UNIVERSITY OF LEICESTER

School of Computing and Mathematical Sciences

CO7201 Individual Project

Interim Report

Exploring HESA Data with Large Language Models for Dynamic Visualisation

Vladimirs Ribakovs

vr112@student.le.ac.uk
239062116

Project Supervisor: Heckel, Reiko (Prof.)
Principal Marker: Goes, Fabricio (Dr.)
Word Count: 1700
(excluding table contents, table labels, headings and references)

26/03/2025

DECLARATION
All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Vladimirs Ribakovs
Date: 26/02/2025

Contents

1. Overview

This project has been completed to approximately 44% as of week 7 of the project schedule. The project has changed it initial approach by shifting the LLM based processing further down to the Phase 3 of the project and instead for the data processing using more deterministic and specialized approach that including mathematics and regular expressions. Using mathematics and regular expressions is considered to be a safe approach in case something is wrong with using LLM data processing.

Over the course of 0 to 7 weeks has been achieved the following functionality:

- Data Processing Pipeline
  - Implemented advanced CSV processing with metadata extraction
  - Created sophisticated keyword-based search system with:
    - Density-based scoring algorithm
    - Synonym support with extensible dictionary
    - Case-insensitive substring matching
  - Established data cleaning and transformation workflows
  - Implemented JSON metadata structure for enhanced data organization
  - Added support for academic year format handling (YYYY/YY)

- Query Processing System
  - Developed parameter extraction from user queries
  - Implemented intelligent query tokenization and normalization
  - Created weighted scoring system for dataset matching
  - Added flexible institution name matching
  - Implemented mission group filtering capability

- Data Storage and Management
  - Implemented file-based storage system with JSON metadata
  - Created structured data organization system
  - Developed metadata extraction and management
  - Implemented basic file versioning

- User Interface and Interaction
  - Implemented basic dashboard interface
  - Added dynamic dataset preview functionality
  - Created configurable match limit system (default: 3)
  - Implemented basic data filtering controls
  - Added institution and academic year range selection

- Visualization Capabilities
  - Implemented basic data display functionality
  - Prepared foundation for advanced visualization features

- Project Structure & Management
  - Organized modular codebase with clear separation of concerns
  - Implemented basic error handling
  - Created foundational testing structure

- Project Submissions
    - Project description
    - Preliminary report

Current Focus:

- Re-implementing LLM integration using direct API approach
- Enhancing the visualization capabilities
- Implementing data export functionality
- Improving user interface and experience

2. Project Progress

The overall progress of the project goes almost as planned. It follows the original Gantt Chart time flow, but the Gantt Chart task have been extended and LLM implementation has be postponed from Phase 2 to Phase 3. Besides only separating the whole process into 4 phases, I also separated the project itself into 7 phases. This will provide mode detailed overview of the project and make it clearer to understand. Overview of major accomplishments:

DONE ☑

- Phase 1: Project Setup and Basic Infrastructure | 100%
    - Initialize Django project structure
    - Set up development environment (virtual environment, dependencies)
    - Configure version control (Git)
    - Implement basic project structure
    - Set up basic routing and views
    - Configure database settings
    - Implement base template with modern UI framework (Tailwind CSS)

- Phase 2: Data Processing and Visualization Foundation | 70 %
    - Implement CSV data processing pipeline with advanced features
        - Metadata extraction and management
        - Keyword-based search with synonym support
        - Density-based scoring system
        - Year-based filtering (YYYY/YY format)
        - Institution substring matching
        - Mission group filtering
    - Implement data storage system
    - Basic data preview functionality
    - Implement data filtering and search functionality
    - Implement basic dataset selection and preview system

IN PROGRESS ◍

- Phase 2: Data Processing and Visualization Foundation
    - Integrate API that would automate the testing process so it would primarily rely on LLM rather than on regular expressions and calculations

- Phase 3: LLM Integration and Advanced Features
    - Re-implementing LLM integration using direct API approach
    - Natural language query processing (partial - keyword extraction implemented)

TODO SOON

- Phase 3: LLM Integration and Advanced Features | 0 %
    - Complete LLM API integration
    - Implement visualization recommendation system

- Phase 4: Advanced Visualization and Analysis | 0 %
    - Implement advanced chart types
    - Add interactive features
    - Create dashboard templates
    - Implement data export functionality (CSV, PDF, Excel)

- Phase 5: User Experience and Documentation | 20 %
    - Basic UI/UX implemented
    - Responsive design partially implemented
    - Need to implement:
        - User preferences
        - Complete documentation
        - Accessibility features

