1. ER/UML Diagram

https://lucid.app/lucidchart/4093bfba-74a7-4c87-90be-ae6641a4627b/edit?viewport_loc=590%2C-3442%2C3446%2C3708%2C0_0&invitationId=inv_41b2f101-5cce-4952-8c24-4124dd3da6e3

2. Explain your assumptions for each entity and relationship in your model. Discuss why you've modeled something as an entity rather than an attribute of another entity. Describe the cardinality of relationships, like why a student is linked to only one advisor. These assumptions might come from customer requirements or application constraints. Please clarify them.
   a.

      i. User (Entity): Each person who uses our app must create an account to access personalized features, such as preferences. Users will sign in using an email account and password.

      ii. Search Preferences (Entity): Users can have multiple preferences based on various factors, including cost, GDP, crime rate, and population change. (1:Many). It uses PreferenceID as a primary key to identify each set of preferences. After taking in user input, it puts a weight on each category and multiplies this weight with the proper data giving us a Quality of Life attribute.

      iii. Region: The ZIP Code serves as a primary key, as it uniquely identifies a specific area within a county, city, or state. We need this because ZIP Codes are used as foreign keys across multiple tables and must be associated with data. We also have county since some of our data sources cannot find data based on zipcode and will get its data based on the county instead.

      iv. Data: This module is used to represent our outside data sources where we get our information about crime rates, population growth, job opportunities etc. Each ZIP code has its own row of data. (1:1)

      v. SearchResult: A search preference provides a ranked list of ZIP codes, each accompanied by a computed Quality of Life (QOL) score. This is calculated using the Socioeconomic data given by its respective entity. Each preference can generate multiple search results (1:Many)

3. Normalize:

Our database is almost fully 3nf normalized to begin with as there is very little redundancy in our database between tables. Each table is functionally dependent on its primary key as a way to connect between tables and any shared information, such as city, state, or county, is moved to a separate table to remove repetition. Our database block which functions as an entity cluster in our ER diagram, shows 3nf normalization through our datasets as well. Each independent table has one primary key, either zip code or county, and has no redundant information between the tables. This allows us to access all relevant information needed to properly determine the best zip code for the user while minimizing the amount of interaction between different tables and redundancy.

4. Relational Schema:

   a. User(
   UserID: INT [PK] ,
   Username: VARCHAR(50),
   Email: VARCHAR(100),
   Password: VARCHAR(100)
)

   b. SearchPreferences(
   PreferenceID: INT [PK],
   UserID: INT [FK to User.UserID],
   HousingAffordibilityWeight: FLOAT,
   JobOpportunitiesWeight: FLOAT,
   PopulationTrendsWeight: FLOAT,
   GDPGrowthWeight: FLOAT
)

   c. Region(
   ZipCode: INT [PK],
   County: VARCHAR(100),
   City: VARCHAR(100),
   State: VARCHAR(2)
)

   d. Data(
   ZipCode: INT [FK to Region.ZipCode],
   County: VARCHAR(100)

HousingAffordability: FLOAT,
JobOpportunities: FLOAT,
PopulationGrowth: FLOAT,
GDPGrowth: FLOAT
)

e. SearchResult(
PreferenceID: INT [FK to SearchPreferences.PreferenceID],
ZipCode: INT [FK to Region.ZipCode],
QOL: FLOAT
)