SDM404 - Assessment 1 – Software Project Proposal and Plan	
Project Name	TTrack – Degree Tracker
Group #	#1
Group Members Names	Hussain Jameel – **A00180177** Luis Guilherme de Barros Andrade Faria - **A00187785** Nomayer Hossain - **A00176827** Rosa Carolina Cortes Galvis – **A00193201** Victor Javier Dorantes Meneses – **A00179705**
Version	1.0.0
Date Created	24, June, 2025

Table of Contents

1. Introduction

1.1. Document Overview

This Software Project Management Plan (SPMP) outlines the proposal, planning, and execution strategy for the ***TTrack - Degree Tracker*** project, developed for Torrens University as part of the SDM404 Software Development Management subject.

The document serves to define the project's scope, objectives, development methodology, technical plan, schedule, budget, and risk management strategies.

It includes the following sections: Introduction, Project Proposal, Project Management Plan, Bibliography, and Contribution Log, ensuring comprehensive coverage of all aspects required for successful project presentation, details and delivery.

1.2. Definitions / Glossary

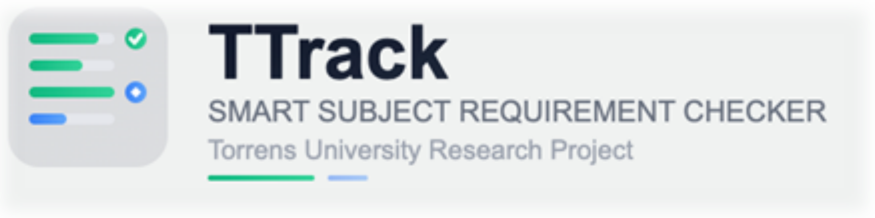
Term	Definition
SDM404	Software Development Management subject we are studying
SPMP	Software Project Management Plan
TTrack	The proposed desktop application for tracking degree progress by comparing academic transcripts with curriculum requirements
Transcript	An Excel file containing a student's completed courses and grades
Curriculum	An Excel file defining the required core, specialization, and elective subjects for a degree
GUI	Graphical User Interface
PyQt5	Python library for desktop GUI apps.
Matching Engine	The core logic that compares transcript vs. curriculum
Dev	Development

2. Project Proposal

2.1. Title of the project

TTrack – Degree Tracker

Figure A1: TTrack Logo and Concept - Represents the visual identity of the TTrack Degree Tracker application.


TTrack
SMART SUBJECT REQUIREMENT CHECKER
Torrens University Research Project

2.2. Problem Description

Torrens admin staff and students face a recurring challenge: tracking degree progress and ensuring no critical core or elective subjects are missing. Despite transcripts and course handbooks existing in separate silos, there is no smart, automated way to cross-check course history against degree requirements. The current manual process is error-prone and frustrating, leading to delays and potential graduation issues.

2.3. Proposed Solution

TTrack is a desktop application that addresses these challenges by automating the comparison of academic transcripts with degree curriculum requirements. It parses Excel files containing transcript and curriculum data, validates completed versus pending subjects, and provides a visual summary of academic progress.

The application operates offline, requiring no internet or admin intervention, thus offering clarity and efficiency. The business value includes reduced manual workload for academic advisors, faster degree audits for course coordinators, and enhanced visibility for students into their degree path, ultimately improving graduation accuracy and user satisfaction.

2.4. Features of the Product / Application

- Upload & parse transcript and curriculum Excel files (.xlsx)
- Match completed units to core, specialization and elective subjects
- Display subject status as Done, Missing or Invalid
- Provide a visual dashboard summarizing academic progress
- Elective suggestion engine based on unmet credits or required categories

3. Project Management Plan

3.1. Scope of the project

The purpose of *TTrack* is to deliver a desktop application that automates degree progress tracking for Torrens University students and staff. The objectives include developing a functional application that parses Excel files, matches subjects, visualizes progress, and suggests electives. The deliverables are a functional desktop app with a GUI, a subject matching engine, a visual dashboard, and comprehensive codebase documentation.

Factors affecting maintainability, as described by Ahmad et al. (2013), informed and influenced our decision to use a simplified UI and well-structured codebase.

