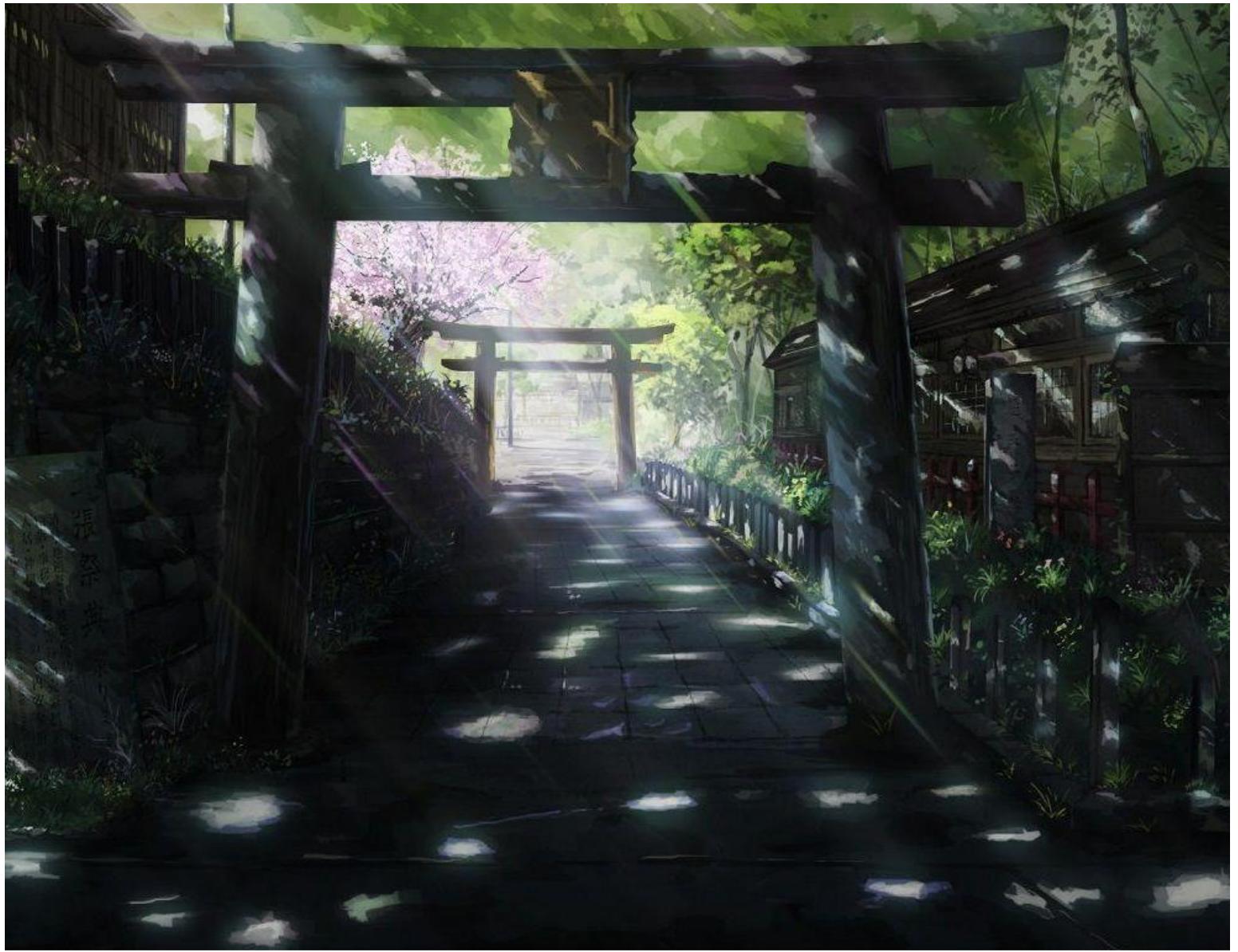


MANGA MAPS REPORT



SULTAN QABOOS UNIVERSITY DEPARTMENT OF COMPUTER SCIENCE

COMP3700-INTRODUCTION TO WEB COMPUTING - FALL 2025 Instructor: Dr. Mohammed Al-Soleimani

