

obilet.com Sr. Full Stack Developer Programming Assignment

Create an ASP.NET MVC or ASP.NET Core MVC application that allows end users to select their desired origin & destination & departure date and lists available journeys for the specified query.

Wireframe & Design

[Click to access wireframe and design specifications](#)

The application consists of two pages

- Index (first wireframe)
- Journey Index (second wireframe)

Functional Requirements

Application-wide requirements

- All requests made to obilet.com business API should be coded in the MVC application backend (no client side requests should be made directly to obilet.com business api, any client side requests implemented should be made to the application backend.)
- A session should be created and maintained for each different end user visiting the application using the **GetSession** method of obilet.com business API. (see the API documentation) Each user should use his/her own session information in the subsequent API requests made by the application on behalf of that user.

Index

- All possible bus locations available should be fetched from the obilet.com business API **GetBusLocations** method (see the API documentation) and be listed as available origins and destinations.
 - Default values for origin and destination fields should be set according to the default sorting provided by **GetBusLocations** method response.
 - Default value for the Departure Date field should be tomorrow.
 - Users should be able to perform text-based search on origin and destination fields. The search keyword user enters should be used in order to fetch related bus locations from the obilet.com business API **GetBusLocations** method (see the API documentation).
 - Users should be able to swap selected origin and destination locations using the swap button shown in the design.
 - Quick selection buttons for setting the date to “**Today**” and “**Tomorrow**” should setting the value of the departure date field properly.
 - Following validations & limitations should be implemented with respective error messages.
 - Users can not select same location as both origin and destination.
 - Minimum valid date for departure date is Today.
 - Search button should redirect user to the journey index page with the specified **origin**, **destination** and **departure date** information.
 - Last queried **origin**, **destination** and **departure date** values should be stored on client side. Whenever a user revisits the application, existing origin, destination and departure date values should be set as default values, if available.
-

Journey Index

- Back button should redirect user to the Index page of the application.
- The origin, destination and departure date provided by the user should be passed to the obilet.com business API **GetJourneys** method. (see the API documentation)
- Journeys returned by the API response should be **sorted by their departure times** and displayed to the user.

Design Requirements

- Try to place your html elements to have a similar look and feel to the provided wireframes.
- **DO NOT try to implement the exact design specifications available on the wireframes.** (having a similar look and feel to the proposed design is enough)

Restrictions:

- You have one week to complete this task.
- Estimated task time is ~ 8h.
- Code should be clean, modular, scalable and reusable.

Bonus points for:

- Use best practices and design patterns
- Code should be readable & commented properly.
- Anything else you think worth adding is welcome.

Attachments

- **ApiClientToken:** JEcYcEMyantZV095WVc3G2JtVjNZbWx1
 - Token is required in the authentication header for making requests to obilet.com API (see the API documentation)
 - **obilet.com Business API Documentation**
 - [Click to view obilet.com API Documentation](#)
 - **Wireframe & Design Specifications for the application**
 - [Click to access wireframe and design specifications](#)
 - **Postman collection containing sample obilet.com api requests for the scope of the assignment.**
 - [Click to download sample postman collection](#)
-