CS122 Project Proposal – Chicago Wanderlust

Group Name: ChicagoExplorer
Team Members: Ran Bi, Weijie Xin, Tommy (Leping) Yu, Minjia Zhu

Part I. Objectives and Software Design

Chicago Wanderlust aims to score and recommend lodging locations customized for user's interests. It allows travelers to explore Chicago's diverse neighborhoods beyond downtown.

The user – travel planner – inputs intended accommodation address and his/her priority preference among four criteria: appetite, attractions, convenience and safety. Chicago Wanderlust will score the location, visualize its comparison with neighborhood/city average, and display detailed information with respect to the four criteria.

If user seeks a better location to satisfy his/her particular interests, Chicago Wanderlust will further generate a heatmap based on user's preference and recommend hotels for best travel experience.

Part II. Fantastic Data and Where to Find Them

1. **Hotels**
   - Source of data: Tripadvisor.com, Google Maps
   - Data of interest: hotel name, hotel address, hotel coordinates
   - Rationale: Generate a pre-saved list of hotel matched with Google map coordinates

2. **Appetite**
   - Source of data: yelp.com (API provided)
   - Data of interest: number of reviews, rating, type of cuisine, price
   - Rationale: Score quality and density of restaurants within certain radius of given location

3. **Convenience**
   - Source of data: Timeout Chicago, Google Maps
   - Data of interest: attraction type (art, park, museum, etc.), attraction name, trip duration
   - Rationale: Generate a list of attractions nearby and extract real time trip duration.

4. **Safety**
   - Source of data: City of Chicago Data Portal (dataset provided), geographical boundaries of community areas (Look for an API)
   - Data of interest: IUCR (crime type), Location: District + Ward + Community Area
   - Rationale: Given a pair of coordinates, determine the community area it belongs to, and provide the corresponding crime indicator

Part III. Tools and Useful Techniques

Stage	Tool

Data gathering	**Scrapy** (for closed-source website)
	API (for open-source website)
Data Cleaning	**Data Wrangler**, Pandas, Numpy
Data consolidation	SQL
Web Development Framework	**Django**
Front-End Framework	**Bootstrap**
Data visualization	Plotly/D3/Tableau/Bokeh (TBD)

Part IV. Tentative Timeline of Completion

1. **Week of 1/23: Data gathering**
    - *Proposal due 1/24*
    - Identify appropriate datasets. Gather data through web scraping and API.
2. **Week of 1/30: Data processing**
    - Data cleaning and wrangling. Consolidate relational database.
    - Consensus on "high level architecture": data structure, class, user interaction, etc.
    - Break out self-study sessions on new techniques and cross training.
3. **Week of 2/6: Site building**
    - *Checkpoint 1: Report preliminary results and issues (if any) from data processing.*
    - Frontend: Sketch user interface, web framework and pseudo-code.
    - Backend: Determine attributes calculation method. Explore scoring algorithm.
4. **Week of 2/13: Site building**
    - Frontend: Functional heat map & graph output on selected data sample.
    - Frontend: Continue working on web development.
    - Backend: Complete searching, scoring and recommendation algorithm.
5. **Week of 2/20: Site building**
    - *Check point 2: Report site building progress. Reflect on scope of the project and adjust if necessary.*
    - Frontend: Incorporate fully-functioning frontend with data visualization.
    - Backend: Link frontend, backend and database.
6. **Week of 2/27: Prototype**
    - Complete software prototype
    - Implement manual tests (QA)
7. **Week of 3/6: Refine prototype**
    - *Final presentation.*
    - Final software modification, troubleshooting, and documentation.
    - Reflect on future enhancement and work plan.
8. **3/14 Final project due.**