**Code Challenge – NFL Crime App**

        Getting Started

o   Prerequisites (Environment used)

Angular: 5

        Technologies used:

o   Angular

        Get code for front end from

o   Run 'npm install' command at same location

o   After successfully having node modules in code, run 'ng serve' command, application is up and running on default port (4200)

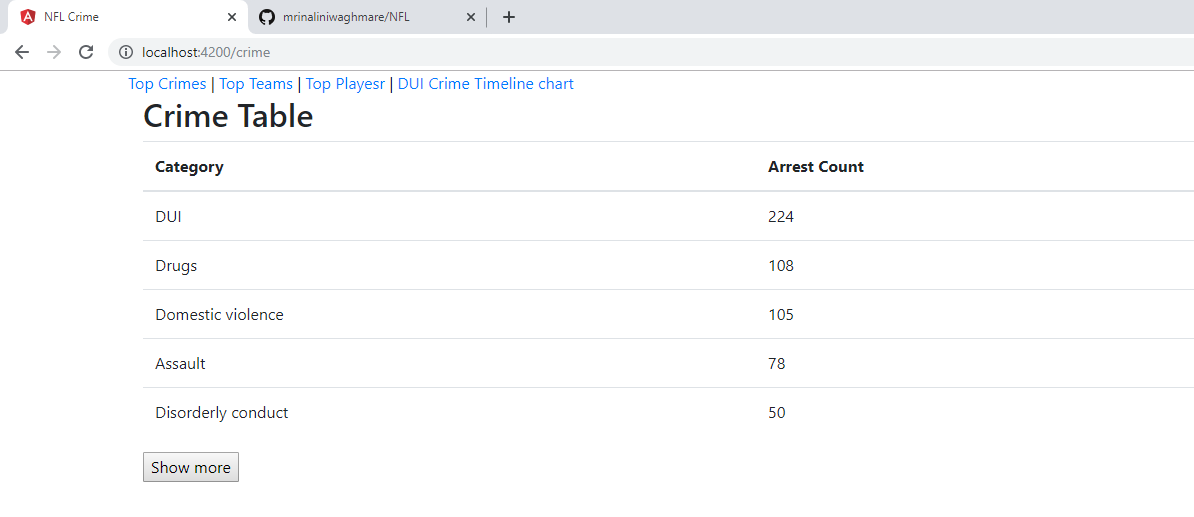
o   For production build – ‘ng build –prod’

        In browser window, hit following url

<http://localhost:4200/>

Public git repository - <https://github.com/mrinaliniwaghmare/NFL.git>

**Evidences –**

****

**Approach**

1. **Interpretation of problem statement**

* Framework used – Angular 5
* Git repository(public)
* Show data for crime, teams and player respectively using appropriate service and UI
* Filter functionality for every category of information fetched
* Show data in the visual(chart) format for better user understanding
* Show Datepicker component, in order to filter information for specific timespan

1. **Prioritizing tasks**

* Angular app up and running with public git repository as per the requirement
* Gather data for all categories
* show in appropriate (table) format applicable
* User friendly record rendering keeping ease in record accessibility
* Provide navigation to display information of selected category from available options (angular routes)
* Show data in chart format as per the requirement (assumption - DUI crime type for
* Crime timeline service)
* Add filter functionality
* Create Datepicker Component

1. **If I had more time to complete this task**
2. Using test driven development approach, unit test comes first
3. Division of folder structure in more sorted manner
4. Pagination for showing up data from all categories
5. User input asking for timeline details (chart, which crimeID or teamID user wants to see chart for)
6. In built angular pipes to filter data for all categories
7. Angular material library to show Datepicker component.
8. Filter of table data with date interval input from user
9. For styling would have used atomic css approach with either tailwind css or atomic css which will reduce css size and avoid redundancy.