# MSc in Computer Science - Team Project

## Project Plan

**Project Title:**

**Data GPT**

**Project Summary:**

**What are you doing?**

A web application will be developed with the goal of creating a robust and user-friendly tool that streamlines the process of cleaning, transforming, and analyzing datasets. The project will display a simple summary of the dataset highlighting basic facts and suggested issues. This will include missing values, a quick statistical overview, and suggested data types of columns. Users will then be able to make changes to their dataset and then query the dataset to provide some insights to questions they may have and export whatever final version they derived.

**Why are you doing it?**

Our primary goal is to create a data cleaning application aimed at enhancing data quality and dependability. Quality data is the foundation of informed decision-making and accurate analytics, making this tool a useful asset. Through our platform, users can complete all the data cleaning features and more in one single platform which will minimize the risk of human error, saving time and resources.

**Who will use it?**

The tool is designed for individuals from diverse industries who may not have advanced technical proficiency with data. Feedback from a survey of over 20 participants reveals that the age group of 20–30-year-olds desire an application that provides a snapshot of their data quality and has the capability to auto-correct prominent issues like repeated records. Our research indicates that, though many in this age bracket work with data, they gravitate towards user-friendly, instantaneous tools that do not necessitate in-depth technical knowledge. This data-cleaning instrument has been optimized for easy online access and is best utilized on desktop platforms, ensuring that the tasks of refining, modifying, and visualizing data remain simple and effective.

