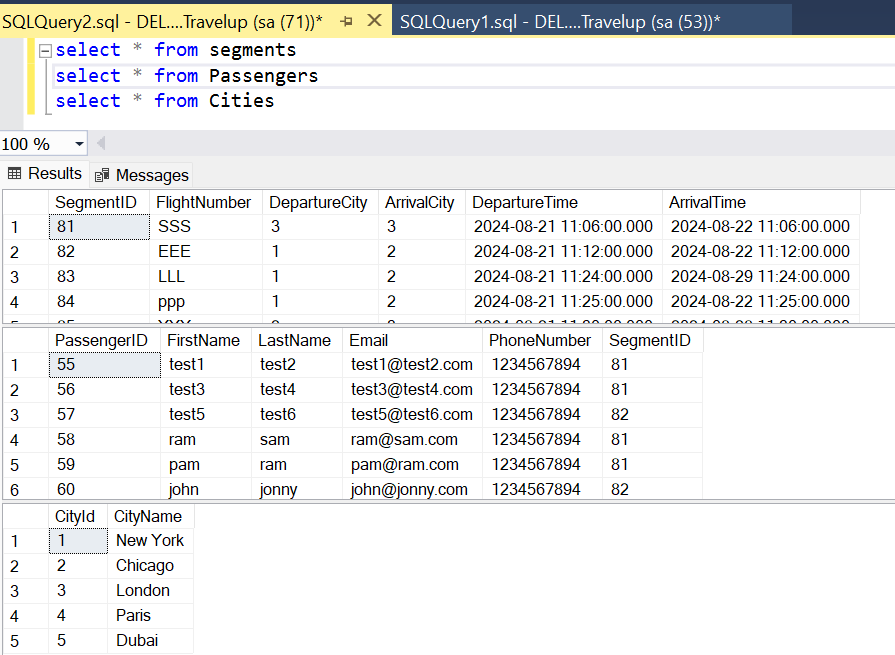
Database Structure:

* Travelup database contains 3 tables – Segmants, passangers and cities
* **DepartureCity** and **ArrivalCity** contains FK reference to **CityId** in City Table
* **SegmentId** in passangers table FK reference to **segmentID** in segments table

**ER Diagram would have been better to understand table structure unfortunately SSMS Database diagrams not working in my system.**



Application Structure:

1. Application contains 2 projects **WebAPI** and **TravelUp (**refer **fig1)**
2. Created restful API services using entity framework core, which can be consumed by any application (MVC/Angular)
3. TravelUp MVC project is used to handle incoming requests hit the required Api to get the response and deserialize the json response and display in Razor view
4. Implemented CORS in web api so that MVC application can access the api
5. Added facebook and linkedin link to footer layout which will redirect to respective pages(refer **fig2**)
6. Added error handling to redirect to error page (refer **fig3**)
7. Added success messages to show the success on web page(refer **fig4**)
8. Implemented Model validations on create and edit functionalities (refer **fig5**)
9. Added confirmation popup on the delete segment to delete the segment only if user confirms (refer **fig6**)

Few screen shots of application:

