UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Faculty of Engineering and Information Sciences
School of Computing & Information Technology

CSIT321: Project

Capstone Project - Spring 2025

Initial Project Description

Team Members

Noor Ahmed - 7295297
Nishad Gyawali - 8124553
Chelsea Okan - 8438675
Azwad Ahnaf Zihan - 7663055

University of Wollongong

Liverpool Campus

Subject Coordinator

Dr. Khin Win

29th August, 2025

Task Allocation (Project Description)

At the start of the project, responsibilities were divided among team members. The table below records the initial allocation, with notes indicating where sections were later shortened, revised, or re-written to maintain consistency.

Section	Primary Author (Assigned)	Notes (Revisions / Secondary)
1. Initial Problem Description	Chelsea	–
2. Meeting Details	Chelsea	Shortened and tightened wording.
3. Initial Project Scope	Chelsea	Shortened for clarity.
4. Scope Management	Azwad	Sections 4.4–4.7 re-written and polished by Chelsea.
5. Project Need	Chelsea	Shortened for conciseness.
6. Market Analysis	Noor	Light edits by Chelsea for clarity.
7. Time Management	Nishad	Re-written and expanded by Chelsea for completeness.
8. Quality Management Plan	Nishad	Re-written and expanded by Chelsea for completeness.
9. Risk Management	Chelsea	Shortened and refined for clarity.
10. Initial Identification of Skills	Chelsea	Shortened and refined for conciseness.
11.1 Stakeholder Analysis	Azwad	Initial four stakeholders listed by Azwad; expanded and power-interest grid created by Chelsea
11.2 Group Project Charter	ChelseaNoor	Chelsea and Noor both drafted individual versions. Final combined and refined version authored by Chelsea.
11.3 Group Ethical Approach	Noor	Edited by Chelsea for flow and consistency.
11.4 GitHub Link	Chelsea	Repository created, configured, and maintained by Chelsea;link and commit history included in appendix for reference.

11.5 Meeting Agendas & Minutes	Chelsea	–
References	Chelsea	Collated and formatted all references.

Colour Coding Key

- Chelsea
- Noor
- Azwad
- Nishad

Note: All document editing, integration, and quality assurance were completed by Chelsea to ensure consistency of style, tone, and structure across sections. Noor additionally assisted with minor QC by suggesting where sections could be shortened, pointing out misnumbering, proposing word count adjustments, and drafting material (e.g., for Section 11.2) to help cover gaps before final integration.

Table of Contents

1. Initial Problem Description

The volume of scholarly output has expanded to the point that “keeping up” with the literature is operationally difficult for most research teams. The global production runs to millions of articles per year, compounding year on year across scientific and technical fields (Our World in Data, 2024). This growth has thus translated into large screening backlogs for even modest scoping or systematic reviews, in which teams must triage large sets of titles, abstracts and full texts in order to identify studies of relevance.

The prevailing workflow – database search, title/abstract screening, full-text screening, manual note-taking, and ad-hoc thematic grouping – exhibits predictable frictions. Title/abstract screening can be relatively rapid (often sitting at around 60-120 records per hour under calibration conditions), but the subsequent full-text skim and evidence extraction stages are markedly slower and more variable (Booth, 2023; Cochrane, 2024). As a result, comprehensive reviews commonly take many months and substantial person-hours to complete, with empirical studies documenting long end-to-end timelines from protocol to publication (Borah et al., 2017; University of Arizona Libraries,2022).

Current toolchains only partially alleviate these pressures. Reference managers and scholarly search engines return lists, not structured understanding; single-document AI summarisers like Scholarcy compress one paper at a time; network-mapping tools like ResearchRabbit and Connected Papers visualise relationships but do not generate standardised digests; and assistants such as Elicit blend search and extraction without providing a batch-first, corpus-level organisation of outputs. In short, there is no integrated workflow that both (a) produces consistent, reviewer-ready summaries at scale and (b) organises a corpus into interpretable topic clusters that are suitable for navigation and synthesis (Scholarchy, 2025; ResearchRabbit, 2025; Connected Papers, 2025; Elicit, 2025).

This deficit has led to material consequences. First, lead-time to analysis is extended by repetitive low-leverage work (re-parsing boilerplate across dozens or hundreds of PDFs). Second, consistency suffers; summaries may vary by reviewer and session, undermining comparability and reuse. Third, discovery is impaired: flat lists obscure thematic structure, which makes related studies harder to recognise early. Collectively, these factors increase screening costs, delay synthesis, and heighten the risk of missed evidence.

Problem statement. Research teams lack a scalable, reliable means to transform large sets of academic PDFs into consistent, structured summaries and interpretable topic clusters. This gap drives excessive screening time, inconsistent evidence capture, and missed thematic connections, thereby degrading the efficiency and quality of downstream synthesis.

2. Meeting Details

2.1 Cadence

Day-to-day discussion occurs asynchronously on Discord. Formal Zoom check-ins are held for milestones, with minutes stored in GitHub `/docs/weekly-minutes/` as the single source of truth.

2.2 Initial client meeting (Kick-off)

Date/Time: 18 Aug 2025, 12:40–13:30 AEST (Zoom)
Attendees: Advisor – Dr Jack Yang; Team – Noor Ahmed, Nishad Gyawali, Chelsea Okan, Azwad Ahnaf Zihan

Purpose: Confirm repository of record, capture functional requirements, and agree on immediate next steps for A1.

