

Assignment 1	Project Summary
Course	Angular 2 and TypeScript Web Application Development

Project author			
No	First name, last name	E-mail	Face-to-face/ online
1	Stilyan Mladenov	stilyan.mladenov@gmail.com	face-to-face
2	Daniela Postolova	daniella.postolova@gmail.com	face-to-face

Project name	Airbnb stats
--------------	--------------

1. Short project description (Business needs and system features)
<p>The project is a client-server app for statistics based on Airbnb dataset for the city of Amsterdam 2018.</p> <p>The main page of the app visualizes an interactive map of Amsterdam and allows the user to select a neighborhood. When a neighborhood is selected, different diagrams for the neighborhood are shown.</p> <p>The app requires a registration for the other functionalities. The system offers different ways for registration like email-password and different external providers (Facebook, Google). This functionality uses Auth0 identity server.</p> <p>Once the user is registered and logged in, he has access to the other main feature – a table with all listings. The table is flexible and offers a lot of different filtering, sorting and paging options. In this way, the user can see more details about particular data he is interested in.</p> <p>Filtering by:</p> <ul style="list-style-type: none"> - Neighborhood - Room type

- Property type
- Availability range
- Price range

Sorting by:

- Price
- Rating
- Accommodates

Paging options:

- 5 listings per page
- 10 listings per page
- 20 listings per page

Other system feature concerns reports based on the Airbnb dataset for the city of Amsterdam 2018. The reports scene gives visual and easy to understand statistic information by different criteria. The information is presented with variety of diagrams realized with Recharts library.

The API is realized with node.js, Express and Typescript. Other used technologies are:

- Auth0 - for authentication
- Sequelize with PostgreSQL - for data management and storing.

The web client is a React app realized in Typescript. It uses technologies like:

- Axios – for api calls
- Mapbox – a free map provider
- Deck-gl – adds layers for data visualization on the map
- Material-ui – for design
- Recharts – a composable charting library built on React components

2. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
2.1. Interactive map	<p>The <i>User</i> can select a neighborhood on the interactive map and see additional information about it:</p> <ul style="list-style-type: none"> - All listings for the selected neighborhood are visualized on the map; - Diagrams for the selected neighborhood are shown; 	<i>All users</i>
2.2. Register	An <i>Anonymous User</i> can register in the system using email-password authentication or an external provider like Facebook and Google.	<i>Anonymous User</i>
2.3. Listings	A <i>Registered User</i> has access to the table containing all listings and brief information about each one.	<i>Registered User</i>
2.4. Listings filter	<p>A <i>Registered User</i> can apply different filters to reduce the data. Filters by:</p> <ul style="list-style-type: none"> - Neighborhood; - Room type; - Property type; - Availability range; - Price range; 	<i>Registered User</i>
2.5. Sorting Listings	A <i>Registered User</i> can sort the data in the listings table by price, accommodates and rating.	<i>Registered User</i>
2.6. Paging options	<p>A <i>Registered User</i> can change the page size of the listings table. Paging options for:</p> <ul style="list-style-type: none"> - 5 listings per page; 	<i>Registered User</i>

	<ul style="list-style-type: none"> - 10 listings per page; - 20 listings per page; 	
2.7. Reports	A <i>Registered User</i> has access to the reports scene containing plenty of diagrams and statistics for all Amsterdam listings. The represented information is grouped by different criteria for easier navigating through the diagrams.	<i>Registered User</i>
2.8. About	A <i>Registered User</i> can see additional info about the project creators.	<i>Registered User</i>

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
3.1. Dashboard	<p>Presents the map of Amsterdam city. Offers abilities to select a neighborhood additional information about it:</p> <ul style="list-style-type: none"> - All listings for the selected neighborhood are visualized on the map - Diagrams for the selected neighborhood are shown 	/dashboard
3.2. Listings	Presents a table with all listings with short information for each one. Offers paging and flexible filtering and sorting by different criteria.	/listings?filters=...
3.3. Listing	Presents a detailed information for a listing.	/listings/:id
3.4. Reports	Presents plenty of diagrams grouped by different criteria. Shows statistic information based on Airbnb dataset for the city of Amsterdam 2018.	/reports

3.5. About	Presents information about the project and his owners.	<i>/about</i>
-------------------	--	---------------

4. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
4.1. Listings	GET listings by given parameters: <ul style="list-style-type: none"> - neights (for filtering by neighborhood); - prop_types (for filtering by property type); - room_types (for filtering by room type); - form_date (for filtering by availability range); - to_date (for filtering by availability range); - form_price (for filtering by price range); - to_price (for filtering by price range); - skip (for paging); - take (for paging); - order_by; The <i>React Client</i> uses it for retrieving the data for the listings table.	<i>/api/listings?skip=...\$take=...\$</i>
4.2. Listings	GET the data for the filters of the listings table.	<i>/api/listings/filters-data</i>
4.3. Listing	GET detailed data for a listing.	<i>/api/listings/:id</i>
4.4. Dashboard	GET the geolocations for each listing.	<i>/api/listings/locations</i>
4.5. Dashboard	GET all neighborhoods data concern the neighborhoods visualization on the map.	<i>/api/neighborhoods</i>
4.6. Dashboard	GET the reports data for the diagrams for all neighborhoods.	<i>/api/neighborhoods/reports</i>

4.7. Dashboard	GET neighborhood record by id.	<i>/api/neighborhoods/:id</i>
4.8. Dashboard	GET the reports data for the diagrams for a neighborhood with the given id.	<i>/api/neighborhoods/:id/reports</i>
4.9. Reports	GET the detailed reports data for a neighborhood with the given id.	<i>/api/neighborhoods/:id/detailed-reports</i>
4.10. Reviews	GET all reviews for listing with given id.	<i>/api/reviews/:id</i>
4.11. Hosts	GET all hosts.	<i>/api/hosts</i>
4.12. Host	GET data for host with given id.	<i>/api/hosts/:id</i>
4.13. Calendars	GET all calendars data for listing with given id.	<i>/api/calendars/:id</i>
4.14. Room Types	GET all room types.	<i>/api/room-types</i>
4.15. Property Types	GET all property types.	<i>/api/property-types</i>