s142471 - Loai Khalfan Al-Ghassani

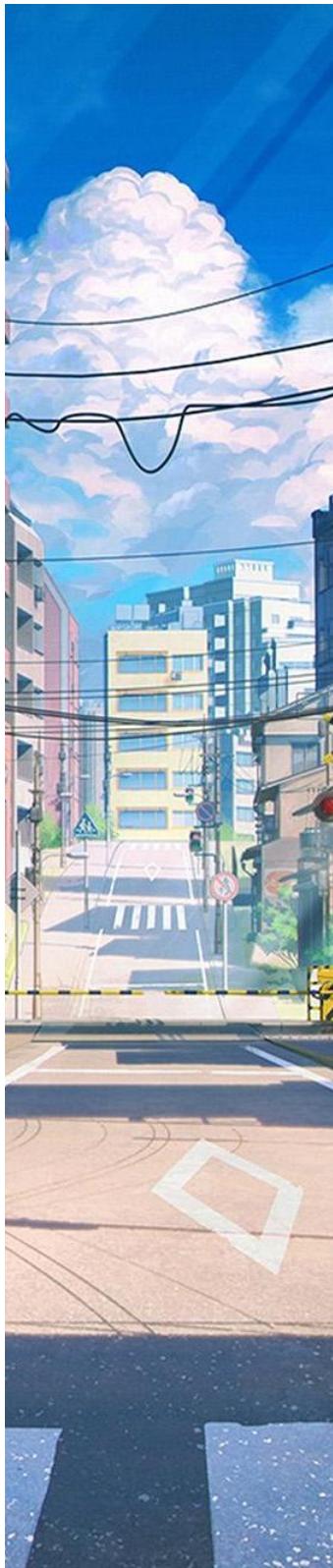
s140056 - Mohammed Ahmed Al-Jabri



TABLE OF CONTENTS

1. Overview
 2. Bootstrap Integration
 3. JavaScript Implementation
 4. Team Collaboration
 5. Screenshots
- 





1. OVERVIEW

In this third phase of the MangaMaps project, our team successfully migrated the static website to the **Bootstrap Framework** and implemented **Dynamic JavaScript** functionality. The website now adheres to modern responsive design standards and features interactive elements such as a cost calculator, a quiz game, and dynamic data tables.

The project now consists of **12 pages**, including the newly added `calculator.html` , `funPage.html` , `questionnaire.html` , and `dynamic-tables.html`.

2. BOOTSTRAP INTEGRATION

We removed all previous custom CSS rules and exclusively used Bootstrap 5.3.2 classes to ensure a consistent, responsive layout across all devices.

A. Grid System & Layout

We structured all pages using Bootstrap's responsive grid system (container, row, col-*).

- **Implementation:** On the **Tours** and **Fun Page**, we used col-12 col-md-6 col-lg-4 to create cards that stack vertically on mobile screens but display in a 3-column grid on desktop screens.

B. Forms & Validation

We applied Bootstrap styling to all form elements (form-control, form-select, form-check) to enhance usability.

- **Feedback:** We implemented Bootstrap's validation classes (is-invalid, invalid-feedback) to provide clear visual cues (red borders and text) when user input is incorrect.

C. Navigation & Components

- **Navbar:** A responsive navigation bar (`navbar-expand-lg`) was added to every page, featuring a hamburger menu for mobile users.
- **Components:** We integrated a **Carousel** for the image gallery on the Index page and **Modals** on the Fun Page to display game results dynamically.

3. JAVASCRIPT IMPLEMENTATION

A. Dynamic Content (Objects & Arrays)

We updated the table pages to use JavaScript objects and arrays.

- **Object Constructors:** We defined two constructors, Tour() and Review(), in dynamic.js to standardize data.
- **Arrays:** Data is stored in tours[] and reviews[] arrays.
- **Dynamic Loading:** The loadToursTable() function dynamically generates HTML <tr> rows based on the array data.
- **Add Functionality:** A form allows users to input new data, which is pushed to the array and immediately rendered in the table using the addTour() function.

B. Questionnaire Page (Form Validation)

We created a feedback form satisfying the requirement for 5+ controls and custom validation.

- **Validation Logic:**
 - **Name:** Uses Regex to allow only letters.
 - **Age:** Ensures the input is between 10 and 100.
 - **Comments:** Checks for a minimum length of 15 characters.
 - **Selection:** Validates that at least one radio button and one checkbox are selected before submission.

C. Calculator Page

We developed a "Tour Cost Calculator" to estimate trip expenses.

- **Logic:** The script calculates (Base Price × Travelers).
- **Conditions:**
 - If travelers > 4, a **10% Group Discount** is applied.
 - If the user enters the code #MANGAWINNER (won from the Fun Page), a **40% Discount** is applied.
- **Output:** The script dynamically updates the DOM to show the Base Cost, Discount, Tax (5%), and Final Total.

D. Fun Page (Interactive Game)

We included a "Japan Landmarks Quiz" where users match images to names.