# MSc in Computer Science - Team Project

## Project Plan

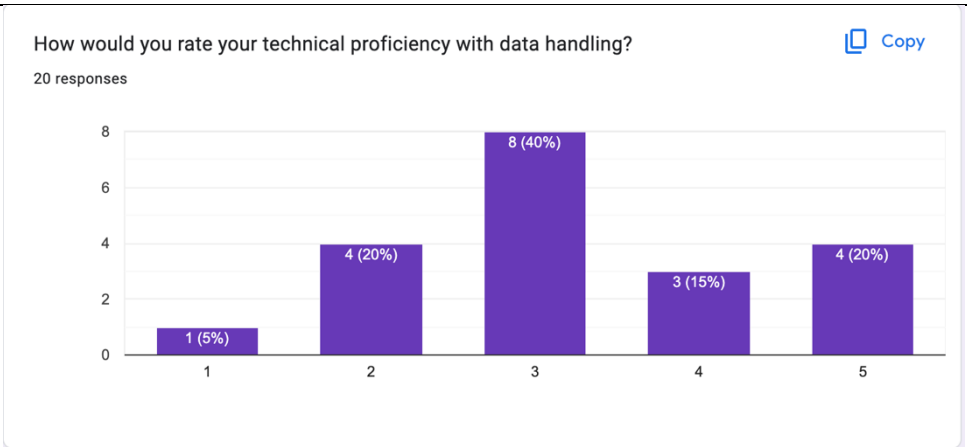


*Fig 1: Technical Proficiency of respondent*

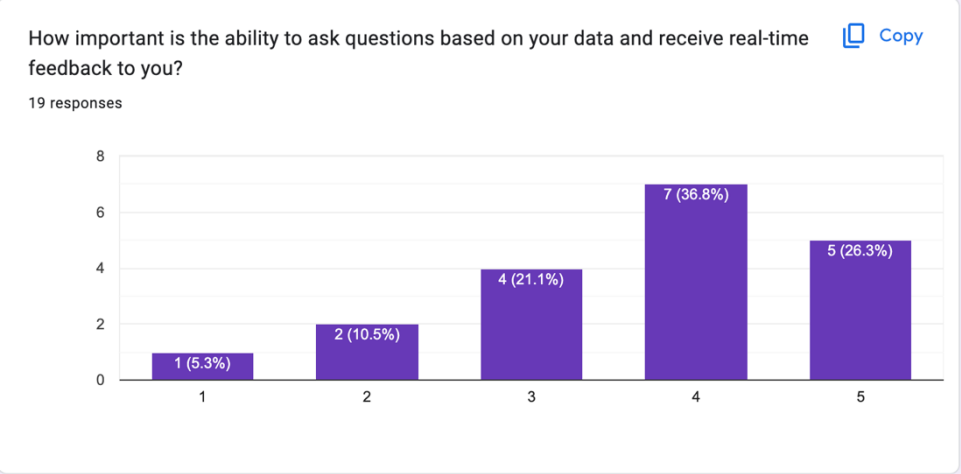


*Fig 2: How many respondents want Query GPT*

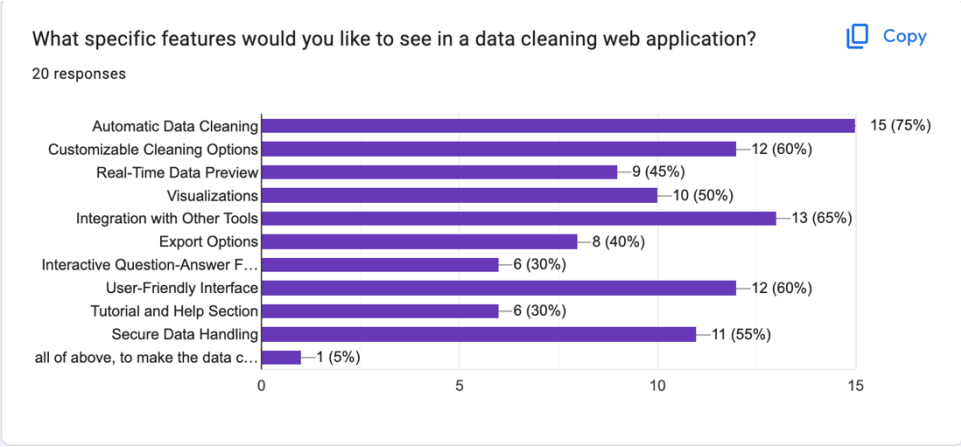


*Fig 3: Desired features for data cleaning web application*

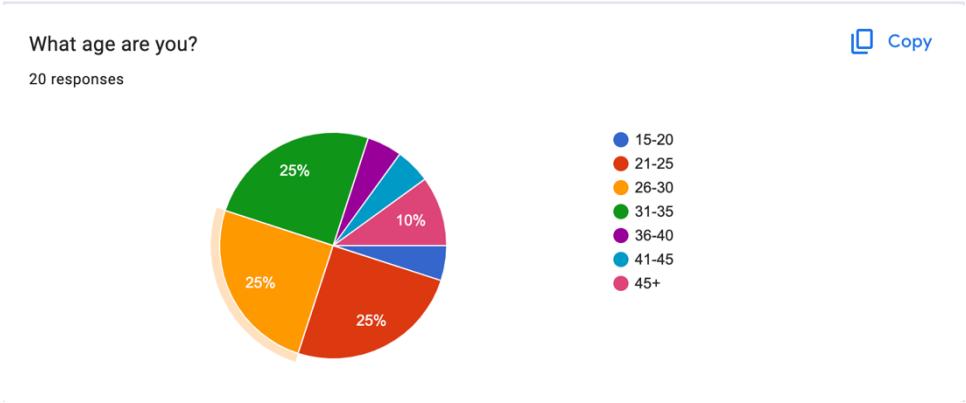


*Fig 4: Age group of respondents*

**How will they use it? (Example use case)**

1. Users will be able to import datasets using the data importing feature.
2. The dataset will be profiled, and summary statistics and data quality issues will be generated to give an understanding of the data.
3. Users can perform data cleaning using data transformation tools to reshape dataset as needed.
4. Users can preview dataset in a tabular format and additional visualizations of trends in columns for various data types.
5. Users will be able to export altered datasets to other file formats.
6. Users will be able to query the dataset to derive insights.
7. Users can learn how to use our application or troubleshoot any issues using the Tutorial and Help feature.
8. Tools like Microsoft Teams can be integrated to facilitate collaboration among team members working on data cleaning projects.
9. People with disabilities can use different versions of the UI for ease.