Fig1: Project structure

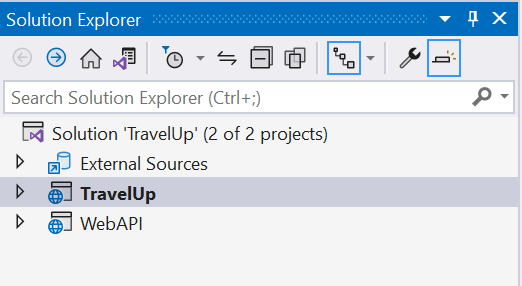


Fig2: FB and linkedin images in footer which redirects to respective pages



Fig 3: Error view when there is any error

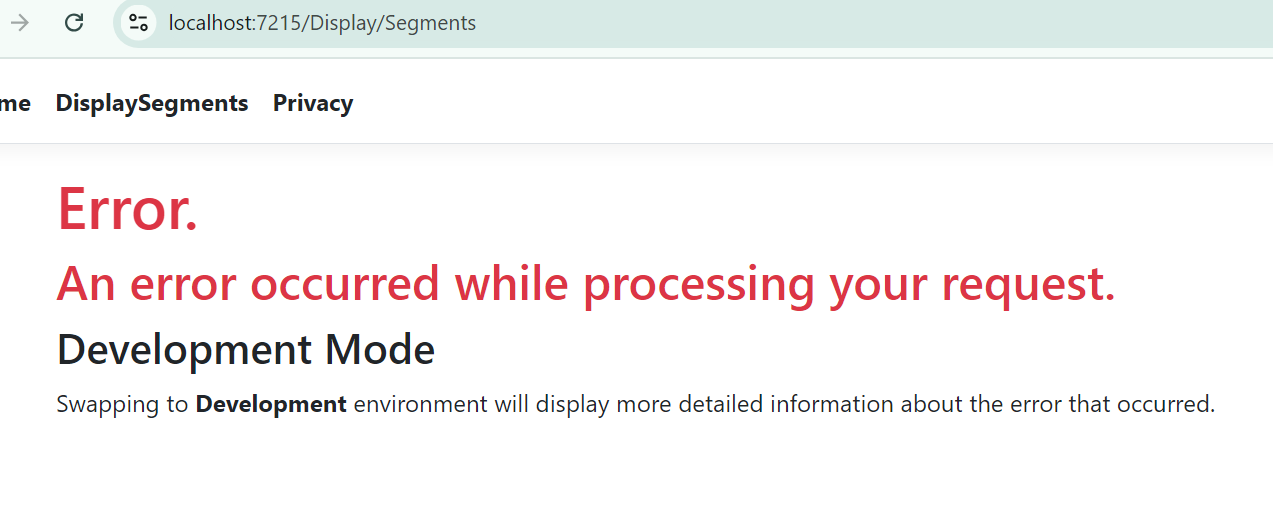


Fig 4 : success message:

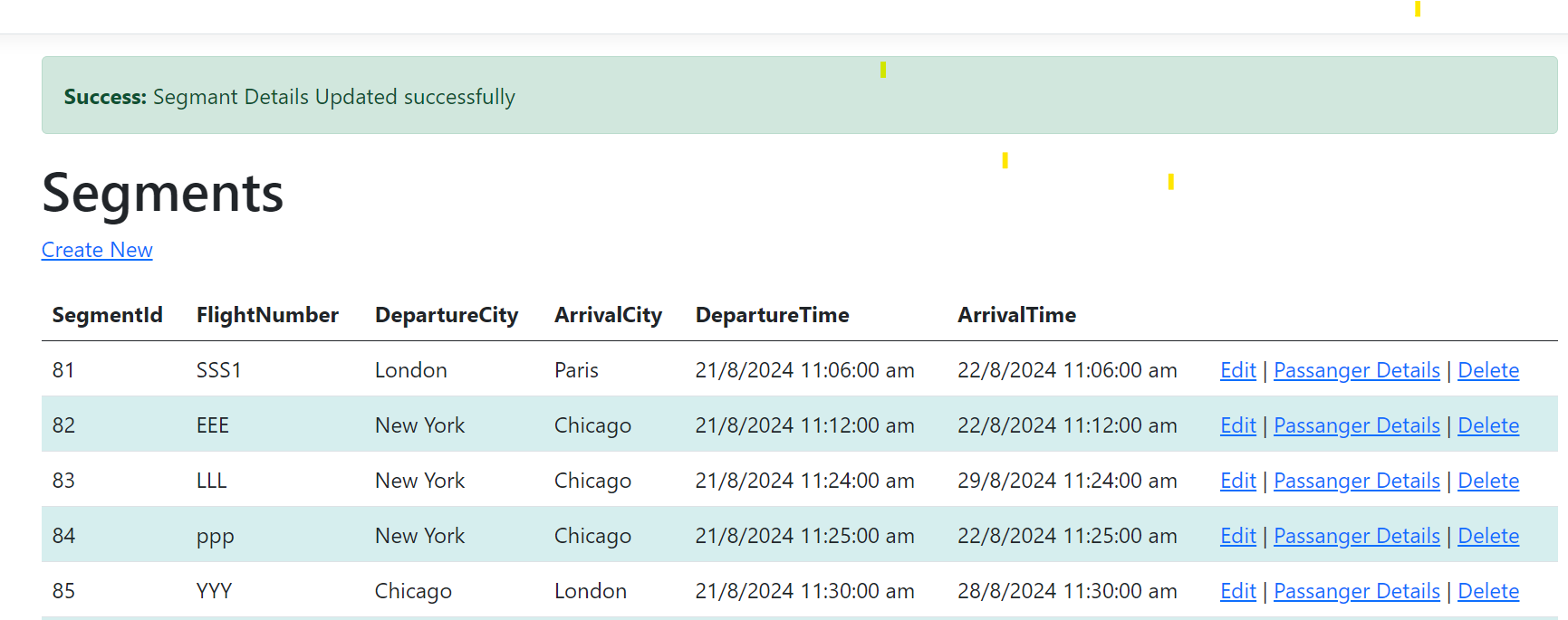


Fig5 : Required and Model validation when arrival and departure city selected is same