- Phase 6: Testing and Optimization | 10 %
    - Basic error handling implemented
    - Need to implement:
        - Comprehensive testing
        - Performance optimization
        - Security enhancements
        - Caching system
- Phase 7: Deployment and Maintenance | 0 %
    - Set up deployment pipeline
    - Create backup and recovery system
    - Establish maintenance procedures

3. Requirements Progress

Essential:

- Natural Language Query Interpretation: Users will type questions, and the LLM-powered system will interpret them to retrieve and process relevant HESA data.
  Progress: 10%
    - Implemented basic parameter extraction from submitted queries

- Integrating and Processing Data: HESA open-source data needs to be extracted, cleaned and be well-structured so it can be analysed.
  Progress: 85%
    - Implemented CSV cleaning with metadata extraction
    - Created JSON metadata structure for enhanced searching
    - Developed keyword extraction from titles and columns
    - Added academic year standardization
    - Implemented data validation and quality checks

- Interactive Dashboard: A user-friendly interface that dynamically displays results. It can display results in different ways like tables, charts or summary reports.

Progress: 50%
    - Created responsive dashboard layout
    - Implemented dataset preview system
    - Added support for multiple file viewing
    - Developed mission group comparison interface
    - Created dynamic result display system

- Comparative Analysis: Comparing universities by grouping them by Mission Groups (Russell Group, University Alliance, etc.), and benchmarking them based on relevant metrics.
  Progress: 65%
    - Implemented mission group filtering
    - Added institution substring matching
    - Created multi-year comparison capability
    - Developed threshold-based dataset matching

- Data Export Capabilities: Users should be able to download reports in formats like (CSV, PDF, or Excel) for further use.
  Progress: 15%
    - Created underlying data structures that support export formatting
    - Implemented methods to access clean formatted data

Recommended:

- Automated Data Retrieval: The system should be able to fetch and update HESA data automatically. This will eliminate the need for manual data downloads and updates.
  Progress: 0%

- Using AI To Improve Analysis: The LLM not only fetches data but also provides analysis about it. It identifies trends and comes up with recommendations based on historical data.
  Progress: 0%

- Custom Query Building: Users who are unfamiliar with free-text queries will be provided with a quick guide to write relevant questions based on available datasets.
  Progress: 75%
    - Implemented max matches selection
    - Added year range filtering
    - Created mission group selection
    - Developed institution filtering with defaults

- Advanced Visualisation: Users can view data using advanced visualisation techniques like interactive graphs and trend projection.
  Progress: 20%
    - Created basic chart structure
    - Implemented data preparation for visualization
    - Added support for multiple chart types

- Historical Data Tracking: Keep track of changes in university performance over time and highlight important shifts and trends.

Progress: 65%
    - Implemented metadata-based storage
    - Created efficient file organization
    - Added support for academic year tracking
    - Developed data versioning system

- Caching for Frequent Queries: Implement a way to cache frequently requested queries and store them to make the next similar query more efficient and faster.
  Progress: 20%
    - Implemented query parameter extraction
    - Added synonym expansion system
    - Created threshold-based matching

- LLM Interpretation Feedback: Show users how the system interpreted their query (showing an English-like readable query).
  Progress: 0%

- Queries Logs and Protection: Log out users' queries and LLM outputs to improve future prompts. Create a protection mechanism so that big queries will not break the system.
  Progress: 70%
    - Added sample queries system
    - Implemented preview controls
    - Created responsive design
    - Added loading indicators

Optional:

- Predictive Modelling: Using historical HESA data and machine learning models to predict future university metrics.
  Progress: 0%

- Chatbot Integration: Embedding an AI chatbot within the dashboard to allow users to converse with the data and refine their queries dynamically.
  Progress: 0%

- API for External Use: Develop an API that will allow external applications to integrate HESA data analysis functionalities.
  Progress: 0%

- Multi-User Concurrency: Allows users to provide multiple queries at the same time using asynchronous job queues (like Celery) and session management.
  Progress: 0%

- Collect feedback: Receive feedback from users and use it to update the UI to make it more user-friendly. And to improve query handling and LLM prompts.
  Progress: 0%

- CSV File Validation: Create a sanitiser that validates files, scans for size anomalies and parses CSVs as plain text to prevent system breaking.
  Progress: 85%

- Implemented comprehensive CSV cleaning
- Added metadata extraction
- Created keyword processing
- Developed validation system

Essential Requirements: ~ 52% Complete
Recommended Requirements: ~ 50% Complete
Optional Requirements: ~ 13% Complete
Total Project: ~44% Complete

4. Changes

1. Instead of using GTP-J model, it will be replaced with Gemini API but since this model does not directly support free tier but rather provide number of calls that can be in certain time, this model can be replaced with different one.
2. Since the implementation of GPT-J have failed due to many reason that will be discussed in final report, I have implemented the way to process data using regular expression and calculation. This process will be replaced as I will integrate the API to automate this process.

