worldskillsuk

WSUK – National Finals Competition

Web Development

2025

MODULE C

FRONTEND DEVELOPMENT – IPT

Val Adamescu
Lewis Newton
Mark Kiss

Contents

Introduction

In this module you are going to develop the public facing website for Spark Studios. The platform showcases pictures taken on a wide range of events and make them available for purchase in various sizes.

While public events can be browsed by anyone, private events can only be viewed and ordered by authenticate and authorised users. (A pre-determined event access list can be found in the end of this document.)

Your solution can be developed using one of the following frontend technologies: *React*, *Vue*, *Svelte*, *Next.js*, or *vanilla JavaScript*. You may use CSS frameworks like *Tailwind* or *Bootstrap* for styling. Please note that any submission using a backend language or framework (e.g., *PHP*, *Laravel*, *Express*) will not be accepted for marking.

Note: *Please select a framework you are proficient in. Your submission will be evaluated as-is, and it is your responsibility to ensure the application is fully functional and accessible. The judges will not debug, modify, or troubleshoot your code.*

***NOTE:** The specific submission/upload requirements of the test project are provided separately.*

You have three (3) hours to complete this task. You need to submit it before time runs out. No additional time will be given for submission.

API

You are given a pre-developed API, that gives you access to various services. The API documentations are available as:

- Api Documentation: ***dist/api/README.md***
- OpenAPI specs: ***dist/api/api.yaml***
- Database schema: ***database/module_c_db_schema.jpg***
- SQL file: ***database/db.sql***

The recommended way to seed the database is by making a ***POST*** request to the ***/db/reset*** endpoint. An SQL file is also provided as a backup, but you should not need it.

To visually access the API documentation and specs you need to use the following plugins:

Visual Studio Code

- Open any ***.md*** file and on top right select "*Markdown Preview Enhanced*"
- Open ***api/api.yaml*** and *CTRL+SHIFT+P* **->** *Preview Swagger*

PHP Storm

- *Default available for* ***.md*** *and* ***.yaml*** *files*

Requirements

★ Design - UI/UX

While the design is not the main scope of this task, given this is the customer facing site, you are *required* to create an appealing layout and design, that is clean, well-organised, and easy to use both on desktop and mobile by customers.

Some ***design guidance has been provided*** (***dist/assets/mock-ups***) by the client, for each important section, please pay attention and follow them as requested. Where stated, you need to go by the design and functionality requirements. On every page, the page title should be consistently "Spark Studios" and the favicon (***dist/assets/images/favicon***).

A CSS pattern file with the brand colours is provided in ***dist/assets/css/pattern.css***

- **Page Titles:** Must be centred and use the ***primary light*** colour (#***D55738***).
- **Subtitles:** If used, place them below the title. They must be a smaller font size and use a grey colour (#***4A5565*** / ***text-gray-600 tw***).
- **Primary Buttons:** The background must be ***primary-light*** (#***D55738***) by default, changing to ***primary-dark*** (#***B14400***) on hover.

★ Header & Navigation

Header Behaviour:

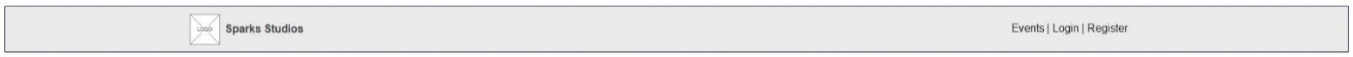
- The header must be displayed on all pages.
- It must remain fixed ("sticky") to the top of the screen when the user scrolls.

Navigation Links:

The links displayed in the header should change based on the user's login status.

For Unauthenticated Users (Guests):

- Events
- Login
- Register



For Authenticated Users (Logged-In):

- Events
- Profile
- Basket
- Logout

(**Note**: The 'Login' and 'Register' links should be hidden for users who are already logged in.)

★ Footer

Positioning:

- The footer must be present on all pages.
- It must always remain at the bottom of the viewport, even if the page content does not fill the screen's height (i.e., a "sticky footer").

Styling:

- Background: The background colour must be the same as the header (dark-1).
- Design: The overall layout and appearance must match the provided design sample.

Content: The footer must contain the following elements, in order:

- The text "Find your Spark!"
- A horizontal separator.
- A copyright line, formatted as © 2025 Spark Studios, which must include:
  - The copyright symbol (©).
  - The current year, which must be generated dynamically (not hardcoded).
  - The text "Spark Studios".

★ Landing page

Page Content:

- Main Heading: "***Public Events***"
- Subtitle: "***Browse from our publicly available events!***"

Core Functionality:

- Event Display: The page must fetch and display a list of all publicly available events.
- Filtering: The interface must include controls that allow users to filter the displayed events.