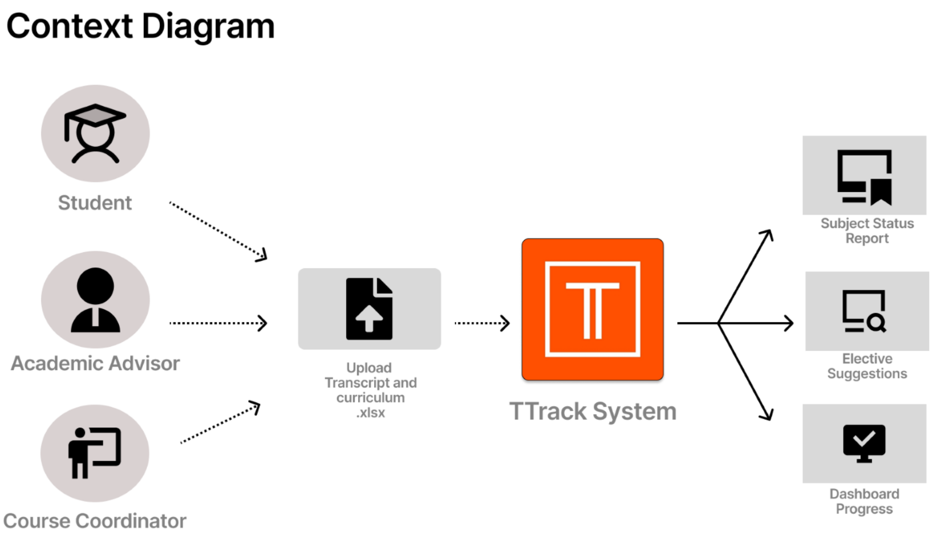
In Scope:

- File upload and parsing for transcript and curriculum Excel files
- Subject matching logic for core, specialization, and elective categories
- Status indicators ('done', 'missing', 'invalid') in the UI
- Elective suggestion engine based on curriculum gaps
- Visual dashboard with progress summaries (e.g., pie charts, bar charts)

Out of Scope:

- Real-time integration with academic systems
- Web or mobile versions of the application

Figure A2: Context Diagram of *TTrack* - Illustrates how TTrack integrates with users, inputs and outputs to automate transcript analysis and curriculum matching.



3.2. Assumptions, Constraints and Risks

Assumptions:

- Transcript and curriculum Excel files follow a consistent structure with identifiable subject codes and credits.
- Users have basic proficiency in navigating desktop applications.
- The project can be completed within the 12-week semester timeline.

Constraints:

- Schedule: The project must be completed by Week 12 of the semester (July 31, 2025).
- Resources: Limited to the five team members with no external funding.
- Technology: Offline operation restricts use of real-time APIs or cloud services.

As discussed by Ewusi-Mensah (2003), poor project planning is a frequent cause of failure in software development, which we aim to mitigate through proactive risk analysis. Our analysis details can be found at table below:

Date	**Risk Description**	**Likelihood (1…5)**	**Impact (1…5)**	**Owner**	**Mitigating Action**
24/06/2025	Inconsistent excel file formats	3	5	**Luis**	Define strict input format guidelines; implement robust error handling
24/06/2025	Team member unavailability	1	3	**All**	Cross-train team member; maintain detailed documentation
24/06/2025	UI complexity overwhelms users	3	3	**Rosa**	Conduct usability testing; simplify UI design based on feedback
26/06/2025	Scope creep (additional features requested)	3	5	**Nomayer**	Define and freeze feature scope during Sprint 2 to prevent uncontrolled expansions. Review and freeze features before Sprint 3.
26/06/2025	Tech unfamiliarity (e.g., Supabase)	5	3	**Victor**	Assign learning phase in Sprint 1. Use fallback local DB if needed.
26/06/2026	Data Scalability (large Excel files)	1	5	**Hussain**	Optimize parsing with pandas chunking, test with files up to 5000 rows, and implement caching or rate limiting mechanisms.