Discussion: GitHub confirmed as the central repository for all artefacts. Frontend requirements: PDF upload, document browsing, summaries, keyword display, clustering. Backend requirements: storage, metadata management, profile handling, and use of open-access PDFs. The team agreed to document features and prepare a small sample pack for prototyping.

Action Items:

- A1: Set up GitHub repo (Chelsea, 20 Aug).
- A2: Document frontend features (Noor, 27 Aug).
- A3: Document backend features (Nishad, 27 Aug).
- A4: Prepare 10 sample PDFs with licence notes (Azwad, 27 Aug).

Next Meeting: 29 Aug 2025 to review action item completion and confirm A1 draft readiness.

3. Initial Project Scope

3.1 Vision Statement

SmartResearch is an AI-driven platform that reduces the time and effort of literature review by producing concise summaries and clustering academic papers into interpretable topics. The vision: minutes, not days to identify relevant research, reveal connections, and move from screening to synthesis.

Phase 1 will deliver a web prototype able to: ingest ≥10 English, text-selectable PDFs; generate ≤120-word structured summaries (objective, method, findings, limitations); embed and cluster documents using unsupervised methods (k-means/hierarchical) with auto keyphrase labels; provide an interactive React UI for navigation; and export results (CSV/JSON).

3.2 Product Scope

The prototype enables users to upload papers, receive summaries, explore thematic clusters, and export structured outputs. It prioritises speed (≤60s per 10-paper batch), reliability (graceful handling of malformed files), and usability (≤3 clicks to locate related papers).

3.3 In scope (Phase 1)

- Batch ingest of ≥10 PDFs with validation and error reporting
- Summarisation pipeline with ≤120-word, four-field outputs
- Embedding + clustering with adjustable k and labels
- UI: upload, progress, summaries list, cluster view, search/filter
- Lightweight storage (SQLite/filesystem) + export (CSV/JSON)
- Basic evaluation metrics and user guide

3.4 Out of scope (Phase 1)

- Non-English or scanned PDFs
- External database integration or web crawling
- Mobile/native apps; OCR; fine-grained citation/claim extraction
  Advanced access control or multi-tenant accounts

3.5 Minimum viable product (MVP) definition

Upload ≥10 PDFs → system returns ≤120-word summaries + topic clusters → user can browse by cluster and locate any paper in ≤3 clicks → export summaries + cluster labels to CSV/JSON.

3.6 Non-functional targets (Phase 1)

- Performance: ≤60s per batch; ≤5s per paper
- Usability: novice users locate related paper in ≤3 clicks
- Reliability: no crashes on malformed files; retry guidance
- Security/Privacy: files stored locally; no third-party transfer
- Maintainability: pinned dependencies, documented API boundary, CI install check

3.7 Deliverables (Phase 1)

- Running prototype demonstrating upload, summarisation, clustering, navigation, export, and performance targets
- User guide + setup notes
- Prototype evaluation summary (metrics + feedback)
- A1 documentation pack

4. Scope Management

4.1 Purpose

SmartResearch uses scope management to ensure project work is compliant with vision, product scope, and Phase 1 deliverables. To avoid scope creep and guarantee delivery within time and resource limitations, scope management offers a structured method for identifying, validating, and controlling scope changes.

4.2 Scope Baseline

The Phase 1 in-scope items, out-of-scope exclusions, and MVP definition listed in Section 3 serve as the baseline. Any deviation from this baseline will constitute a trigger for scope management.

4.3 Scope Change Process

- **Initiation:** Any team member can raise a scope change request (SCR).
- **Documentation:** SCRs are logged in the GitHub issue tracker under the "Scope/Change" label, including rationale, impact on deliverables, timeline, and resources, and relevant additional information.
- **Review:** Project Manager (Chelsea) reviews each SCR, consulting with the Technical Lead (Nishad), Frontend Lead (Noor), and Infra Lead (Azwad). All SCRs must be reviewed and a decision issued with seven calendar days of flagging.
- **Decision:**
    - *Minor changes* (UI tweaks, text edits, bug fixes) may be approved directly by the Project Manager.
    - *Major changes* (new features, expanded NLP tasks, non-English support) require advisor consultation (Dr. Jack Yang) before approval.
- **Implementation:** Approved changes are assigned and integrated into the backlog. Rejected changes are documented with justification separately.
- **Communication:** All decisions are noted in weekly minutes and reflected in updated GitHub project boards.

4.4 Scope Validation

- **Acceptance Criteria Reference:** Formal acceptance criteria are maintained in the GitHub repo `/docs/acceptance-criteria/` to keep them version-controlled and auditable, and updated as the project progresses.
- **Deliverable Check:** All outputs are validated against the acceptance criteria before submission.
- **QA Role (Chelsea):** Confirms compliance with acceptance criteria and ensures deliverables remain within scope.
- **Formal Sign-off:** Advisor (Jack) confirms acceptance during milestone reviews; outcomes are logged in weekly minutes.

- **Traceability:** Each validated deliverable is linked to the corresponding GitHub repo and mapped against acceptance criteria in the repo.