API Integration:

- Data Source: Use the ***GET /events*** endpoint to retrieve the list of public events.
- **NOTE**: By default, the ***/events*** endpoint does not return image data. To include images in the response, you must append the pictures query parameter to the API request (e.g., ***GET /events?pictures=true***).

★ Event Display & Interaction

Event Filtering

- Filter Controls: The following filter options must be available to the user:
  - Search: A single text input for searching by both event name and city.
  - Date: A date range selector (from/to).
  - Category: A dropdown menu populated with data from the ***GET /categories*** API endpoint.
  - Sort: A dropdown to sort events by date.
  - Clear All: A button that resets all filters and shows the complete list of events.
- **Behaviour and UX:**
  - Live Filtering: Filters must apply instantly as the user interacts with them, updating the event list without a page reload or a separate "submit" button.
  - Loading State: A visual loading indicator must be displayed while new event data is being fetched after a filter is changed.
  - No Results: If a filter combination returns no events, a clear message (e.g., "No events found matching your criteria") must be displayed.
- **Styling:**
  - The container for the filter controls should have a subtle grey background (#***F9FAFB*** / Tailwind: ***bg-gray-50***).

Event Grid & Pagination

- Layout: Events must be displayed in a 3x2 grid, with a maximum of 6 events per page.
- Pagination Logic: The application must handle the paginated data returned by the API.
- Pagination Controls:
  - Provide navigation controls (e.g., ***`Next`***, ***`Prev`***, page number buttons) to move between pages of results. When there are no further pages, the next/prev buttons are disabled.
  - Display a text indicator for the number of results currently visible (e.g., "*Showing 1-6 of 42 events*").

Individual Event Card Design

- Each event in the grid should be presented as a card with the following elements:
  - Hero Image: The first available picture of the event.
  - Category Badge: The event's category name, styled as a badge and overlaid on the hero image.
  - Event Name: The full name of the event.
  - Location: Preceded by a map pin icon (e.g. ***fa-map-pin***).
  - Date: Preceded by a calendar icon (e.g. ***fa-calendar***).
  - Time: Displayed in HH:MM format, preceded by a clock icon (e.g. ***fa-clock***).
  - Picture Count: The total number of pictures, preceded by a camera icon (e.g., ***fa-camera***).
  - Call to Action: A button with the text "***View Pictures***" that links to the event's detail page.

★ Event Detail Page

Page Header & Content

- Main Title: The full name of the event.
- Subtitle: The event's location and date, formatted as: "*In {city}, on {date}*".
- Picture Gallery: All pictures associated with the event must be displayed in a responsive grid, similar in style to the main events page.

Individual Picture Card Design

Each picture in the gallery grid must be presented as a card containing the following elements:

- Image: The picture itself.
- NOTE: Display the picture's note text, preceded by a sticky note icon (***`fa-note-sticky`***). If a note is not available (null), display a hyphen (-) instead.
- Picture ID: The unique pictureLocator ID, preceded by a file signature icon (***`fa-file-signature`***).
- Upload Date: The date the picture was uploaded, formatted as per the design sample and preceded by an upload icon (***`fa-upload`***).
- Add to Basket: An "Add to Basket" button, styled as per the design sample and including a shopping basket icon (***`fa-shopping-basket`***).

Picture Viewer (Lightbox) Functionality

- **Trigger:**
  - When a user hovers over any picture card, an expand icon (fa-expand) must appear in its top-right corner.
  - Clicking this icon must launch the picture viewer.
- **Features & Design:**
  - The viewer must display a large version of the selected picture.
  - It should appear over a semi-transparent black backdrop that covers the page content.
  - A close button must be present in the top-right corner to dismiss the viewer.
  - The overall design of the viewer must match the provided sample.

★ Login page

This page allows users to log in with an email and password to purchase images and browse their private content. See ***dist/assets/mock-ups/login.jpg***

Content & Typography:

- Main Title: "***Login***"
- Subtitle: "***To order from our wide variety of images!***"

Styling:

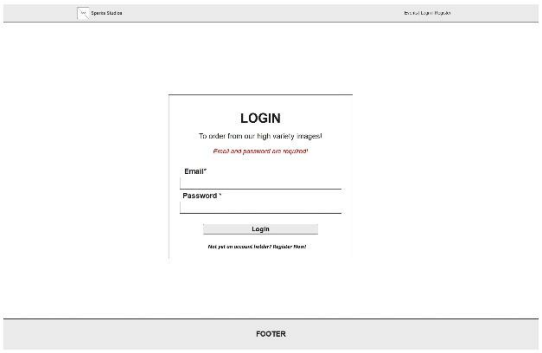
- Page Background: Grey (#***F9FAFB*** or Tailwind ***bg-gray-50***).
- Login Form Container: White background (#***FFF***).
- Registration Link: The link text ("***Register now!***") should use the primary brand colour.

