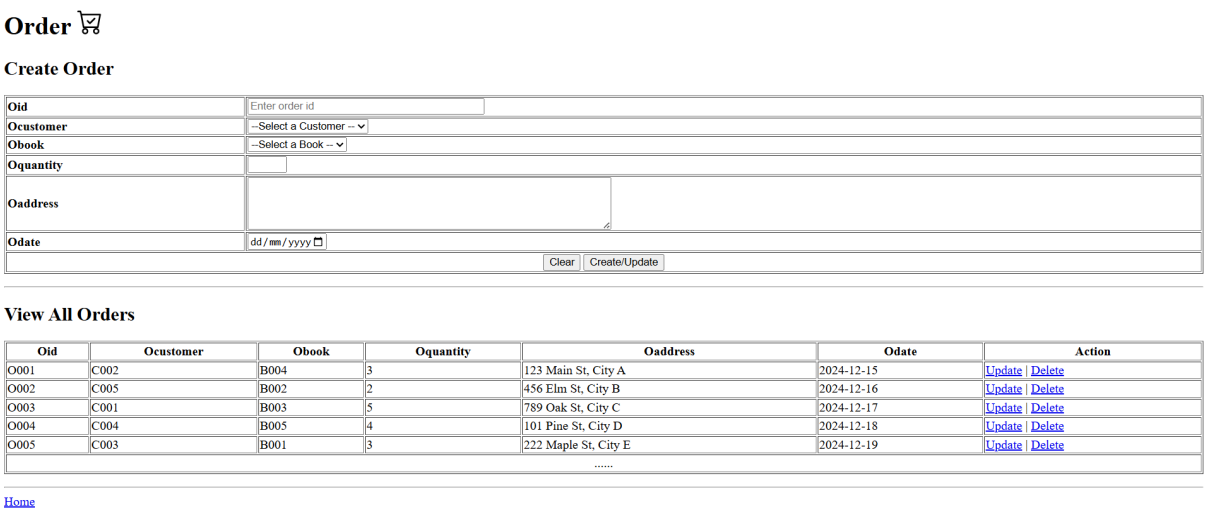
View All Customers

Cid	Cname	Cgender	Cdob	Action
C001	John Smith	M	1990-05-15	Update | Delete
C002	Jane Doe	F	1985-08-23	Update | Delete
C003	Michael Johnson	M	-	Update | Delete
C004	Emily Brown	F	-	Update | Delete
C005	David Wilson	M	1995-07-07	Update | Delete

.....

Home

Finally, the `order.html` is the web page that is used to perform CRUD operations for the order records. The screenshot of the web page is provided below.



Order

Create Order

Oid	Enter order id
Ocustomer	--Select a Customer --
Obook	--Select a Book --
Oquantity	
Oaddress	
Odate	dd/mm/yyyy

Clear Create/Update

View All Orders

Oid	Ocustomer	Obook	Oquantity	Oaddress	Odate	Action
O001	C002	B004	3	123 Main St, City A	2024-12-15	Update | Delete
O002	C005	B002	2	456 Elm St, City B	2024-12-16	Update | Delete
O003	C001	B003	5	789 Oak St, City C	2024-12-17	Update | Delete
O004	C004	B005	4	101 Pine St, City D	2024-12-18	Update | Delete
O005	C003	B001	3	222 Maple St, City E	2024-12-19	Update | Delete

.....

Home

Study the static HTML in the web pages carefully and ensure that you understand how they are being rendered by the web browser before proceeding to Exercise 2.

Exercise 2 – Manipulating DOM with JavaScript

For this exercise, all the required HTML documents and JavaScript files can be found in `dynamic.zip`.

In this exercise, you will be replacing the static HTML with dynamic HTML by manipulating the DOM with JavaScript.

Perform the following tasks to convert `dynamic/book.html` from static HTML to Dynamic HTML:

1. Open `dynamic/book.html` and inspect the static HTML. Can you identify the main differences between this version and the version in `static`?

2. Open `dynamic/scripts/book.js` for editing. Define a JavaScript array of object `books` containing the following data:

   - 'B001', 'The Great Gatsby', '2022-06-15', 'USA', 15.99
   - 'B002', 'To Kill a Mockingbird', '2022-07-05', 'USA', 12.50
   - 'B003', 'Pride and Prejudice', '2022-06-30', 'England', 10.99
   - 'B004', '1984', '2022-08-10', 'UK', 14.75
   - 'B005', 'The Catcher in the Rye', '2022-07-20', 'USA', 13.25

   Each book record should contain the following five attributes:

   - Bid
   - Bname
   - Bpd
   - Bcountry
   - Bprice

3. Write a JavaScript function `renderBooksTable()` that uses the `books` array of object to render the data in the `tableBook` HTML table.

   Render the Update and Delete action links with anchor tags:

   - Update – The `href` attribute should be set to `"#"` and the `onclick` event attribute should be set to call a JavaScript function `updateBook` that takes in Bid of the book as an input parameter.

   - Delete – The `href` attribute should be set to `"#"` and the `onclick` event attribute should be set to call a JavaScript function `deleteBook` that takes in Bid of the book as an input parameter.

   Call this function in the `onload` event attribute of the `body` tag. Open `dynamic/book.html` in your web browser. Can you see the data in the HTML table?

4. Write a JavaScript function `createUpdateBook()` to retrieve the values inputted by the user in the HTML form.

   By default, the edit mode should be "create". That is, insert a new object into `books` using the retrieved values. Then render the HTML table again to display the updated `books` data with the new record.

   You should call this function in the `onsubmit` event attribute of the `form` tag in order to trigger the HTML5 input validation. However, since we do not have a backend at this juncture, we should cancel the form submission after rendering the HTML table.

   The correct syntax should be `onsubmit="return createUpdateBook()"` and in the `createUpdateBook ()` function, you should `return false`.

5. Write a JavaScript function `updateBook()` to populate the HTML form with the values of the corresponding book record obtained from `books`. This would allow the user to edit the required values.

   Should you allow the user to edit Bid? If not, how should you prevent the user from editing Bid?

   Change edit mode to "update" and modify the `createUpdateBook()` function such that when user next click the "Create/Update" button, the corresponding book record should be updated with the new values edited by the user.

   Remember to render the HTML table again to display the updated `books` data with the edited record.

   You should also reset the edit mode back to "create".

6. Write a JavaScript function `deleteBook()` to confirm with the user whether to delete the selected book. If user clicks "OK", remove the corresponding book record from `books`.

   Remember to render the HTML table again to display the updated `books` data without the deleted record.

7. When you are with all the above tasks, test the dynamic web page in your web browser and ensure that all the CRUD operations are working correctly.

-- End of Lab --