4.5 Scope Control

- **Standing Agenda:** Weekly check-ins include a "scope review" item to detect and address unapproved work.
- **Risk Register Linkage:** Scope creep is tracked as a recurring risk (see R4 in 9.2).
- **Non-Compliance:** Any unapproved work identified is halted immediately and flagged as a risk item.
- **Exclusions Enforced:** Out-of-scope features (OCR, multi-language support, mobile apps, citation extract) remain locked until Phase 2. Even if technically feasible, they will not be worked on in Phase 1.

4.6 Responsibilities

- **Project Manager / QA / Swing Dev (Chelsea):** Owns the scope management process; documents all SCRs; ensures timely review and compliance with decisions.
- **Infra Lead (Azwad):** Reviews integration and system stability implications and scope changes.
- **Tech Lead (Nishad):** Reviews algorithmic feasibility and timeline impact.
- **Frontend Lead (Noor):** Reviews usability impact and interface implications.
- **Advisor (Dr. Yang):** Provides final approval on major scope changes and resolves escalations if reviews exceed the 7-day window.

4.7 Tools

- **GitHub Project & Issues:** Central system for logging, tracking, and documenting scope changes.
- **Risk Register:** Maintained as a living document (see Section 9, Risk Management) with scope creep tracked as a standing item.
- **Weekly Minutes** (`/docs/weekly-minutes/`): Official record of scope decisions, sign-offs, and change approvals/rejections.
- **GitHub Labels:** "Scope/Change" label used to flag and manage scope change requests.

5. Project Need

5.1 Rationale

SmartResearch is required to bridge the gap between existing literature-management tools and the needs of research teams. Current platforms provide partial solutions. Search engines return lists, visualisation tools show citation networks, and AI digests summarise one paper at a time, but none offer integrated batch-level summarisation with thematic clustering. This limitation extends review timelines, creates inconsistency between reviewers, and makes the discovery of related studies less efficient (Our World in Data, 2024; Borah et al., 2017; Cochrane, 2024).

5.2 Value Proposition

The project delivers value in four areas:

- **Efficiency:** Screening currently consumes ~10–15 minutes per paper, meaning 100 papers require 17–25 hours before analysis begins. SmartResearch reduces this to minutes per batch by generating consistent digests and topic clusters, moving teams faster toward synthesis.
- **Consistency:** Summaries vary by reviewer and session. Fielded outputs (objective, method, findings, limitations) paired with labelled clusters provide standardisation and comparability across projects.
- **Discovery:** Flat lists hide structure. Clusters reveal themes and gaps early, improving inclusion decisions and reducing the chance of missing relevant evidence.
- **Governance and Reuse:** Structured outputs provide a transparent audit trail, easier to review, teach with, and archive for supervisors and future students.

5.3 Beneficiaries and Outcomes

- **Researchers and students:** reduced time-to-insight and more reliable screening artefacts.
- **Supervisors/advisors:** stronger oversight via consistent, inspectable outputs.
- **Institutions:** higher throughput of reviews and stronger evidence bases for projects, assessments, and grants.

5.4 Feasibility and Appropriateness

Feasibility is supported by open-source NLP and clustering methods, enabling a demonstrable prototype within semester constraints and modest computing. The project scope is deliberately bound to English, text-based PDFs to ensure timely delivery and credible evaluation.

5.5 Timeliness

Growth in publication volume makes manual review increasingly untenable. Addressing the need now allows validation within the academic cycle and reduces risks in subsequent coursework phases (Our World in Data, 2024).

5.5 Measures of success (Phase 1)

- **Time reduction:** ≤2 minutes per paper.
- **Interpretability:** ≥75% cluster purity; intelligible labels.
- **Consistency:** summaries remain materially similar across users.
- **Throughput/Reliability:** batch of 10 PDFs processed ≤60s; errors handled gracefully.
- **Reusability:** CSV/JSON exports are sufficient for downstream analysis.

6. Market Analysis

There are many digital tools that can be used to assist in literature review, although none of them offer a complete and integrated end-to-end workflow. Google Scholar and Semantic Scholar are the most popular discovery systems with powerful search and citation tracking abilities. Yet they give simple flat lists of results devoid of any summarisation or clustering. Scholarcy generates digests of single PDFs using AI but is not capable of dealing with complete corpora or coherent, structured outputs. Elicit.org supports question-driven data extraction, like study populations or outcomes, but is limited in scope and does not provide thematic clustering.

Visualisation tools such as ResearchRabbit and Connected Papers allow users to explore citation networks. Although these tools are useful for discovery, they focus on relationships between papers rather than creating thematic groupings or summaries. In the same way, Rayyan and Covidence support systematic reviews by facilitating decisions regarding inclusions and exclusions, yet still rely on manual reading of abstracts and texts.

Large language models (LLMs) like ChatGPT and Copilot offer flexible summarisation, but they are general-purpose systems. They need manual prompting per paper, produce non-standardised outputs, and lack clustering or batch automation. Furthermore, their black-box nature makes them inappropriate for academic settings where a high level of transparency and reproducibility is required.

From this analysis, three persistent gaps in the market emerge:

- **Batch Summarisation**: No tool currently generates summaries for multiple papers simultaneously.
- **Corpus-Level Clustering**: Citation mapping exists, but interpretable clusters with labelled summaries are absent.
- **Integrated Workflow**: Researchers must rely on a patchwork of separate platforms for discovery, note-taking, and synthesis.