Bottom Call to Action:

- A centred message must be displayed at the bottom of the page.
- Text: "***Not yet an account holder? Register now***!"
- Link: The "***Register now!***" portion must be a link that navigates to the registration page.

Access Logic:

- If a user who is already logged in attempts to visit the login page, they must be automatically redirected to the homepage (/).



★ Registration page

General Design & Layout:

- Consistency: The overall design should be consistent with the login page.
- Responsive Form: The layout of the input fields must be responsive:
  - Large Screens: Arrange fields in a two-column grid (2x4).
  - Mobile Screens: Stack fields vertically in a single column.

Form Validation & Error Handling:

- Required Fields: All mandatory fields must be clearly marked with a visual indicator (e.g., a red asterisk *).
- Client-Side Validation: Implement checks to prevent form submission if any required fields are left empty.
- Server-Side Validation: If the server returns a 422-validation error, you must display the specific error messages to the user in a clear and helpful way.

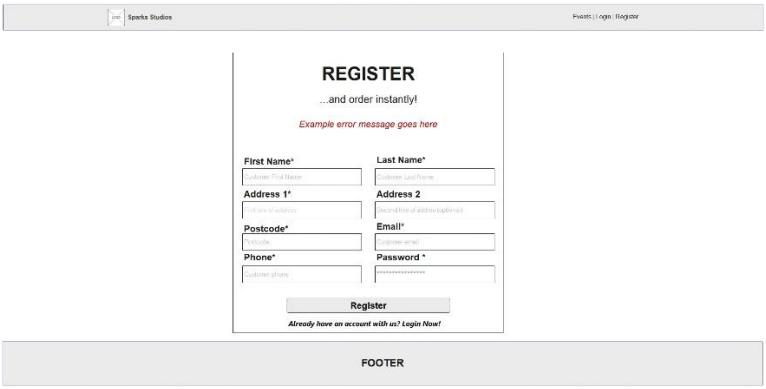
Call to Action (Link to Login):

- At the bottom of the page, include a centred text and link that directs users to the login page. The exact text and style should match the provided design sample.

Post-Registration Flow:

- Upon successful registration, the user must be automatically logged in.
- After being logged in, they must be redirected to the landing page (/).

Note: A list of available customers and their event access details can be found at the end of this documentation for reference.



★ User Access Control Rules

Unauthenticated Users (Guests):

- Allowed: May freely browse all public event pages.
- Restricted: Cannot access any protected pages or features.
- Action: Any attempt to access a protected area must redirect the user to the login page.

Authenticated Users (Logged-In):

- Have full access to both public and protected pages and functionality.

Protected Pages & Functionalities: The following features require a user to be logged in:

- Shopping Basket (including adding items)
- Accessing Private Events
- User Profile (including details, personal events, and order history)
- The checkout/ordering process

★ Authentication Flow & API Integration

User Login Process:

- Endpoint: ***POST /login***
- Request Body: The request payload must be a JSON object containing the user's email and password.

Handling Successful Logins:

- Upon a successful response, the API will provide an authentication token.
- Token Storage: This token must be stored securely on the client-side to manage the user's session.
- Redirection: After storing the token, redirect the user to the main landing page (***/***).

Authenticated API Requests:

- For all API calls that require authentication, the stored token must be included in the request headers.
- Header Format: Use the Authorization header with the Bearer scheme.
- Example: Authorization: Bearer <***your_api_token***>

Session Persistence:

- The user's session must persist across page refreshes. They should remain logged in until they explicitly log out.

Error Handling:

- If the /login endpoint returns a ***401 Unauthorized*** status (due to invalid credentials), you must handle this error gracefully.
- Provide clear, user-friendly feedback, such as displaying an "*Invalid email or password*" message.

★ Shopping Basket & Checkout Flow

"Add to Basket" Modal:

- Requires size and quantity selection (quantity >= 1).
- Must have live price calculation and input validation with error messages.
- On success, adds item to local basket, shows a success message, and updates the header counter.

Basket Page (Empty):

- Shows a specific view with a "Start shopping now" link to the homepage.

Basket Page (With Items):

- Title includes an item-count badge.
- Items are listed in a single, striped column.
- Each item row shows image, event name, ID, and **editable** size/quantity fields.
- The total costs of an order are calculated/updated immediately.
- Each row has a delete button (orange trash icon).
- A "Finalise order" button (orange, fa-check-double) is at the bottom.

