Course:
JavaScript Programming Language and jQuery

Your examination is to complete one of two tasks given below.

Task 1

Create a single-page weather forecast website.
To obtain a forecast, use https://openweathermap.org/. Do not forget
to register and get the key.

The page should have two tabs:

- **Today** is the weather forecast for today;
- **5-day forecast** is the weather forecast for the next five days.

The **Today** tab is displayed when the page loads. The current city
is determined by the user's coordinates; if the browser does not support
geolocation, your city is displayed. To select another city, the user can
enter its name in the search box.

The Today tab displays three blocks.

1. Current weather summary:

- date;
- icon;
- text description;
- temperature;
- real feel;
- sunrise;
- sunset;
- day length (duration).

2. Hourly forecast for the rest of the day:

- time;
- icon;
- text description;
- temperature;
- real feel;
- wind speed and direction.

3. **Nearby cities and forecast:**

- name;
- icon;
- temperature.

The search box must always display the name of the city, for which the forecast is displayed, even if the geolocation was determined.

If the user enters a non-existent city or if API cannot return information on the entered city, inform the user about it using this page:
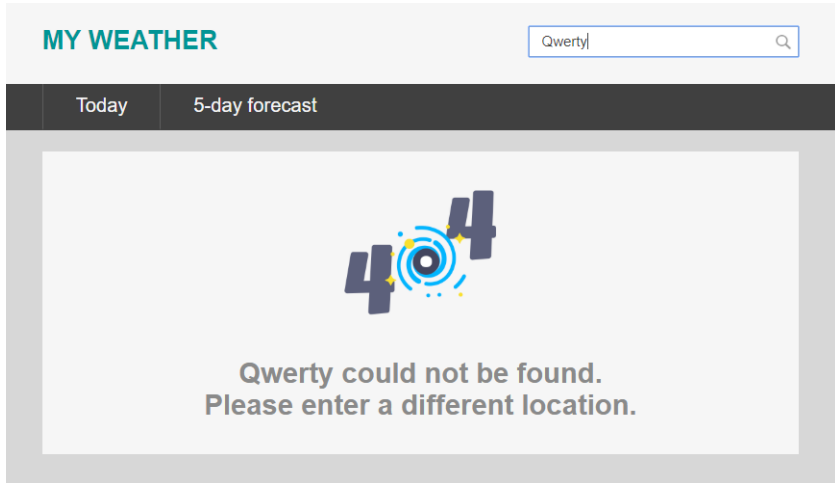
The **5-day forecast** displays 2 blocks.

1. Short forecast for each of the five days:
- day of the week;
- date;
- icon;
- temperature;
- text description.

2. Hourly forecast for the selected day:
- time;
- icon;
- text description;
- temperature;
- real feel;
- wind speed and direction.

When the short forecast of one of five days is clicked, it should be visually highlighted and have an hourly forecast below.

When this tab opens, today must be selected by default.
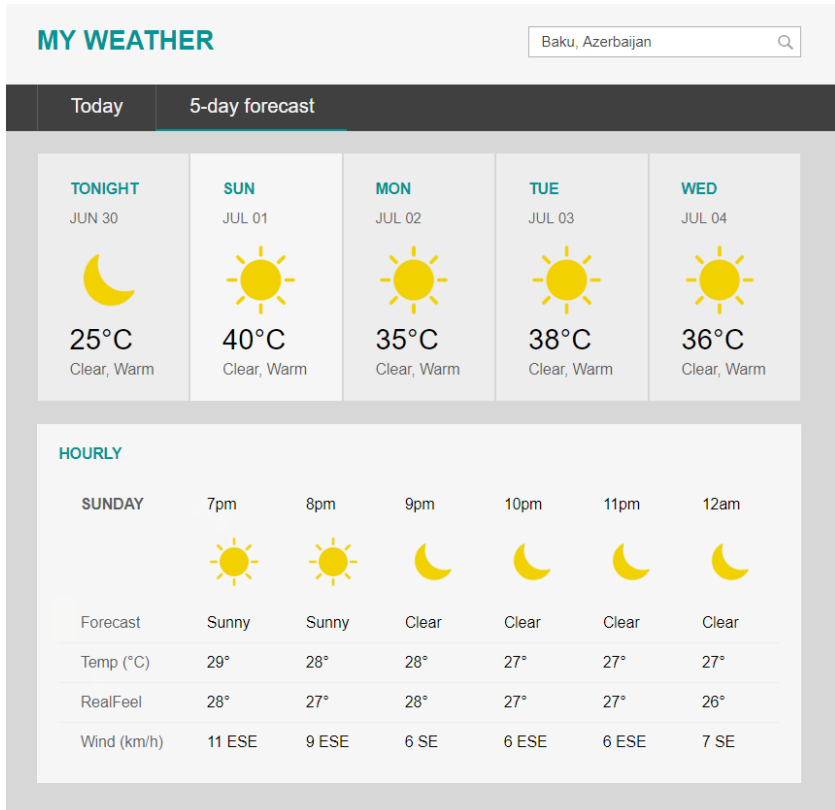
Remember that this is a single-page site. This means that clicks—be it searching for another city or navigating through tabs—must NOT (!) refresh the page in the browser. Change the page structure with JavaScript.

