

# SI 507 Final Project Overview & Proposal Guidelines

SI 507 Teaching Team

## 1 Project Goal

The final project showcases what you've learned in SI 507 through a project you can discuss in interviews or internships. You will pick an interesting topic, find data related to that topic, use a **graph (network) data structure** to analyze that data, and build a program that allows users to interrogate that dataset and examine your analysis.

## 2 Core Requirements

### 2.1 1. Network/Graph Structure (Required)

- Your project **must** use a graph data structure as the primary analytical framework
- Clearly define what your **nodes** represent (e.g., players, artists, restaurants, books)
- Clearly define what your **edges** represent (e.g., teammate relationships, playlist co-occurrences, shared user reviews)

### 2.2 2. Real Data (Required)

- Work with **real-world data** from genuine sources (APIs, web scraping, CSV files, databases)
- **Synthetic data is allowed** only if clearly specified in your proposal and you provide justification for why real data isn't feasible
- Demonstrate appropriate complexity through either:
  - Volume of data processed, OR
  - Difficulty of data access (complex APIs, web scraping, data cleaning)

### 2.3 3. Multiple Datasets (Tiered Grading)

- Projects using **two or more related but distinct datasets** receive full points in this category
- Projects using **only one dataset** receive partial points
- Integrating multiple datasets allows you to reveal insights impossible with a single source
- Example: NBA player statistics + social media mentions, or Spotify artists + concert venue data

### 2.4 4. Four Modes of Interaction (Required)

Your program must provide **at least four different ways** for users to explore the data:

**Examples of interaction modes:**

- **Search/Query:** Find a specific node or type of node in your network
- **Node Information:** Get detailed data about a specific node (stats, attributes, metadata)

- **Relationship Analysis:** Find most closely related nodes or most similar nodes to a node of interest
- **Path Finding:** Find the shortest path between two nodes
- **Centrality:** Identify the most connected nodes in the network
- **External Links:** Provide links to more information about nodes
- **Filtering:** View subsets of the network based on criteria
- **Ranking:** Show top nodes by various metrics

## 2.5 5. Presentation (Not Required but Encouraged)

- Text-based command-line interaction is **perfectly acceptable**
- Graphical visualization (Flask, web interface, graph plotting) is **encouraged and impressive** but not required
- Focus on functionality and analytical depth over visual polish

## 3 Graph Analysis Methods

You can use any combination of these network analysis techniques:

- Identify most highly connected nodes (centrality measures)
- Find shortest paths between data points
- Identify clusters or communities of related nodes
- Calculate node importance or influence
- Detect patterns in network structure

You don't need to discover anything novel. Highlighting obvious or expected relationships is completely fine.

## 4 Example Projects

### 4.1 Example 1: NBA Teammate Network

Modeled after the Kevin Bacon project, this examined teammate relationships in the NBA over time.

- **Data:** NBA team rosters for every year from basketball-reference.com (CSV files)
- **Network:** Players as nodes, edges connect players who were teammates
- **Interactions:** Enter two players and find connecting chain, identify most connected players per season, find all teammates of a player, launch browser to view player page
- **Complexity:** Multiple layers of interactivity compensate for straightforward data access

### 4.2 Example 2: Spotify Artist Network

Built a network based on which artists appeared together in Spotify playlists.

- **Data:** Spotify playlist data via API
- **Network:** Artists as nodes, edges weighted by co-occurrence in playlists
- **Interactions:** Enter artist for recommendations, request playlists, view commonly paired artists, analyze popularity
- **Complexity:** API integration and sophisticated recommendation algorithm

## 5 PROJECT PROPOSAL GUIDELINES

Submit a **0.5 to 1 page proposal** for your final project. This proposal will receive iterative feedback until it is approved.

**Due:** [Date]

### 5.1 Required Elements in Your Proposal

#### 5.1.1 1. Area of Interest & Network Structure

Describe your topic and explain how a **graph (network)** is involved.

**Address:**

- What topic are you exploring?
- What will your **nodes** represent?
- What will your **edges** represent?
- Why is a network/graph a good structure for this data?

#### 5.1.2 2. Data Sources

Identify what data sources you think are **available and attainable**.