- **Dynamic HTML:** JavaScript generates the game cards from an array of image objects.
- **Logic:** A checking function compares user selection against the correct answer stored in the object.
- **Reward:** A **Bootstrap Modal** displays the score and provides the discount code if the user answers all questions correctly.

E. Index Page Updates

- **Banner:** A scrolling text banner displays a welcome message with the current live date and time using JavaScript's Date object.
- **Gallery:** A rotating photo gallery (Bootstrap Carousel) cycles through images every 3 seconds.

4. TEAM COLLABORATION

GitHub Repository: https://github.com/yohikami/COMP3700_Project.git

Task	Team Member	%Contribution	Description of Work
Bootstrap Layout	Loai	100%	Migrated all pages to Grid system; styled Navbar and Footer.
Dynamic Tables (JS)	Loai	100%	Created dynamic.js, Object Constructors, and Array logic.
Questionnaire Page	Mohammed	100%	Designed form and implemented Regex validation logic.
Calculator Page	Loai	100%	Implemented math calculations and conditional discount logic.
Fun Page (Game)	Mohammed	100%	Created Quiz logic, array shuffling, and Modal integration.
Report & Testing	Loai & Mohammed	40% / 60%	Compiled report, managed GitHub repo, and tested features.

5. SCREENSHOTS

5.1 Calculator Page

The screenshot shows a web browser window for the 'Tour Cost Calculator - MangaMaps' page. The URL in the address bar is C:/Users/YTMvw/OneDrive/Documents/GitHub/COMP3700_Project/calculator.html. The page has a dark header with the MangaMaps logo and navigation links: Home, Tours, **Calculator**, Booking, Reviews, Questionnaire, About, Contact, and Login. A red banner at the top says 'Tour Cost Calculator' and 'Estimate your Japan tour cost with tax and discounts'. Below the banner is a form titled 'Calculate Your Total' with three input fields: 'Select Tour' (dropdown menu showing 'Tokyo City Lights – \$99 per person'), 'Number of People' (text input field), and 'Discount Code (optional)' (text input field). A red 'Calculate Total' button is at the bottom of the form. At the bottom of the page is a dark footer with the MangaMaps logo, a copyright notice ('© 2025 MangaMaps. All rights reserved.'), and a 'Back to top ↑' link.

5.2 Dynamic Tables

The screenshot shows a web application titled "Dynamic Data Tables". The page is divided into several sections:

- Tours Table:** A table with columns for Name, Price, and Duration. It contains three rows: Tokyo City Lights (Price: 99, Duration: 2 days), Osaka Food Adventure (Price: 50, Duration: 2 days), and Kyoto Heritage Escape (Price: 89, Duration: 3 days).
- Reviews Table:** A table with columns for Name, Rating, and Comment. It contains five rows:
 - Sarah Johnson: Rating 5, Comment: Our Tokyo City Lights tour was absolutely incredible!
 - Michael Chen: Rating 5, Comment: The Osaka Food Adventure was a culinary delight!
 - Emma Williams: Rating 4, Comment: Kyoto Heritage Escape was peaceful and beautiful.
 - David Brown: Rating 5, Comment: MangaMaps made our first trip to Japan very easy.
- Add New Tour:** A form with fields for Tour Name, Price, and Duration, followed by a red "Add Tour" button.
- Add New Review:** A form with fields for Name, Rating (1-5), and Comment, followed by a red "Add Review" button.
- Footer:** A dark footer bar containing the MangaMaps logo and copyright information: "MangaMaps © 2023 MangaMaps. All rights reserved. Back to top ↑".

5.3 Fun Page (Showing Quiz Cards)

Japan Landmarks Quiz - Manga

C:/Users/YTMvv/OneDrive/Documents/GitHub/COMP3700_Project/funPage.html

MangaMaps

Home Tours Calculator Booking Reviews Questionnaire Quiz Game About Contact | Login

Japan Landmarks Quiz Game

Test your knowledge of famous Japanese landmarks!

How to Play
Look at each image and select the correct name of the famous Japanese landmark. Answer all 6 questions correctly to win a discount code!