3.3. Software Development Process

As noted by Cobb (2015), Agile practices such as iterative delivery improve adaptability to change, making Scrum a suitable choice for this project. The team adopted Scrum with 2-week sprints to enable rapid feedback, iterative development, and continuous refinement of both UI and logic. This approach also aligns with the team's schedule and supports incremental delivery, which is critical in dynamic academic settings like this.

The process includes:

- Backlog: Prioritized tasks defined in a GitHub project board.
- Sprint Planning: Bi-weekly meetings to assign tasks and set sprint goals.
- Development: Coding, testing, and integration of features.
- Review and Retrospective: Demo features and reflect on process improvements.

3.4. Technical Plan

By following the best practices described by Stephens (2015), *TTrack's* architecture emphasizes modularity and maintainability. The plan is to develop the application using two technology stacks, with a final decision pending prototype evaluation:

- Python Stack:
    - Python 3.10+ for core logic
    - PyQt5 for desktop GUI
    - pandas and openpyxl for Excel processing
    - matplotlib for dashboard visualizations
    - SQLite (optional) for local persistence
    - PyInstaller for packaging as .exe/.app

- Node.js Stack:
    - ElectronJS for desktop app packaging
    - ReactJS for interactive UI
    - Node.js for backend logic and file parsing
    - xlsx (npm) for Excel file handling
    - Chart.js for dynamic visualizations
    - SQLite or JSON for local data persistence

Testing tools include Pytest (Python) or Jest (Node.js) for unit testing, and manual usability testing with mock users.

3.5. Quality Management Plan

To ensure high software quality, we adopted a layered approach incorporating coding standards, peer reviews, and automated testing. The core logic (engine) is tested using unit tests built with Pytest, focusing on matching accuracy and data validation. Usability testing was scheduled in Sprint 4 to gather peers' feedback on the interface and improve the user experience iteratively.

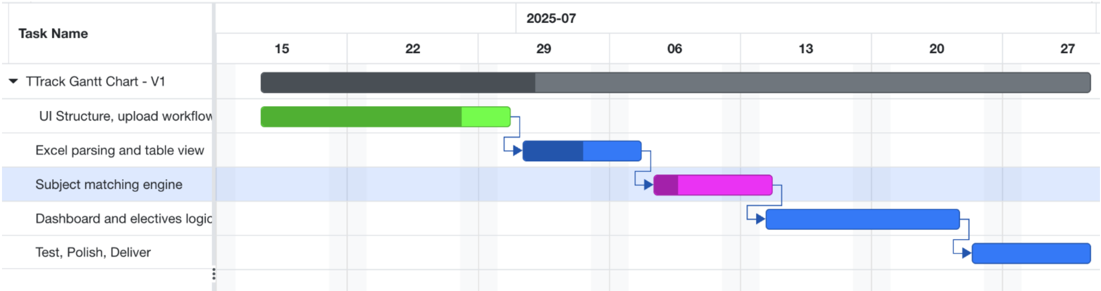
Code reviews are conducted via GitHub pull requests to maintain codebase consistency and spot early bugs. We also enforced documentation standards within each module (docstrings, in-code comments) to support maintainability. Quality gates were defined per sprint to ensure no major regressions occurred before release to the results screen.

3.6. Project Schedule

The project spans 12 weeks, divided into five 2-week sprints, aligning with the semester timeline ending July 31, 2025 as detailed on following table:

Milestone	Task	Dependency	Owner	Start	End
Sprint 1	UI structure + upload workflow	None	Rosa	17/06	23/06
Sprint 2	Excel parsing + table view	Sprint 1	Victor	24/06	30/06
Sprint 3	Subject matching engine	Sprint 2	Luis	01/07	14/07
Sprint 4	Dashboard + electives logic	Sprint 3	Hussain	15/07	25/07
Sprint 5	Test, polish and deliver	Sprint 4	All	26/07	31/07

Figure A3: Gantt Chart Figure - A timeline visualization of TTrack project sprints, tasks and key delivery dates.



3.7. Project Budget Estimates

The budget is based on estimated hours per role, aligned with the Scrum process and semester timeline, and the effort was estimated based on historical data from past subject workload, divided into five sprints.

The tasks were broken into story points and converted into estimated hours using team consensus.