6.1 Competitor Comparison

Feature	**Existing Products**	**SmartResearch (Proposed)**
Summarisation	Scholarcy/Elicit, one paper at a time, inconsistent outputs with no standardisation	Batch summaries with consistent four-field output (objective, method, findings, limitations).
Clustering	ResearchRabbit/Connected Papers, citation graphs only.	Unsupervised clustering with auto keyphrase labels for thematic navigation.

Batch Processing	Most tools process papers individually or not at all.	Handles ≥10 PDFs in a single batch, saving hours of manual effort.
Export / Reuse	Limited or basic export options.	CSV/JSON export of summaries and clusters for downstream analysis.
Integration	Researchers must switch between multiple tools.	End-to-end workflow: upload, summarisation, clustering, export.
Transparency	Proprietary or black-box AI tools.	Open-source NLP/ML with transparent evaluation metrics.

6.2 Differentiation of SmartResearch

SmartResearch is specifically designed to address these gaps. Combining batch summarisation, clustering, and structured export makes it less time-consuming, more consistent, and enables discovery beyond existing tools. Its use of open-source NLP and reproducible evaluation can provide academic credibility. Existing systems excel in only one area: search, visualisation, or single-document summarisation. SmartResearch is the first integrated, research-oriented system that combines all three.

7. Time Management

7.1 Objective

The objective of the time management plan is to schedule all project activities so deadlines are met without compromising quality. It provides a framework for allocating resources, sequencing tasks, and monitoring progress against milestones.

7.2 Milestones

Phase	Milestone	Date	Deliverable Output
Phase 1 - Initiation	Initial project description	29 August 2025	Project description document
	Project requirements & interface presentation	7 September 2025	Requirements doc + interface mockup
Phase 1 - Development	Prototype presentation	20 October 2025	Working prototype demo
Phase 2 - Development	Design progress	16 March 2026	Design review report + demo updates
Phase 2 - Delivery	Marketing & technical presentation	25 May 2026	Marketing pitch + technical presentation
	Final product	1 June 2026	Final integrated system + report

7.3 Chosen Approach

Time management will use a Work Breakdown Structure (WBS) to break the project into smaller tasks, each linked to a milestone and deliverable from Section 7.2 This ensures every task has a clear owner, deadline, and expected output.

A Gantt chart will supplement the WBS by mapping these tasks and milestones onto a timeline. This visual schedule highlights dependencies between activities, making it easier to identify bottlenecks early and adjust for delays. Progress will be tracked against both the WBS and Gantt chart to keep the project aligned with the planned phases and deliverables.

8. Quality Management Plan

8.1 Objective

The quality management plan ensures all deliverables meet client expectations and stay within scope. Quality refers to compliance with functional and nonfunctional requirements, usability, consistency of design, and system reliability. This plan defines criteria, roles, and methods for both quality assurance (preventing defects) and quality control (detecting and correcting defects).

8.2 Criteria

- Deliverables must meet all functional and nonfunctional requirements.
- Interfaces must be intuitive, consistent, and accessible.
- The system must perform reliably under varying workloads.
- All modules must be integrated seamlessly without regression.
- Deliverables must be verified against specifications and validated with client expectations.

8.3 Roles and Responsibilities

- **PM & QA / Swing Dev (Chelsea):** Owns the quality management plan; ensures deliverables comply with standards; supports both frontend and backend to close quality gaps where needed.
- **Frontend Lead (Noor):** Ensures UI/UX meets usability, consistency, and accessibility requirements; oversees design implementation.
- **Tech Lead (Nishad):** Ensures backend logic aligns with functional requirements; validates NLP/cluster components.
- **Infra Lead (Azwad):** Overseas integration and deployment; ensures system stability and module interaction.

8.4 Quality Assurance

QA methods establish consistent standards and prevent defects during design and development. Methods include:

- Peer reviews of requirements, code, and design artifacts.
- Monitoring compliance with usability, accessibility, and integration standards.
- Maintaining consistent documentation and coding practices.

8.5 Quality Control

QC methods evaluation deliverables against the defined quality criteria. Methods include:

- System, integration and user acceptance testing
- Verification of functional and nonfunctional compliance.
- Usability reviews across modules and interfaces.
- Logging risks, defects, and resolutions in the risk register.

9. Risk Management

9.1 Scoring method

Risks are assessed using a qualitative 3x3 matrix:

- Likelihood (L): Low=1, Medium=2, High=3
- Impact (I): Low=1, Medium=2, High=3
- Exposure (E): Lx I (1-9)
- Bands: Green 1-3, Amber 4-6, Red 7-9.

This approach provides a simple, transparent way to identify priority risks for ongoing management.