| FEATURE NAME | COMPLEXITY | PRIORITY |
|---|---|---|
| DATA IMPORTING | LOW | HIGH |
| DATA PROFILING | HIGH | HIGH |
| DATA CLEANING | HIGH | HIGH |
| DATA PREVIEW | MID | MID |
| INTERGRATING WITH OTHER TOOLS | HIGH | LOW |
| EXPORTING | LOW | HIGH |
| QUERY GPT | MID | MID |
| TUTORIAL | LOW | LOW |
| DATA ENCRYPTION | MID | HIGH |
| ACCESSIBILITY (DISABLED) | HIGH | LOW |

*Fig 5: Complexity and priority of each feature*

**Project Development:**

In Initial release, end user will be able to upload the dataset only CSV file, and the user can perform basic cleaning functions like delete duplication and handling missing values. Since the application is role-based, admin can create users and give necessary privileges to perform above tasks.

By deploying the application, we will be developing the knowledge of cloud-based technologies like

- AWS EC2 to run our applications on server.
- DNS to configure networking so that the application can be accessed via the internet.
- Various serverless models and compare with initial measurements of EC2 to move the applications to serverless architecture.

**How will you build your system? (System diagram)**

- **Front-end:** The User will have the ability to log-in, upload a file, have their data preprocessed based on their personal customizations, this will result in a table provided to the user and a typing search bar, in the search bar user can use a GPT model in order to ask basic questions on the data including visualizations or data query related questions. We will be using Angular framework to implement the front end.

# MSc in Computer Science - Team Project

## Project Plan


*Fig 6: Front-end Flow*

- **Back-end:**
    - Event based architecture will be used in this project.
    - Java Spring boot API will be used for backend requirements for web application. Any binary object uploaded like CSV files by the user, will be saved in BLOB storage and URI will be inserted into Oracle SQL DB.
    - Data from DB will be cached in Redis service so next time, when API needs data it can get from cache rather than DB by this implementation time to get data will be minimal.
    - AWS Key Management Service – All the keys, configs and secrets will be stored in here
    - Python flask Microservice will be running separately waiting for messages published into Kafka topic, once it receives the message service will get data from blob storage, process and save it back and DB will be updated.
    - Once data cleaning is completed by flask microservice the result will be displayed to user since all the services have access to DB and blob storage.