Role	Hours	Rate	Cost
Project Manager	30	AUD 45	AUD 2,100
Frontend Dev	40	AUD 40	AUD 2,400
Backend Dev	45	AUD 40	AUD 2,700
DBM	35	AUD 42	AUD 1,575
Fullstack Dev	40	AUD 42	AUD 2,400
TOTAL	**180**		**AUD 11,175**

3.8. Resource Allocation Plan

We have carefully defined the following Roles and Responsibilities for the Team Members:

Team Member	**Role**	**Main Tasks**	**Primary Tools**
Luis Faria	Full-Stack Dev + Project Manager	Architecture, UI, File Parsing, Engine	Python, PyQt5, pandas, GitHub
Hussain Jameel	Full-Stack Dev (Alt)	ElectronJS Prototype, frontend dashboard	NodeJS, React, Electron, Chart.js
Rosa Galvis	UI/UX Designer	Wireframes, UI themes, dark mode	Figma, PyQt5
Victor Dorantes	DB & Backend Integrator	Supabase setup, curriculum schema definition	PostgreSQL, Supabase, pgAdmin
Nomayer Hossain	QA / Stakeholder Liaison	Functional testing, professor communication	Excel, QA test plan, GitHub Issues

4. Bibliography

- Ahmad, M. O., Markkula, J., Oivo, M. (2013). Factors affecting software maintainability from customer perspective. Journal of Systems and Software.
- Cobb, C. G. (2015). The Project Manager's Guide to Mastering Agile. Wiley.
- Ewusi-Mensah, K. (2003). Software Development Failures. MIT Press.
- Heath, F. (2021). The Professional Scrum Master Guide. Packt Publishing.
- Schwaber, K., & Sutherland, J. (2020). The Scrum Guide.
- Stephens, R. (2015). Beginning Software Engineering. Wrox Press.

5. Contribution Log

Student Name	Contribution
Luis Faria	**Tasks Performed:** As a full-stack developer, I led the architecture design and core implementation of TTrack. My responsibilities included setting up the file parsing logic using pandas and openpyxl, configuring the PyQt5 UI for local use and orchestrating the subject matching engine logic to compare completed subjects with curriculum requirements. **Challenges:** The biggest challenge was mapping course codes accurately between transcript and curriculum sheets, especially considering inconsistent formatting in real-world transcripts. I also faced complexity when trying to keep the UI simple while still delivering rich information, such as course status and suggestions. **Lessons Learned:** This project helped me apply software engineering principles in a real-life context. I deepened my skills in Agile project management, version control (via GitHub) and GUI frameworks. Working in sprints helped us deliver incrementally and handle scope adjustments without losing momentum. I also learned how critical it is to clarify data assumptions early, especially when working with unstructured inputs like transcripts. And finally, collaborating in a multidisciplinary group sharpened my communication and coordination skills, which I consider key in project's delivery.
Nomayer Hossain	**Tasks Performed:** Developed a comprehensive QA Test Log covering all major functionalities: file upload, subject matching, UI dashboard, and elective suggestions. Conducted manual tests for functionality and edge cases (e.g., missing electives, incomplete data). Logged results and confirmed accuracy of progress indicators and recommendation engine outputs. **Challenges:** Some test scenarios revealed mismatches due to inconsistent Excel formats from stakeholders. **Lessons Learned:** Gained hands-on experience in preparing QA documentation aligned with Agile development. Learned how early user feedback can shape functional expectations and avoid scope creep. Improved ability to communicate technical results clearly to both technical and non-technical stakeholders.