9.2 Risk Register

Risk	**Livelihood**	**Impact**	**Mitigation Strategy**
Summaries lack fidelity	Medium	High	Compare multiple models; hybrid summarisation; spot-checks
Clusters incoherent/hard to label	Medium	High	Grid search *k*; prune noise; keyphrase labelling
Schedule slippage	Medium	Medium	Milestone gates; two-week sprints; cut stretch goals
Scope creep	Low	Medium	Change control; baseline scope; backlog hygiene
PDF parsing failures	Low	Medium	Robust parser; pre-flight validation; fail gracefully
Performance bottlenecks	Low	Medium	Batch jobs; caching; distilled models
UI usability issues	Low	Medium	Quick UX tests; simplify screens; empty-state copy
Acceptance mismatch w/ advisor	Medium	Medium	Fortnightly demos; written acceptance criteria
Data/privacy concerns	Low	Low	Local processing; deletion option; no third-party calls
Toolchain/library conflicts	Low	Medium	Pin versions; lockfile; CI install check

Evaluation ambiguity	Low	Medium	Define rubric; labelled sample; multiple raters
Stakeholder availability	Low	Low	Async workflows; early agenda/docs; escalate via email

9.3 Heat map (snapshot)

- **Red (7–9, high priority):** R1 (summary fidelity), R2 (cluster coherence)
- **Amber (4–6, moderate priority):** R3 (schedule slippage), R4 (scope creep), R6 (performance), R7 (UI usability), R8 (acceptance mismatch), R10 (toolchain), R11 (evaluation)
- **Green (1–3, low priority):** R5 (parsing failures), R9 (data/privacy), R12 (availability)

9.4 Monitoring & governance

The register will be reviewed weekly. Any triggered risks are converted into GitHub issues with clear owners and deadlines. Items are closed when exposure falls to ≤3 or after contingency actions are executed and documented in the repository. This ensures the register remains a living artefact rather than a static appendix.

10. Initial Identification of Skills

10.1 Technical Skills Needed

- **Frontend:** React, Tailwind/shadc
- **Backend:** Python (FastAPI/Flask), REST APIs
- **NLP/ML:** HuggingFace, scikit-learn (K-means, hierarchical)
- **Data Handling:** PyPDF/pdfminer.six, JSON/CSV
- **Storage:** SQLite
- **Tooling/DevOps:** GitHub, Markdown, CI checks

10.2 Non-Technical Skills Needed

- Project management (scheduling, risk register, meetings)
- Documentation (README, user guide, minutes)
- Evaluation (summary rubric, cluster metrics)

10.3 Member Experience and Responsibilities

The team brings a mix of management, frontend, backend, and ML skills, ensuring coverage of all project layers. Each member also has a defined upskilling pathway to balance workload and build capacity beyond their current comfort zone.

Member	Role	Existing Skills	Responsibilities	Skills to Learn
Chelsea	PM & QA / Swing Dev	Project management, QA, GitHub, FE/BE basics	Repo setup, QA checks, documentation, advisor liaison, supports FE/BE where needed	React, Tailwind, FastAPI, HuggingFace, SQLite
Noor	Frontend Lead (UX/UI)	HTML/CSS/JS, UX design, small web apps	Leads UI/UX, builds React components, runs usability testing, presents mockups/demo	React integration, export/filtering, usability testing
Nishad	Tech Lead (NLP/Cluster)	React, Python, MongoDB, Java,	Designs summarisation & clustering pipeline, ensures reproducibility, presents results	HuggingFace, evaluation metrics
Azwad	Infra Lead (Integration)	Python, DBs, CI/CD, backend scripting	Sets up backend (FastAPI/Flask), manages storage (SQLite), integration, CI checks, and ensures performance	Performance tuning, deployment, cloud/enterprise integration

10.4 RACI Matrix

Deliverable/Task	**PM & QA (Chelsea)**	**Tech Lead (Nishad)**	**Frontend (Noor)**	**Infra (Azwad)**
Project scheduling & risk reg	R/A	C	C	C
GitHub repo & documentation	R/A	C	C	C
QA checks & advisor liaison	R/A	C	C	C
Frontend UI	C	C	R/A	I
Backend & integration	C	C	I	R/A
NLP summarisation/clustering	I	R/A	I	C
PDF parsing & data handling	I	C	I	R/A
CI/environment consistency	I	I	I	R/A
A1 deliverables	R/A (compile)	C (inputs)	C (inputs)	C (inputs)
A2 deliverables	R/A (scope/vision)	R (algorithms)	R (UI flows)	R (integration)
A3 prototype demo	R/A (demo flow)	R (results)	R (UI demo)	R (stability)