# MSc in Computer Science - Team Project

## Project Plan

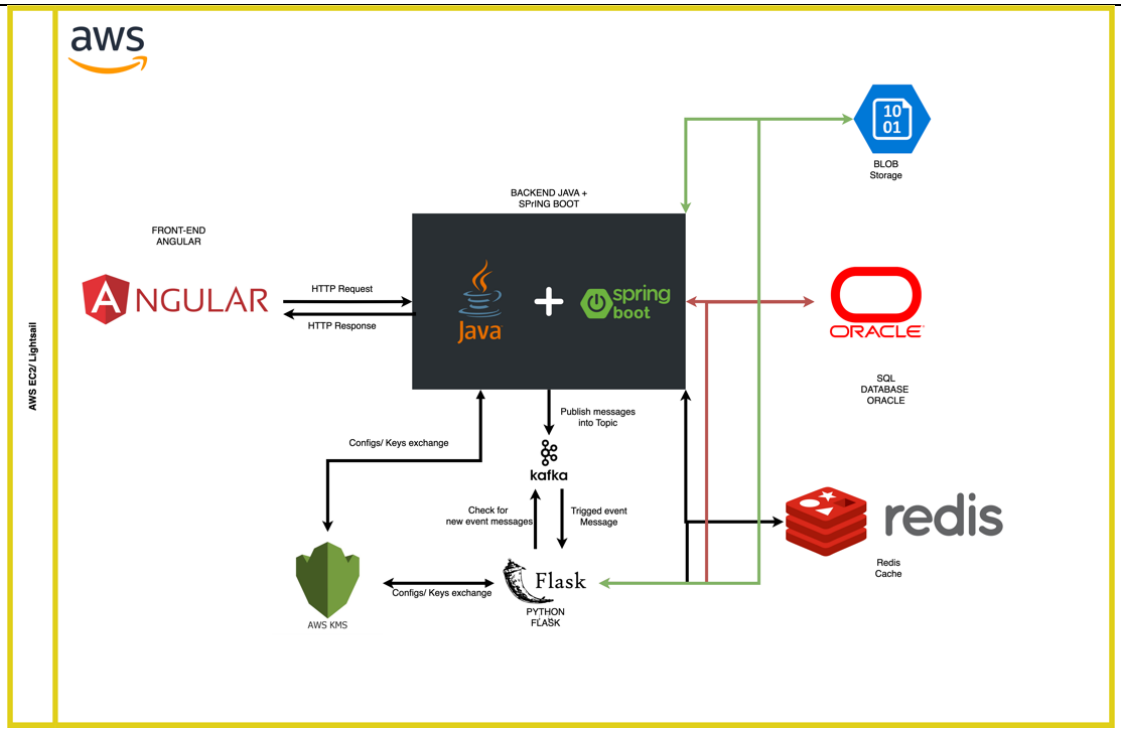


*Fig 7: System Diagram*

**Specified methodology:**

1. SCRUM

   On this project we will be using SCRUM methodology and JIRA tool to develop and release the features. After every two sprints an MVP will be released and deployed into the production environment. Project will be split into multiple sprints of one week each. All the features will be going through multiple testing phases in Development and QA environments before going to production.

2. GIT Flow

   For this project we will be using GIT flow to handle branching and releasing of code. Developers are only able to push their code to Feature, Bugfix & Hotfix branches. Once the code is pushed into feature branch developer has to raise Pull Request (PR) to merge with upper environment code base. After the code has been reviewed by one of the team members PR will be approved to merge into the upper code base, i.e., Development. Afterwards, code will be deployed and if unit testing is successful, PR for the code will be requested to QA branch by team member. Similarly, code base will reach production environment. Main branch will have backup of the last stable release if anything goes wrong with the production code then main branch code will be used. Once Production is stable and all the test cases are passed, it will be merged into the main branch.

# MSc in Computer Science - Team Project

## Project Plan

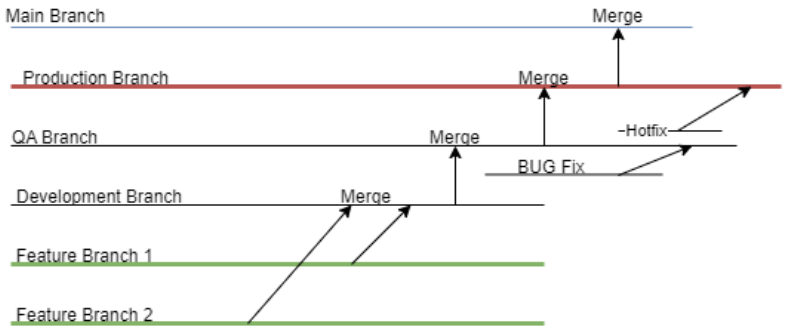


*Fig 8: Branch flow in GitHub*

**Evaluation:**

In the development of our application, an effective evaluation process is vital to ensure the ongoing enhancement of our system. Throughout our application-building process, we will consider the reviews and suggestions from the users (approx. 20).