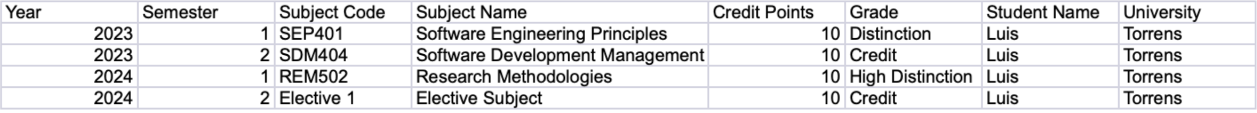
Hussain Jameel	**Tasks Performed:** As a full-stack developer in our group project, I focused on building the ElectronJS prototype. My main responsibility was UI development and integrating a dynamic dashboard interface. I created key components that allowed users to navigate smoothly between views and interact with real-time data. This involved using modern JavaScript, React, and Electron APIs to ensure a responsive and user-friendly experience. **Challenges:** One of the main challenges I tackled was finding a reliable cross-platform solution. I conducted research on various frameworks, including ElectronJS and its alternatives such as Tauri and NW.js, to evaluate their suitability for desktop applications. I created an initial prototype using ElectronJS and tested it across different systems. This included running builds for Windows and macOS, ensuring the application functioned properly in both development and production environments. **Lessons Learned:** This process taught me the value of evaluating multiple approaches before committing to a stack. Understanding the differences in system behavior helped me make more informed technical decisions. It also gave me experience in managing builds and platform-specific configurations and strengthened my skills in desktop app development and cross-platform deployment.
Rosa Galvis	**Tasks Performed:** In my role as UI/UX designer, I was responsible for developing the initial wireframes and the project's visual structure. To ensure the interface was intuitive and accessible for users, the file upload flow was implemented using PyQt5. I also developed a visual design, including dark mode, to improve the user experience and reduce eye strain. **Challenges:** Achieving a balance so that the design was appealing, intuitive, and functional for all users was one of the biggest challenges. I took into account the team's feedback to reduce interface complexity and optimize usability. Additionally, adapting the wireframes to the PyQt5 framework required additional learning and adjustments to maintain visual consistency. **Lessons Learned:** Through this project, I was able to delve deeper into UI/UX design principles by applying them to a real-world software project. I learned how to use design tools like Figma and translate those designs into practical implementations in PyQt5. I also improved my skills in collaborating closely with developers, integrating technical and aesthetic aspects into a functional product.

Victor Dorantes	**Tasks Performed:** As the database manager, I set up the data schema and implemented Excel parsing using pandas and openpyxl. **Challenges:** Encountered issues with normalizing inconsistent Excel data and ensuring efficient data processing. Faced issues while testing cells containing non-ASCII characters. Found opportunity to learn technologies that allow to connect to SQL server directly within python. **Lessons Learned:** Learned the importance of robust data validation and schema design in handling real-world data. Learned to use PyQT to deploy python projects as apps. Learned to integrate openpyxl along pandas library to ensure we take advantage of Excel features.

6. Appendices

6.1. Sample Transcript and Curriculum

Figure A4: Transcript Sample File - Sample table layout used to store completed subjects, grades and credits for a student.

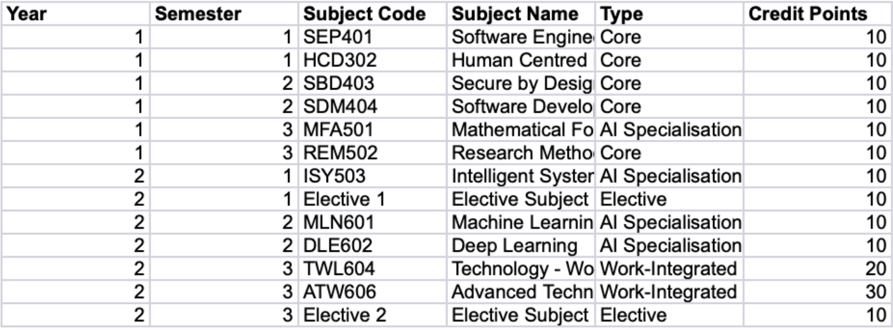


Year	Semester	Subject Code	Subject Name	Credit Points	Grade	Student Name	University
2023	1	SEP401	Software Engineering Principles	10	Distinction	Luis	Torrens
2023	2	SDM404	Software Development Management	10	Credit	Luis	Torrens
2024	1	REM502	Research Methodologies	10	High Distinction	Luis	Torrens
2024	2	Elective 1	Elective Subject	10	Credit	Luis	Torrens

Download link for Academic Transcript *.xlsx* file:

- https://docs.google.com/spreadsheets/d/1UU8IYJF7eGxBe_l2ygdObkcxZae4irP6B74AdtAK9wQ

Figure A5: Transcript Sample File - Sample table layout detailing required core, specialization and electives for a degree.