11. Appendices

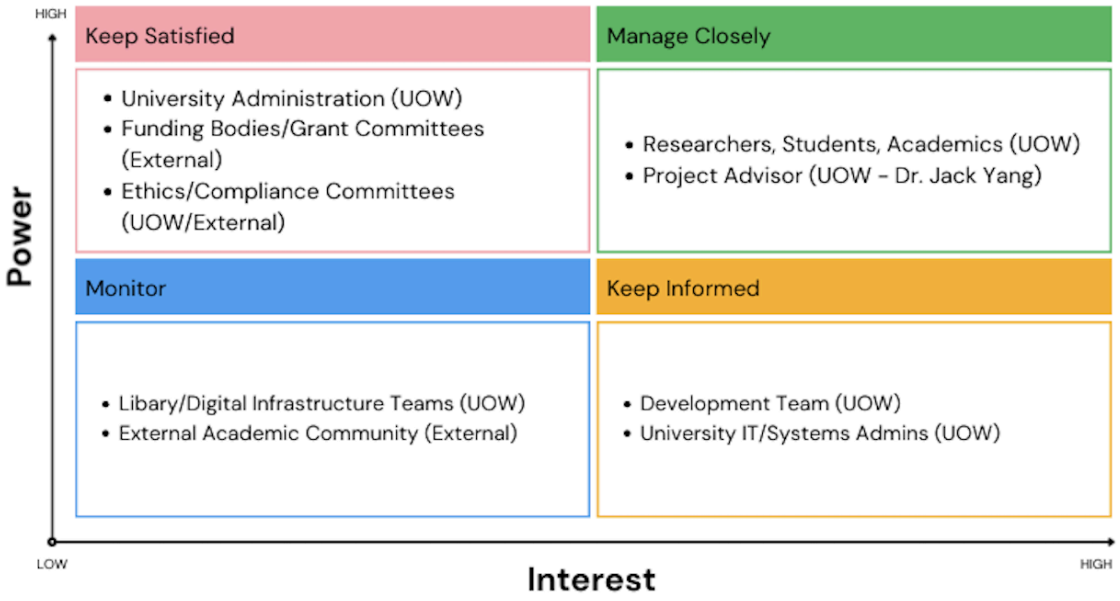
11.1 Stakeholder Analysis

11.1.1 Stakeholder Table

Stakeholder Group	**Affiliation**	**Function in the project**	**Power**	**Interest**
Researchers, Students, Academics	UOW	End users – use the app for summarisation & topic exploration.	High	High
University IT / System Admins	UOW	Provide functional requirements & support integration.	Medium-High	Medium-High
Library / Digital Infrastructure Teams	UOW	Define the technical environment & ensure compatibility.	Medium	Medium
University Administration / Supervisors	UOW	Approve product use, policy compliance, and adoption decisions.	High	Medium
Project Advisor (Dr. Jack Yang)	UOW	Guides technical and academic direction.	Medium-High	High
Development Team (Chelsea, Noor, Nishad, Azwad)	UOW	Build and maintain the product, deliver milestones.	Medium	High
Funding Bodies / Grant Committees	External	Approve resources if pitched beyond coursework.	High	Medium
External Researchers / Wider Academic Community	External	Potential adopters outside UOW	Low	High
Ethics/Compliance Committees	UOW/External	Ensure compliance with copyright, data, and institutional policy	High	Low-Medium

11.1.2 Power-Interest Grid

This grid maps stakeholder groups from 11.1.1 by their relative power and interest.



11.2 Group Project Charter

11.2.1 Purpose and Goals

The SmartResearch team agrees to the following charter to ensure effective cooperation and successful project delivery. The purpose of this charter is to establish the responsibilities, expectations, and communication protocols for all members. Our shared goal is to design and implement a prototype platform that automates academic paper summarisation and clustering. Together, we commit to achieving project milestones, delivering a high-quality prototype, and maintaining professionalism in teamwork and communication.

11.2.2 Team Members and Roles

Project Manager & QA / Swing Developer (Chelsea)
Oversees documentation, repository maintenance, QA, scope control, and advisor liaison.

Technical Lead (NLP/Clustering) (Nishad)
Designs and implements the summarisation and clustering pipeline; ensures reproducibility and performance.

Frontend Lead (UX/UI) (Noor)
Leads user interface design and usability testing; develops React components.

Infrastructure Lead (Integration) (Azwad)
Manages backend integration, storage, and CI/CD; ensures performance and stability.

11.2.3 Communication

The team will use Discord as the primary channel for day-to-day communication. Formal check-ins will be held fortnightly through Zoom, with minutes recorded and stored in the GitHub repository under /docs/weekly-minutes/. All project artefacts, including documents, code, and meeting records, will be maintained in the GitHub repository as the single source of truth.

11.2.4 Decision-Making and Scope Control

- Decisions will be reached by consensus when possible.
- If consensus cannot be achieved, the Project Manager will call for a majority vote.
- Scope changes or major issues are logged in GitHub under the “Scope/Change” label, with advisor consultation required for significant alterations.

11.2.5 Accountability and Deliverables

- Each member is responsible for their assigned sections, code contributions, and deadlines.
- All contributions must be traceable through GitHub commits or documented in shared/submitted files.

- Missed deliverable without prior notice may be reassigned to ensure project progress.

11.2.6 Conflict Resolution

- **Minor disputes:** Addressed in Discord with Project Manager mediation.
- **Escalation:** Unresolved disputes or repeated failures to deliver will be referred to the project advisor (Dr. Jack Yang).