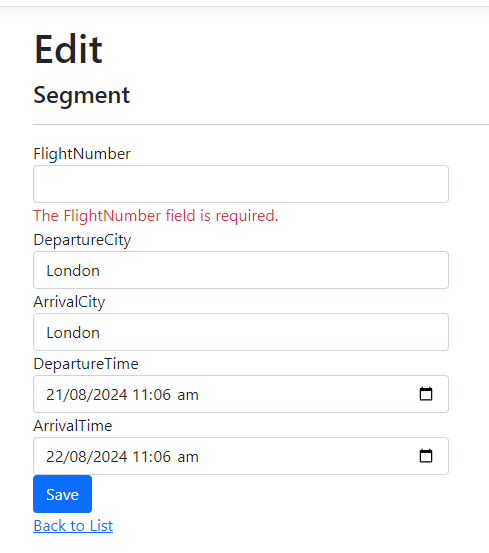
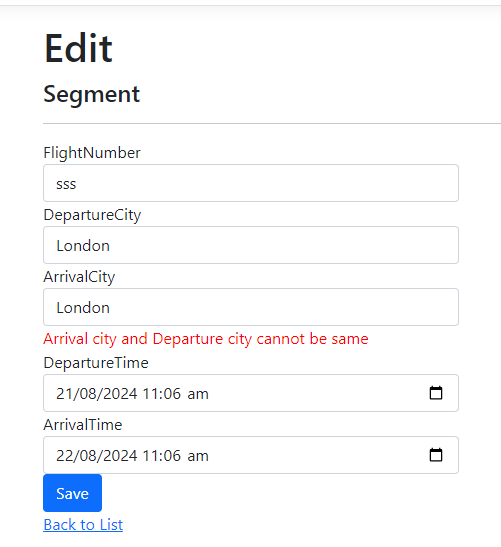
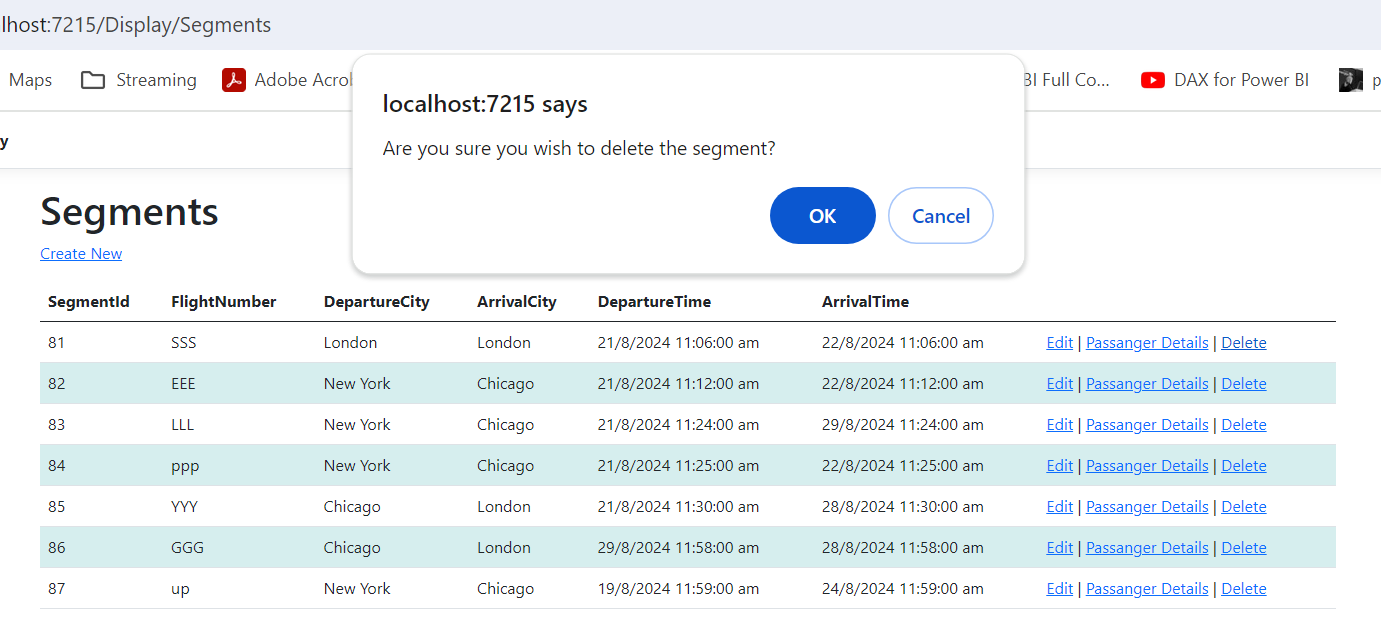


Fig6: Confirmation message on Delete



Future Scope:

* Need to implement more validation messages to check if arrival time is less than departure time
* More webapi validations can be added to prevent server side invalid data
* Login functionality can be added
* Testing framework implementation on web API to give more code coverage