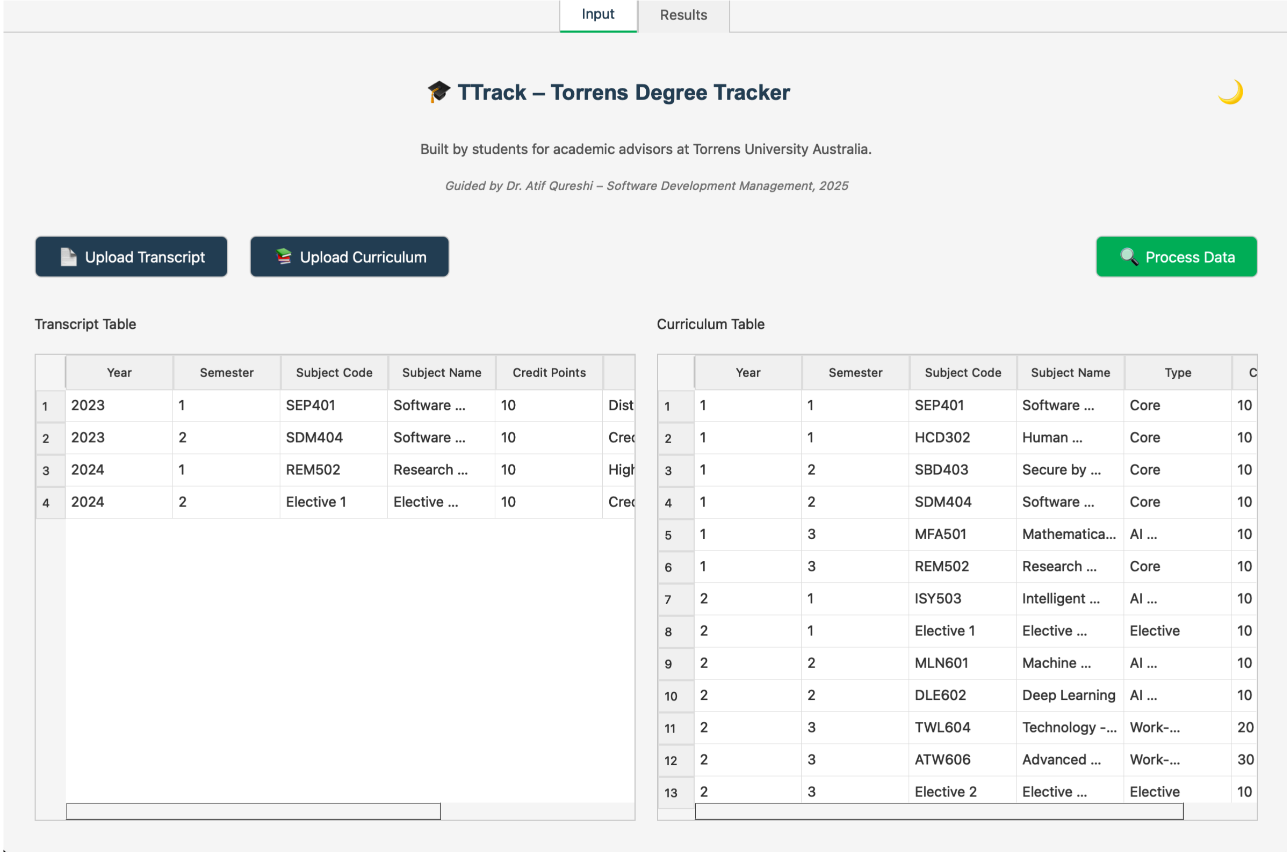
Year	Semester	Subject Code	Subject Name	Type	Credit Points
1	1	SEP401	Software Engine	Core	10
1	1	HCD302	Human Centred	Core	10
1	2	SBD403	Secure by Desig	Core	10
1	2	SDM404	Software Develo	Core	10
1	3	MFA501	Mathematical Fo	AI Specialisation	10
1	3	REM502	Research Metho	Core	10
2	1	ISY503	Intelligent Syster	AI Specialisation	10
2	1	Elective 1	Elective Subject	Elective	10
2	2	MLN601	Machine Learnin	AI Specialisation	10
2	2	DLE602	Deep Learning	AI Specialisation	10
2	3	TWL604	Technology - Wo	Work-Integrated	20
2	3	ATW606	Advanced Techn	Work-Integrated	30
2	3	Elective 2	Elective Subject	Elective	10