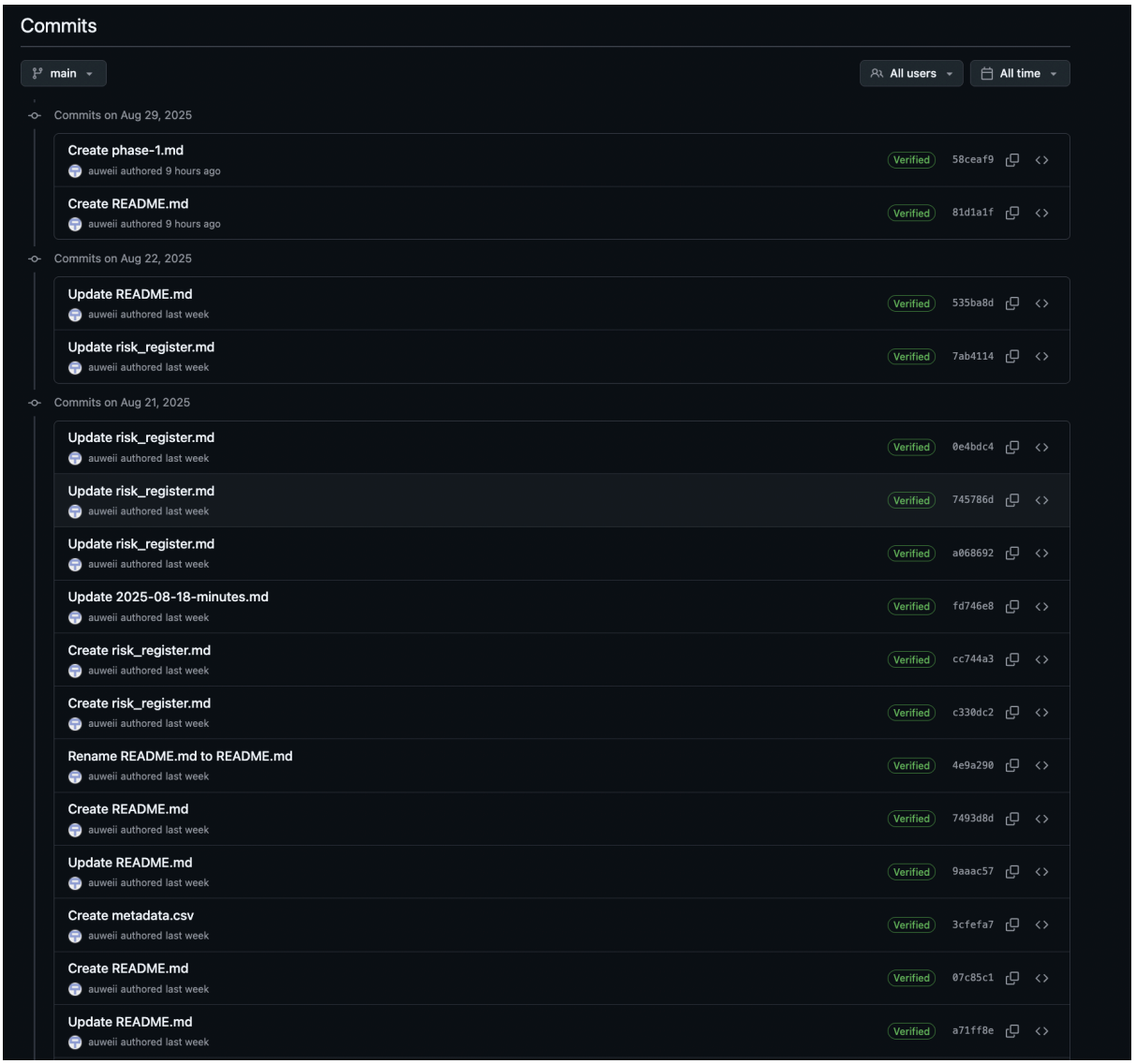
11.3 Group Ethical Approach

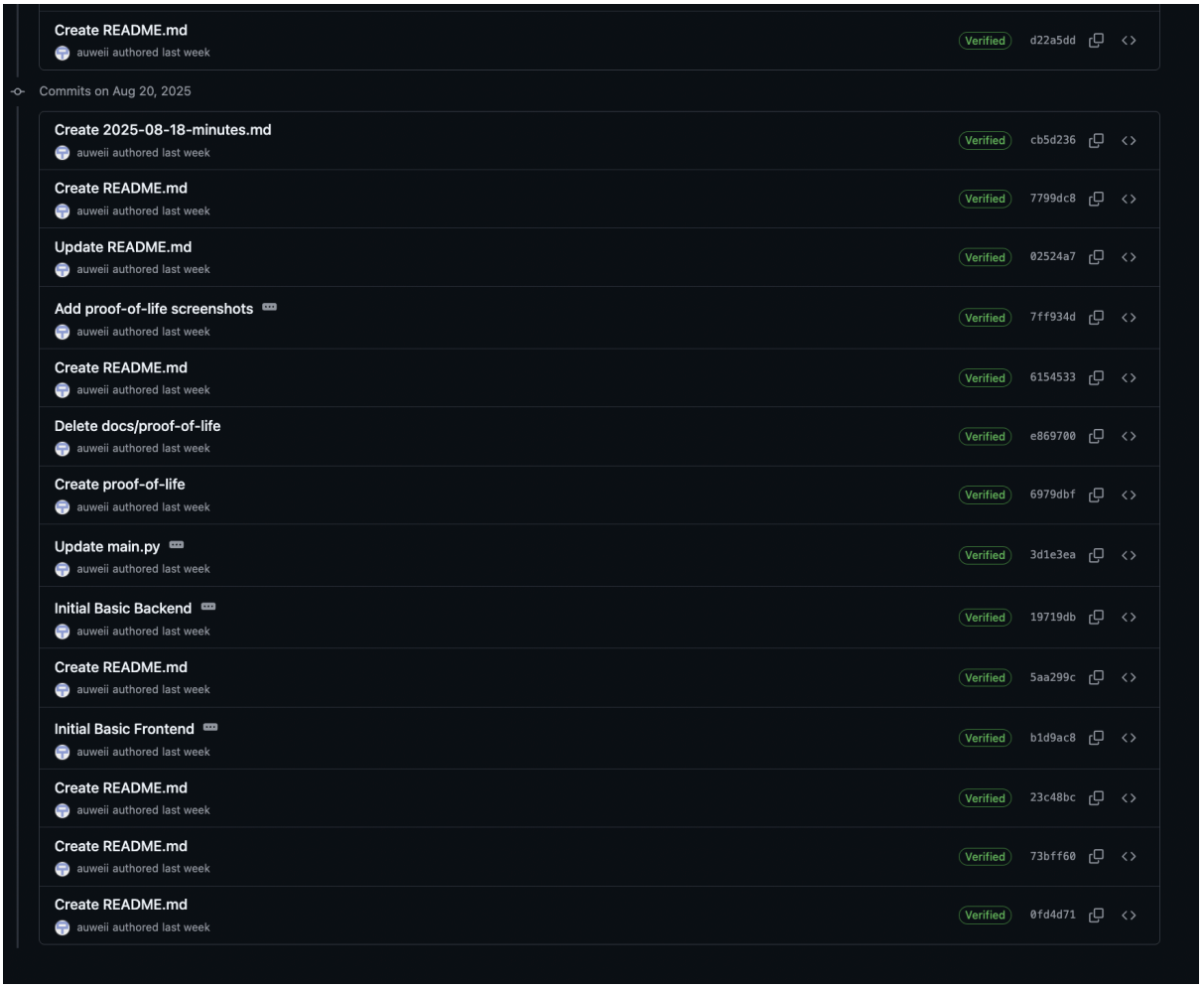
The team's ethical approach is grounded in transparency, fairness and academic integrity:

- **Copyright & IP:** Only open-access scholarly articles will be used to respect copyright and intellectual property.
- **Fair Workload:** Tasks and responsibilities will be divided equitably among members.
- **Academic Integrity:** All sources will be properly referenced, and plagiarism will be strictly avoided.
- **Transparency:** Contributions, ownership, and decisions will be clearly documented.
- **Collaboration:** Team members will promote mutual understanding, inclusion and cooperation.
- **Honesty:** Members will keep each other informed and be truthful about progress and developments.

11.4 GitHub link with commit history

https://github.com/auweii/smartresearch




Create README.md
auweii authored last week
Verified
d22a5dd
Commits on Aug 20, 2025
Create 2025-08-18-minutes.md
auweii authored last week
Verified
cb5d236
Create README.md
auweii authored last week
Verified
7799dc8
Update README.md
auweii authored last week
Verified
02524a7
Add proof-of-life screenshots
auweii authored last week
Verified
7ff934d
Create README.md
auweii authored last week
Verified
6154533
Delete docs/proof-of-life
auweii authored last week
Verified
e869700
Create proof-of-life
auweii authored last week
Verified
6979dbf
Update main.py
auweii authored last week
Verified
3d1e3ea
Initial Basic Backend
auweii authored last week
Verified
19719db
Create README.md
auweii authored last week
Verified
5aa299c
Initial Basic Frontend
auweii authored last week
Verified
b1d9ac8
Create README.md
auweii authored last week
Verified
23c48bc
Create README.md
auweii authored last week
Verified
73bff60
Create README.md
auweii authored last week
Verified
0fd4d71

11.5 Meeting Agendas & Minutes

Meeting Minutes — 18 August 2025
Time: 12:40pm
Location: Zoom
Type: Initial meeting

Attendees
- Advisor: Jack Yang
- Noor Ahmed
- Nishad Gyawali
- Chelsea Okan
- Azwad Ahnaf Zihan

Discussion Summary
- Setup of GitHub repo and share link with Jack.
- GitHub will be the central repository for all code, documents, and project-related materials.
- Homework assigned:
  - Design / brainstorm frontend and backend requirements.
  - Document what features/functions to provide to end users.

Frontend (brainstormed features)
- Upload documents
- Browse all documents
- View generated summaries
- Display topics/keywords of documents
- Display clustering results

Backend (brainstormed features)
- Save documents
- Store metadata (topics, keywords, summaries)
- Manage profiles of stored documents

Data Sources
- Data sources: academic papers available for download.

Action Items
- All team members: brainstorm/design frontend and backend features (document ideas).
- Chelsea: set up a GitHub repo and share the link with Jack.

Next Meeting
- Scheduled for next Friday

References

Booth, A. (2023) *Study selection (ESG workshop slide set).* EPPI-Centre.

Borah, R., Brown, A.W., Capers, P.L. and Kaiser, K.A. (2017) ‘Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry’, *BMJ Open*, 7(2), e012545.
https://doi.org/10.1136/bmjopen-2016-012545

Cochrane (2024) *Cochrane Handbook for Systematic Reviews of Interventions.* Version current at time of access.

Connected Papers (2025) *About Connected Papers.*

Elicit (2025) *Elicit: The AI research assistant.*

Our World in Data (2024) ‘Scientific and technical journal articles’.

ResearchRabbit (2025) *Guide to using ResearchRabbit.*

Scholarcy (2025) *Article summariser overview.*