Question 1
What is the name of this famous Japanese landmark?
-- Select Answer --

Question 2
What is the name of this famous Japanese landmark?
-- Select Answer --

Question 3
What is the name of this famous Japanese landmark?
-- Select Answer --

Question 4
What is the name of this famous Japanese landmark?
-- Select Answer --

Question 5
What is the name of this famous Japanese landmark?
-- Select Answer --

Question 6
What is the name of this famous Japanese landmark?
-- Select Answer --

Check Answers

The screenshot shows a web browser window with the title 'Japan Landmarks Quiz - Manga' and the URL 'C:/Users/YTMvv/OneDrive/Documents/GitHub/COMP3700_Project/funPage.html'. The page has a red header bar with the title 'Japan Landmarks Quiz Game' and the subtitle 'Test your knowledge of famous Japanese landmarks!'. Below the header is a light blue box containing the text 'How to Play' and 'Look at each image and select the correct name of the famous Japanese landmark. Answer all 6 questions correctly to win a discount code!'. The main content area contains six questions, each with an image and a dropdown menu for the answer. Question 1 shows Mount Fuji with cherry blossoms. Question 2 shows a red torii gate. Question 3 shows a traditional Japanese castle. Question 4 shows a city skyline with the Tokyo Tower. Question 5 shows a busy street intersection. Question 6 shows a bamboo forest. At the bottom is a red button labeled 'Check Answers'.

5.4 Questionnaire

The screenshot shows a web browser window with the title "Questionnaire - MangaMaps". The URL in the address bar is "C:/Users/YTMvv/OneDrive/Documents/GitHub/COMP3700_Project/questionnaire.html". The page itself is titled "Travel Questionnaire" and has a red header bar with the text "Tell us about your travel preferences". Below the header is a white "Survey Form" section containing various input fields and questions:

- Survey Form**
- Full Name ***: Input field with placeholder "Your full name".
- Email ***: Input field with placeholder "your.email@example.com".
- Age ***: Input field with placeholder "Your age".
- How interested are you in visiting Japan? ***: Radio buttons for "Low", "Medium", and "High". A note below says "Please select one option."
- Which activities are you interested in? ***: Checkboxes for "Food", "Nature", "Culture", and "Shopping". A note below says "Select at least one activity."
- Which city are you most excited to visit?**: A dropdown menu with the placeholder "-- Select a city --".
- Budget Range (USD)**: A horizontal slider with a blue dot indicating the range.
- Additional Comments ***: A text area with placeholder "Write at least 15 characters...".
- Submit Questionnaire**: A red button at the bottom of the form.

5.5 Commented JavaScript Code

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing a project structure with files like funPage.html, questionnaire.html, and dynamic.js. The main editor area displays the content of the dynamic.js file. The file contains several functions, notably loadTourTable and addTour, which handle the creation and manipulation of a tour table. The code is heavily annotated with multi-line comments explaining its purpose and logic. A GitHub Copilot suggestion box is visible in the bottom right corner, prompting the user to "give me screenshot of dynamic.js". The status bar at the bottom provides build information and other tool settings.

```
File Edit Selection View Go Run Terminal Help COMP3700 Project ... JS dynamic.js CHAT give me screenshot of dynamic.js You need to set up GitHub Copilot and be signed in to use Chat. ... + - x File Edit Selection View Go Run Terminal Help COMP3700 Project ... JS dynamic.js CHAT give me screenshot of dynamic.js You need to set up GitHub Copilot and be signed in to use Chat. ... + - x
```

```
dynamic.js > ...
function loadTourTable(data) {
  data.forEach(t => {
    let row = document.createElement('tr');
    row.innerHTML = ` ${t.name} | ${t.price} | ${t.duration} |`;
    document.querySelector('tbody').append(row);
  });
}

// Load tours table when page starts
loadTourTable(tours);

***** * ADD NEW TOUR
* -----
* This function reads the user's input from the
* three text fields, validates that all fields
* are filled, then creates a new Tour object and
* adds it to the tours array.
*
* After updating the array, it refreshes the table
* and clears the input boxes.
*****
function addTour() {
  let name = document.getElementById("newName").value;
  let price = document.getElementById("newPrice").value;
  let duration = document.getElementById("newDuration").value;

  // Basic validation: all fields must be filled
  if (!name || !price || !duration) {
    alert("Please fill all fields.");
    return;
  }

  // Create new object and push to array
  tours.push(new Tour(name, price, duration));

  // Refresh table display
  loadTourTable(tours);

  // Clear inputs after adding
  document.getElementById("newName").value = "";
  document.getElementById("newPrice").value = "";
  document.getElementById("newDuration").value = "";
}

*****
```