5. Revised Gantt Chart

Progress indicators:

- ☑ **Completed** (Green)
- **In Progress** (Yellow)
- SOON **To be done** (Purple)

Phase 1: Project Initiation & Setup (Green)
Phase 2: Research, Data Processing & Preliminary Report (Blue)
Phase 3: Iterative Development & Prototyping (Purple)
Phase 4: Final Integration, Testing & Documentation (Orange)

The red line illustrating the progress of the project, from left to right.

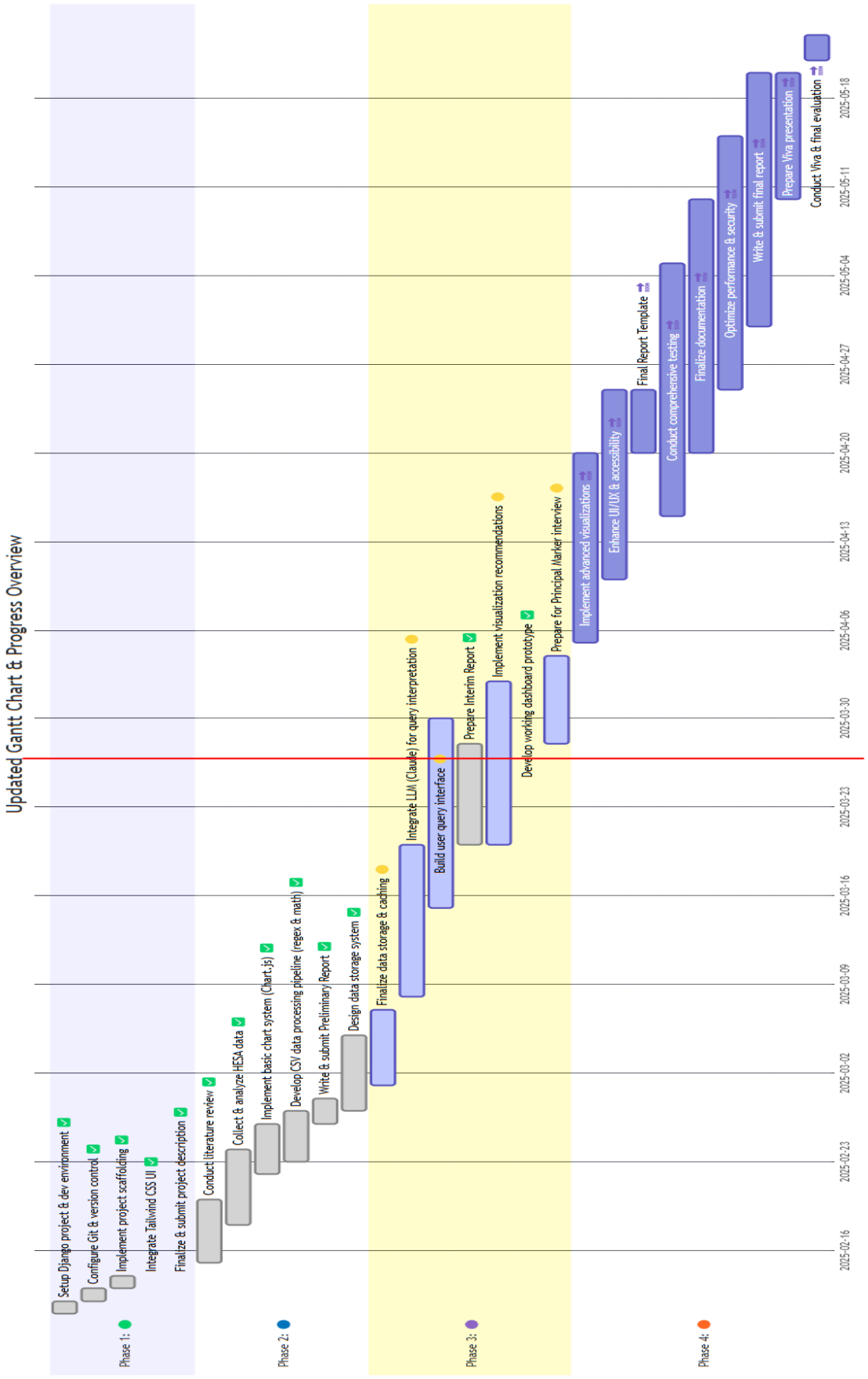

Figure 4: Updated Gantt Chart of the Project Timeline