Task 2

Create a website for a librarian to keep track of books.

The site should consist of four sections (menu items).

1. Books.

On this page you can:

- view a list of all books;
- add a new book;
- edit the existing book;
- delete the existing book.

When viewing a list of all books, you can sort and search. You can sort by name of the book, author's name, or number of copies in the library. The user begins entering the word in the search box, and the program checks it in the name of a book or author or publisher.

When adding or editing a book, validate the following data: all fields are required, numeric fields cannot contain negative values.

2. Visitors.

On this page you can:

- view a list of all visitors;
- add a new visitor;
- edit the existing visitor.

When viewing the list of all visitors, you can sort and search. You can sort by ID or name. The user starts entering a word in the search box, and the program checks it in the visitor's name or phone number.

When adding or editing visitors, validate the following data: all fields are required, and the phone number must contain only numbers, space, and a dash.

3. Cards.

A **card** is one book lent by one visitor with the borrow and return dates. The return date is set only after the book is returned.

On this page you can:
- lend out a book (create a new card);
- take a book (set the return date).

When a book is borrowed, the number of its copies in the library decreases by 1. When returned, it increases by 1.

When creating a card, the drop-down list must display only the books currently available in the library (the number of copies must be greater than 0).

4. Statistics.

On this page you can:
- view the list of five most popular books;
- view the list of five most active visitors.

The main entities that you will need when developing the site (there may be more, this is the minimum list).

Book:
- ID;
- name;
- author's name;
- year of publishing;
- publisher name;
- number of pages;
- number of copies in the library.

Visitor:
- ID;

- full name;
- phone number.

Cards:

- ID;
- visitor's ID;
- book's ID;
- borrow date;
- return date.

Data must be stored in `localStorage` or `sessionStorage`.

The below Figures show a variant of the interface for **Visitors** and **Cards**.

List of visitors, sort and search

LIBRARY

| Books | Visitors | Cards | Statistics |

ALL VISITORS: New visitor

Sort by: [ID ▼] [Sort]   Search: [   ] [Search]

ID	Name	Phone	Edit
1	Colette Kelley	012 435 45 67	✏
2	Ruby-Rose Lennon	012 647 34 24	✏
3	Leanne Gibbons	012 879 78 45	✏
4	Rumaisa Peel	012 456 64 67	✏
5	Gene Medrano	012 245 47 89	✏

Editing a visitor
(the same form is used for creating a user)

List of cards, sort and search

The **Return Date** column displays either the return date or a button (an arrow). Click on the button to return a book, and the current date appears in the **Return Date** column.

LIBRARY

Books	Visitors	Cards	Statistics

ALL CARDS: New card

Sort by: [Return date ▼] [Sort]   Search: [   ] [Search]

ID	Visitor	Book	Borrow Date	Return Date
1	Colette Kelley	JS Part 1	12.01.2019	12.03.2019
2	Ruby-Rose Lennon	JS Part 2	12.01.2019	↩
3	Leanne Gibbons	HTML+CSS book	12.01.2019	↩
4	Rumaisa Peel	Bootstrap	12.01.2019	12.03.2019
5	Gene Medrano	SQL Part 1	12.01.2019	12.03.2019
6	Sheridan Tucker	SQL Part 2	12.01.2019	12.03.2019
7	Christina Mack	SQL Part 3	12.01.2019	12.03.2019
8	Everly Moses	PHP for beginner	12.01.2019	↩
9	Kara Feeney	C# for beginner	12.01.2019	↩
10	Cameron Rennie	C++ for beginner	12.01.2019	↩

Creating a new card

To create a new card, select a **Visitor** and a book from the drop-down lists, and the current date appears in the **Borrow date** column.