* Use Case
  + user can select locations
    - user can type in address
      * autocompletion locations
  + user can pick isochrone time for driving or walking
    - 15, 30, 45, 60
  + user can change the map tile settings
    - street
    - satellite
    - dark
  + user can view recreational places and restaurants nearby
    - will show details about the place and google link
  + user can measure distance between two places
* Technologies
  + google api place search
    - details about the place
      * <https://developers.google.com/places/web-service/search>
    - address autocompletion
      * <https://developers.google.com/maps/documentation/javascript/examples/places-autocomplete?csw=1>
  + mapbox api
    - map tiles style
      * <https://docs.mapbox.com/api/maps/#styles>
    - isochrone data
      * <https://docs.mapbox.com/api/navigation/#isochrone>
  + MEAN stack
  + webpack
* Path analysis (click analysis)

1. user visits site
   1. default map tile is streets
   2. different location based on the day with interesting fact
      1. similar to google doodles
2. user selects location on map
   1. isochrone times appear
3. user can add up to 3 locations at a time
   1. typing address into search bar
      1. can choose the color of the marker
4. user can move marker
   1. by changing location in address bar

* Data Model
  + Restaurant and recreational locations
    - id
    - name
    - naics\_code
    - web\_address
    - address
    - location (geojson)
* Issues

1. Set up development environment
2. Set up the credentials for external services/APIs
   1. Google Places API
   2. Mapbox API
   3. Mapquest API
3. Get the locations of restaurants/recreational in Mississauga
4. Design the Wireframe
5. Set up on MongoDB Atlas
6. Implement the backend routes
7. Test the backend routes
8. Implement the backend DB
9. Add location data to DB
10. Test the backend DB
11. Implement the frontend
12. Interface the frontend and backend

* API endpoints
  + /api/v1/locations – GET all locations
  + /api/v1/locations/{coordinates} – GET locations in an area by latitude and longitude