Download link for Course Curriculum *.xlsx* file:

- https://docs.google.com/spreadsheets/d/1PdzsKB8ocSxMn3XVQ3HaaTXIEkeVMTqS9BSKoZNrOkc

6.2. Screenshots

Figure A6: UI Input Tab - Displays the user interface section for uploading and previewing transcript and curriculum data files.



🎓 TTrack – Torrens Degree Tracker

Built by students for academic advisors at Torrens University Australia.

Guided by Dr. Atif Qureshi – Software Development Management, 2025

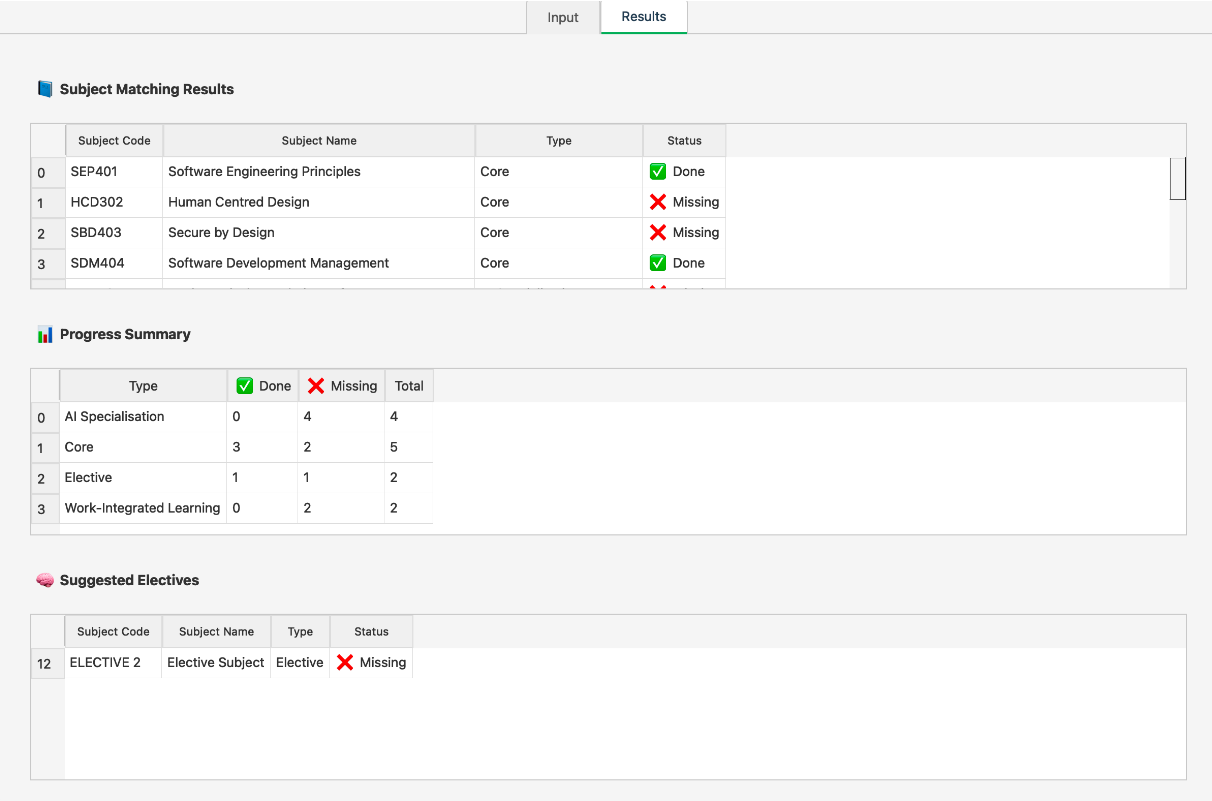
Transcript Table

	Year	Semester	Subject Code	Subject Name	Credit Points	
1	2023	1	SEP401	Software ...	10	Dist
2	2023	2	SDM404	Software ...	10	Cred
3	2024	1	REM502	Research ...	10	High
4	2024	2	Elective 1	Elective ...	10	Cred

