Jim, Mark, Krishna – July 23rd

Data Visualization ?

1. **Mortgage Refinance Comparison Tool:**

The **Mortgage Refinance Comparison Tool** aims to assist homeowners in evaluating mortgage refinance options. Users can input their existing mortgage details and compare different refinance scenarios. The tool will provide visualizations and insights to guide their decision-making process.

* + **Project Overview:**
    - Create a web application that assists homeowners in evaluating mortgage refinance options.
    - Users input their existing mortgage details (loan amount, interest rate, remaining term) to compare different refinance scenarios.
  + **Key Features:**
    - **Input Form:** Design an HTML form for users to enter current mortgage information.
    - **API Integration:** Fetch real-time mortgage rates from external APIs (e.g., Freddie Mac).
    - **Visualizations:**
      * Bar chart comparing current interest rate with potential refinance rates.
      * Line chart showing the impact of different refinance terms on monthly payments.
    - **Database:**
      * Store historical mortgage rate data in a database (e.g., PostgreSQL).
    - **User Interaction:**
      * Allow users to explore scenarios (e.g., shorter term, lower rate) and visualize payment changes.
  + **Challenges:**
    - Ensure accurate and up-to-date rate data.
    - Design an intuitive interface for users.
    - Consider rate lock scenarios during the refinance process.

1. **Health Equity Tracker:**

The **Health Equity Tracker** aims to address health disparities among marginalized groups. Using COVID-19 and other social determinants of health data, the tool will visualize disparities and trends across demographics.

* + **Project Overview:**
    - Build a data visualization platform to explore health disparities among marginalized groups.
    - Use COVID-19 and other social determinants of health data.
  + **Key Features:**
    - **Data Source:**
      * [Utilize publicly available health equity data (e.g., from the Health Equity Tracker](https://healthequitytracker.org/)[1](https://healthequitytracker.org/)).
    - **Visualizations:**
      * Heatmaps showing disparities in health outcomes (e.g., infection rates, vaccination rates) across demographics.
      * Line charts tracking changes over time.
      * Geospatial maps highlighting regional disparities.
    - **User Interaction:**
      * Allow users to filter by demographics (race, income, education) and explore trends.
  + **Challenges:**
    - Ensure data accuracy and privacy.
    - Address potential biases in the data.
    - Create an accessible and informative user experience.

1. **Long-term Vacation Rental Property Finder**

**Project Overview: Long-term Vacation Rental Property Finder**

The Long-term Vacation rental property finder/comparison tool aims to assist users in finding and comparing rental properties for extended stays. Whether planning a long getaway or relocating, users can explore available properties, filter based on preferences, and visualize essential information.

**Key Components:**

1. **Data Collection and Storage:**
   * Gather data on vacation rental properties (e.g., location, price, amenities).
   * Store data in a database (e.g., PostgreSQL, MongoDB).
2. **Backend Development:**
   * Set up a Flask backend with interactive API routes.
   * Implement user-driven interactions (e.g., filtering by location, budget).
   * Serve data to frontend based on user queries.
3. **Visualization Libraries:**
   * Choose a visualization library (e.g., Matplotlib, Plotly).
   * Create visualizations (maps, price distributions, amenity comparisons).
4. **Frontend Development:**
   * Design a user-friendly interface.
   * Allow users to input preferences (location, budget).
   * Display dynamic visualizations based on user selections.
5. **Views (Minimum of Three):**
   * **Map View:** Show rental properties on a map (using Leaflet or similar).
   * **Price Distribution:** Display rental prices (histogram or box plot).
   * **Amenities Comparison:** Compare amenities across properties.
6. **Deployment:**
   * Deploy the application (frontend and backend) to a server or cloud platform.