"Finalise Order" Modal:

- Allows for an optional order note.
- A "Place order" button (fa-credit-card) submits the order to the server.
- On success, a "Thank You" message is displayed, and the basket is cleared.

UI & Display:

- Visibility: The "Basket" navigation link must only be visible when a user is logged in.
- Item Counter:
    - The link must display a badge that shows the current number of items in the basket.
    - This badge must have an orange background colour (#***E05600***).

Functionality & Data Management:

- Persistence: The basket's contents must be stored locally in the browser to ensure they are not lost on page refresh.
- On Logout: The basket must be completely cleared when the user logs out.

★ User Profile Section

Main Profile Page Layout

- Page Header:
    - Main Title: The current user's full name (e.g., "John Doe").
    - Subtitle: "Manage your profile".
- Tab Navigation:
    - Implement a tabbed menu with three sections: "My Details", "My Orders", and "My Events".
    - Styling: All tabs should have a border. The currently active tab must be visually distinct with a thicker border and orange text.

"My Details" Tab

- Functionality: Provide a form for the user to update their personal information.
- Data: The form must be pre-filled with the user's current details.
- Fields:
    - All personal details are editable, except for the email and password.
    - The user's email address should be displayed in a read-only (disabled) field.

"My Orders" Tab

- Functionality: Display a history of all orders for the logged-in user.
- Layout: Arrange orders in a responsive grid with 3 columns.
- Order Card Content: Each order in the grid must be a card containing:
    - Card Header:
        - Status Badge: A coloured badge indicating the order status (e.g., Green for Paid, Orange for Confirmed, Red for Cancelled).
        - Order Number: Formatted as "Order #...".
        - Total Price (£): If the order is Cancelled, the price must be greyed out with a strikethrough.
    - Card Body:
        - Item Count: Preceded by a hashtag icon (fa-hashtag).
        - Date Placed: Preceded by a calendar icon (fa-calendar).
        - Note: Display the order note, preceded by a sticky note icon (fa-note-sticky). If no note exists, show -.
    - Card Action:
        - A red "Cancel Order" button must be visible only if the order status is Confirmed.
- Interaction: Clicking an order card must navigate the user to that specific Order Detail Page.
- Developer Note: The test user john.doe@mail.com has 9 pre-existing orders for development purposes.

Order Detail Page (Navigated to from "My Orders")

- Page Title: The order number (e.g., "Order #..."), matching the selected order.
- Content: List every picture item included in the order. For each item, display:
    - Size: The selected size and its price, with an info icon (fa-circle-info).
    - Quantity: The number ordered, with a ribbon icon (fa-ribbon).
    - Line Total: The total price for that item, with a sterling icon (fa-sterling-sign).
- Styling: If the order is Cancelled, all prices on this page must be greyed out with a strikethrough.

"My Events" Tab

- Functionality: Display only the private events that the current user has been granted access to.
- Layout: The layout of the event list should be identical to the public events page.

- Constraint: This page must not include any filtering or sorting controls.

Data

The API gives you access to some pre-seeded data, which makes testing easier for you and for us.\

Customers

Each customer has the password 'password123'.
The available email addresses are:

- john.doe@mail.com, id: 1
- alice.smith@mail.com, id: 2
- michael.brown@mail.com, id: 3
- emily.davis@mail.com, id: 4
- olivia.wilson@mail.com, id: 5
- sophie.taylor@mail.com, id: 6
- david.smith@mail.com, id: 7
- marcus.wilson@mail.com, id: 8

Customer private event access

Each customer has access to different private events, and some customers have access to multiple events. These are fix accesses, and remain the same, even if you reseed the database.

Media Files

All media files are provided in ***dist*** folder

- api
  - the API
  - README.md (documentation)
  - api.yaml (documentation)
- assets
  - css
  - images
  - mock-ups
- database
  - db.sql
  - db_schema

General Guidance

Implement as many elements as you can within the allocated timeframe but aim to implement from each requirement.

Task Submission

Ensure that the module is accessible through URL ***ws{XX}.worldskills.uk/module-c*** where XX is you station number.

Marking Scheme Summary

SECTION	CRITERIA	TOTAL
C1	Design & Responsiveness	3
C2	Header & Footer	2
C3	Public Events Page	3
C4	Event filtering & Pagination	3
C5	Event Detail Page	3
C6	Basket & Checkout Flow	3
C7	Login & Registration	3
C8	Profile Page	3
C9	Order Detail Page	1
C10	Maintainability	1
	TOTAL ALLOCATED MARKS	**25**