- For the initial plan development, we conducted a survey with 8 questions to understand what features we can include in our application as well as what are the problems and expectations of users.
- We created wireframe diagrams as part of the process to construct the prototype, outlining the intended appearance and functionality of our application.
- For each release or version of the application we are going to perform unit testing and integration testing to ensure every component is working fine individually and as a whole too. Additionally, Spring Boot back-end components will be tested using Postman and front-end will be tested using automation tools like UiPath.
- Continuous data quality assessment will measure the impact of our data cleaning algorithms. By comparing pre-cleaning and post-cleaning data quality, we can gauge the system's effectiveness in enhancing data reliability.
- Throughout the journey, usability test will be performed by actual users who will interact with the system to evaluate user interface (UI) and overall user experience (UX). This will enable us to detect and resolve any usability issues, thus ensuring a more intuitive and user-friendly application.
- We will be continuously fixing bugs and issues, giving high priority to the ones reported by users.

# MSc in Computer Science - Team Project

## Project Plan

- We will closely monitor the system's performance. We will track key performance metrics such as processing speed, data cleansing accuracy, and resource utilization to ensure that the app operates efficiently and meets our performance benchmarks.

**Project Management:**

The project will run based on the SCRUM methodology commonly used for software development projects in the IT industry. The team has a mix of Advanced Software development personnel and Data science personnel. Members of the group will be expected to engage in an interdisciplinary approach to the project in which members may touch on tasks not directly associated with their stream. Below is a highlighted breakdown of project roles and roles based on the SCRUM framework.

As we are using SCRUM, the project plan will be divided into sprint cycles each lasting a week and daily stand-up meetings to denote what was done, what will be done soon, and the problems encountered and how they were solved. GitHub will serve as the project repository and chief tool for managing version edits to source code. Jira will be used the project management tool as members were already familiar with using Jira opposed to Zenhub.

| **Name** | **Student No.** | **Roles** | **Project Management Roles** | **Testing/Evaluation Roles** |
|---|---|---|---|---|
| Umama Sumlin Tasnuva | D22124465 | Frontend Dev. | Development Team | Tester |
| Naveen Maheswaran | D22124491 | Backend Dev. | Development Team | Evaluator |
| Oluwatobi Omole | D22125039 | Data Visualization | Product Owner | Evaluator |
| Sean McCrossan | D22124413 | Data Cleaning | Scrum Master/ Project Manager | Tester |
| Nikodem Adamski | C18415776 | Insight Analysis | Development Team | Tester |

The project is intended to be completed by the end of the 11$^{th}$ week to enable additional time for testing and evaluation.

# MSc in Computer Science - Team Project

## Project Plan

### References


Kaludii. (n.d.). Kaludii/CSV-Data-Cleaning-Tool: This is a Streamlit app that helps users preprocess and clean CSV files quickly and stress-free. Retrieved from https://github.com/Kaludii/CSV-Data-Cleaning-Tool

Cleanlab. (n.d.). cleanlab/cleanlab: The standard data-centric AI package for data quality and machine learning with messy, real-world data and labels. Retrieved from https://github.com/cleanlab/ HYPERLINK "https://github.com/cleanlab/cleanlab"cleanlab

Datacleaner. (n.d.). datacleaner/DataCleaner: The premier open source Data Quality solution. Retrieved from https://github.com/datacleaner/DataCleaner

Pyjanitor-devs. (n.d.). pyjanitor-devs/pyjanitor: Clean APIs for data cleaning. Python implementation of R package Janitor. Retrieved from https://github.com/pyjanitor-devs/pyjanitor

Yobulkdev. (n.d.). yobulkdev/yobulkdev: Open Source & AI driven Data Onboarding Platform: Free flatfile.com alternative. Retrieved from https://github.com/yobulkdev/yobulkdev

Huzztech. (n.d.). huzztech/laravel-6-full-project: Laravel 6 full project with login, authentication, register, create | update | delete record through forms. Retrieved from https://github.com/huzztech/laravel-6-full-project

PySimpleGUI. (n.d.). PySimpleGUI/PySimpleGUI: Launched in 2018. It's 2023 and PySimpleGUI is actively developed & supported. Retrieved from https://github.com/PySimpleGUI/PySimpleGUI