Curriculum Table

	Year	Semester	Subject Code	Subject Name	Type	C
1	1	1	SEP401	Software ...	Core	10
2	1	1	HCD302	Human ...	Core	10
3	1	2	SBD403	Secure by ...	Core	10
4	1	2	SDM404	Software ...	Core	10
5	1	3	MFA501	Mathematica...	AI ...	10
6	1	3	REM502	Research ...	Core	10
7	2	1	ISY503	Intelligent ...	AI ...	10
8	2	1	Elective 1	Elective ...	Elective	10
9	2	2	MLN601	Machine ...	AI ...	10
10	2	2	DLE602	Deep Learning	AI ...	10
11	2	3	TWL604	Technology -...	Work-...	20
12	2	3	ATW606	Advanced ...	Work-...	30
13	2	3	Elective 2	Elective ...	Elective	10

Figure A7: UI Results Tab - Shows matched subjects and progress indicators, including Done, Missing and elective recommendations.



📘 **Subject Matching Results**

	Subject Code	Subject Name	Type	Status
0	SEP401	Software Engineering Principles	Core	✅ Done
1	HCD302	Human Centred Design	Core	❌ Missing
2	SBD403	Secure by Design	Core	❌ Missing
3	SDM404	Software Development Management	Core	✅ Done

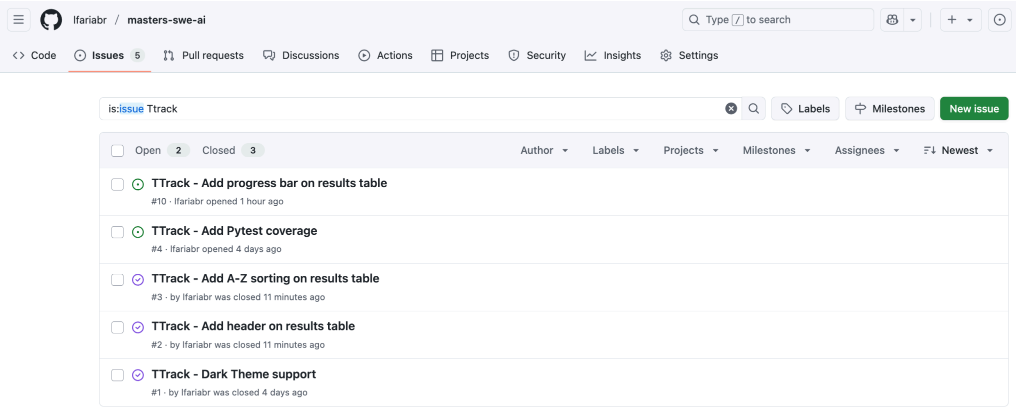
📊 **Progress Summary**

	Type	✅ Done	❌ Missing	Total
0	AI Specialisation	0	4	4
1	Core	3	2	5
2	Elective	1	1	2
3	Work-Integrated Learning	0	2	2

🧠 **Suggested Electives**

	Subject Code	Subject Name	Type	Status
12	ELECTIVE 2	Elective Subject	Elective	❌ Missing

Figure A8: Github Issue board - Demonstrates the Agile task tracking and sprint management for TTrack's development workflow.



is:issue Ttrack

- TTrack - Add progress bar on results table
  #10 · lfariabr opened 1 hour ago
- TTrack - Add Pytest coverage
  #4 · lfariabr opened 4 days ago
- TTrack - Add A-Z sorting on results table
  #3 · by lfariabr was closed 11 minutes ago
- TTrack - Add header on results table
  #2 · by lfariabr was closed 11 minutes ago
- TTrack - Dark Theme support
  #1 · by lfariabr was closed 4 days ago

Open source link for TTrack's *GitHub* repository:

- github/projects/TTrack_v1