**Address:**

- Where will you get your data? (Be specific with URLs if possible)
- What format is the data in? (API, CSV, web scraping, database)
- How difficult will data access be?
- If using **two or more datasets**, how will they be integrated?
- If using **synthetic data**, clearly state this and explain why real data isn't feasible

#### 5.1.3 3. Planned Interactions

**In your proposal, describe the types of interactions you plan to support.** You don't need to specify all four in detail, but give a sense of the analytical capabilities your program will provide.

### 5.2 Proposal Format

Your proposal should be **0.5 to 1 page** and address all three elements above. Here's a suggested structure:

[Project Title]

**Topic & Network Structure:** [1-2 paragraphs describing your area of interest, what your nodes and edges represent, and why a network is appropriate]

**Data Sources:** [1-2 paragraphs describing where you'll get data, formats, access methods, and integration approach if using multiple datasets]

**Planned Interactions:** [1 paragraph describing the types of interactions you plan to support, such as searching, finding paths, analyzing centrality, viewing node details, etc.]

[Optional: 1 paragraph on any concerns or questions you have]

## 6 Milestones

### Total Project Points: 200

- Proposal: 40 points (completion grade, banked before final submission)
- Checkpoint: 20 points (completion grade, banked before final submission)
- Final Submission: 140 points (graded on rubric below)

By the time you submit your final project, you will have already earned 60 points from the proposal and checkpoint milestones, assuming you completed them satisfactorily.

### 6.1 Milestone 1: Project Proposal (0.5-1 page) - 40 points

**Due:** [Date]

Submit proposal as described above. You will receive feedback until approved.

**Grading:** Completion grade. You may resubmit based on feedback until you receive full points. All proposals must be approved before proceeding to the next milestone.

### 6.2 Milestone 2: Project Checkpoint (1 page PDF) - 20 points

**Due:** [Date]

This checkpoint ensures your project is viable and that you're making progress. Submit:

1. **Screenshot of your data in Python:** Prove you can access your data (not just that it exists in the world)
2. **Data checkpoint:** Demonstrate you're successfully collecting, caching, and storing all relevant data from your sources in JSON or HTML cache files
3. **Three questions for yourself:** Questions guiding your next steps or decisions you need to make
4. **Interactive presentation design:** Brief description (1 paragraph) of your plans for implementing interactive presentation capabilities, including user options supported and presentation types
5. **Questions for instructors OR confidence statement:** Either list questions for us, or state "I have no questions for the instructor and am confident in my current direction for this project"

**Grading:** Completion grade. Full points awarded if all components are present and demonstrate genuine engagement with your project. You will only receive detailed feedback if something is wrong.

### 6.3 Milestone 4: Final Submission

**Due:** [Date]

See detailed submission guidelines in separate document.

## 7 Final Submission Grading Rubric

The final project will be graded out of 140 points according to the following rubric:

**Note:** Projects with basic text-based interfaces and single datasets can still pass but will receive fewer points in those categories. To achieve the full 140 points, you need exceptional work including multiple integrated datasets and a high-quality GUI or visualization.

Component	Requirement	Points
<b>README - Project</b>		
	README is a .txt file	5
	README contains description of how user interacts with program	10
	Network structure clearly specified (nodes & edges)	5
<b>README - Data</b>		
	README is a .txt file	5
	README includes all data sources with URLs	10
	Data access techniques are clearly described	5
	Data summary provided with relevant fields described	10
<b>Data Structure Code</b>		
	Required .py and data files provided	5
	.py files are appropriately docstringed	5
<b>Project Code</b>		
	.py file(s) and data file(s) submitted that launch and run as shown in demo	15
<b>Demo Video</b>		
	Link to demo video or MP4 provided and works	5
	Application capabilities described and demonstrated clearly	5
	Four or more different user interactions demonstrated	25
	Data presentations are clear, readable, and make sense	15
<b>Data Integration</b>		
	Two or more distinct datasets integrated (full points)	10
	Single dataset only (partial points)	5
<b>Writing Quality</b>		
	Writing is clearly original work, not obviously AI-generated	10
<b>Presentation/GUI</b>		
	Exceptional graphical user interface or visualization	10
	Basic text-based interface (partial points)	5
<b>Total</b>		<b>140</b>

## 8 Getting Help

- Start early and come to office hours
- Post questions on the discussion board
- Iterate on your proposal based on feedback
- Remember: this is meant to be portfolio-worthy work you can discuss in interviews