Alan-turing-institute. (n.d.). alan-turing-institute/CleverCSV: CleverCSV is a Python package for handling messy CSV files. Retrieved from https://github.com/alan-turing-institute/CleverCSV

(n.d.). Retrieved from https://colab.research.google.com/drive/1ZnO-njhL7TBOYPZaqvMvGtsjckZKrv2E?usp=sharing

# MSc in Computer Science - Team Project

## Project Plan

**Team Name:**

**Fabulous Five**

**Team Members:**

| Name | Student Number | Contact Number |
|---|---|---|
| **Umama Sumlin Tasnuva** | **D22124465** | **+353894400174** |
| **Naveen Maheswaran** | **D22124491** | **+353894495761** |
| **Oluwatobi Omole** | **D22125039** | **+353892214845** |
| **Sean McCrossan** | **D22124413** | **+353879466065** |
| **Nikodem Adamski** | **C18415776** | **+353030131662** |

**Team Meetings:**

**Meeting Schedule**

Our meeting schedule draws from the Atlassian Agile Manifesto (West, 2019). We have set daily stand-ups from Monday to Thursday, where each participant addresses three core questions:

- What tasks were completed yesterday?
- What's on your agenda for today?
- Are there any obstacles hindering your progress?

Additionally, the schedule includes a 1-hour consultation with our project mentor every Monday, collective team sessions of 1.5 hours on both Monday and Thursday, and a personal 1-hour interaction on Friday. This structure might evolve based on the project's demands—we aim to eliminate unnecessary meetings. Presence at these discussions is mandatory for all team members. Absences must be justified and approved by most team members.

# MSc in Computer Science - Team Project

## Project Plan

| Weekly Sprint Meeting Schedule | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
| 09:00 | | | | | | | |
| 10:00 | | Daily stand up | | | | | |
| 11:00 | Mentor | | | | Lecture | | |
| 12:00 | Sprint Planning | | | | | | |
| 13:00 | | | | | | | |
| 14:00 | | | | | Sprint review | | |
| 15:00 | Bi Weekly All hands | | | Bi Weekly All hands | Sprint retrospective | | |
| 16:00 | | | | | | | |
| 17:00 | | | | | | | |
| 18:00 | | | | | | | |

*Fig 9: Weekly Sprint Meeting Schedule*

Development-specific meetings will be set up as needed by the relevant team members. Although certain members might not be directly involved in some meetings, they are welcome to join to stay updated on project facets. Scheduled meetings will take place online via MS Teams and MS teams will also be used for spontaneous collaboration.

We also have the face-to-face session slated for Fridays. Additionally, there is a space reserved on campus for any impromptu meetings if team members prefer an in-person discussion. Most decisions will typically be settled by a simple majority. However, there will be scenarios where either a unanimous decision or specialized expertise is required. Such situations will be determined by a majority vote on a case-by-case basis. Roles, including that of the scrum master and other project management positions, will circulate among team members weekly, following the sequence in the previously mentioned team member table.

**Team Conflict:**

Discuss issues like:

- How we as a group deal with the habits of individual members is we sit down on weekly catchup and make sure that the deadlines we have can be completed and our work does

not interfere with our life outside of college. We decide on the time that suits all meetings and compromise to video call for members who cannot attend in person.

- We address unresolved issues during daily catch-up. This is a crucial part of our work culture as minimizing inter team conflicts maximizes our total output.
- If a conflict arrives, we all listen to both parties that are involved with the conflict and resolve the issue quickly. Most issues arise from miscommunication as our group is diverse in its cultures, we must all understand that cultural differences are our greatest strength as it allows for a plethora of viewpoints on subjects which are not restricted to one viewpoint.
- We will attempt to avoid conflict by communicating efficiently, we have adopted six hat methods and voting to address critical decision-making process. This allows us to understand both parties and decide efficiently and effectively.
- We do not have any ultimate VETO; all conflicts shall be resolved through a democratic voting system.