Line 32, Col 3 Spaces: 4 UTF-8 CRLF JavaScript Signed out Port: 3000

File Edit Selection View Go Run Terminal Help

funPage.html questionnaire.html JS dynamicjs

```
javascript > JS dynamicjs > ...
93  ****
94  * REVIEW OBJECT (Constructor Function)
95  *
96  * Similar to Tour, this constructor defines a
97  * Review object with name, rating, and comment.
98  ****
99  function Review(name, rating, comment) {
100    this.name = name;
101    this.rating = rating;
102    this.comment = comment;
103  }
104
105  ****
106  * REVIEWS ARRAY (Initial Data)
107  *
108  * Pre-filled list of reviews that appear when the
109  * user opens the page. Each item uses the Review
110  * constructor and stores user-like feedback.
111  ****
112 let reviews = [
113   new Review(
114     "Sarah Johnson",
115     5,
116     "Our Tokyo City Lights tour was absolutely incredible!"
117   ),
118   new Review(
119     "Michael Chen",
120     5,
121     "The Osaka Food Adventure was a culinary delight!"
122   ),
123   new Review(
124     "Emma Williams",
125     4,
126     "Kyoto Heritage Escape was peaceful and beautiful."
127   ),
128   new Review(
129     "David Brown",
130     5,
131     "MangaMaps made our first trip to Japan very easy."
132   )
133 ];
134
135  ****
136  * DISPLAY REVIEWS TABLE
137  *
138  * Works the same way as loadTourTable, but for
139  * Review objects. It loops through the reviews
140  * array and builds table rows dynamically.
```

give me screenshot of dynamicjs

You need to set up GitHub Copilot and be signed in to use Chat.

CHAT

Ln 32, Col 3 Spaces: 4 UTF-8 CRLF JavaScript Signed out Port: 3000

File Edit Selection View Go Run Terminal Help

funPage.html questionnaire.html JS dynamicjs

```
javascript > JS dynamicjs > ...
140  * array and builds table rows dynamically.
141  ****
142 function loadReviewTable(data) {
143   let tbody = document.querySelector("#reviewsTable tbody");
144   tbody.innerHTML = "";
145
146   data.forEach(r => {
147     tbody.innerHTML += `
148       <tr>
149         <td>${r.name}</td>
150         <td>${r.rating}</td>
151         <td>${r.comment}</td>
152       </tr>
153     `;
154   });
155
156  // Load reviews table when page starts
157  loadReviewTable(reviews);
158
159  ****
160  * ADD NEW REVIEW
161  *
162  * Similar to addTour(), this function:
163  * - Reads inputs
164  * - Validates entries
165  * - Creates a new Review object
166  * - Pushes it to the array
167  * - Reloads the table
168  * - Clears form fields afterwards
169  ****
170 function addReview() {
171   let name = document.getElementById("newNameR").value;
172   let rating = document.getElementById("newRatingR").value;
173   let comment = document.getElementById("newCommentR").value;
174
175   // Validate: all fields required
176   if (!name || !rating || !comment) {
177     alert("Please fill all fields.");
178     return;
179   }
180
181   // Add new review
182   reviews.push(new Review(name, rating, comment));
183
184   // Refresh reviews table
185   loadReviewTable(reviews);
186
187  // Clear form fields
188}
```

give me screenshot of dynamicjs

You need to set up GitHub Copilot and be signed in to use Chat.

CHAT

Ln 32, Col 3 Spaces: 4 UTF-8 CRLF JavaScript Signed out Port: 3000

File Edit Selection View Go Run Terminal Help

funPage.html questionnaire.html JS dynamicjs

```

156 // Load reviews table when page starts
157 loadReviewTable(reviews);
158
159
160 ****
161 * ADD NEW REVIEW
162 *
163 * Similar to addTour(), this function:
164 * - Reads inputs
165 * - Validates entries
166 * - Creates a new Review object
167 * - Pushes it to the array
168 * - Reloads the table
169 * - Clears form fields afterwards
170 ****
171 function addReview() {
172     let name = document.getElementById("newNameR").value;
173     let rating = document.getElementById("newRatingR").value;
174     let comment = document.getElementById("newCommentR").value;
175
176     // Validate: all fields required
177     if (!name || !rating || !comment) {
178         alert("Please fill all fields.");
179         return;
180     }
181
182     // Add new review
183     reviews.push(new Review(name, rating, comment));
184
185     // Refresh reviews table
186     loadReviewTable(reviews);
187
188     // Clear input fields
189     document.getElementById("newNameR").value = "";
190     document.getElementById("newRatingR").value = "";
191     document.getElementById("newCommentR").value = "";
192 }
193

```

give me screenshot of dynamicjs

You need to set up GitHub Copilot and be signed in to use Chat.

CHAT

Ln 32, Col 3 Spaces: 4 UTF-8 CRLF JavaScript Signed out Port: 3000

