

# **UOW Online Innovation Gallery - Technical Report**

# Table of Contents

<b>Table of Contents.....</b>	<b>2</b>
<b>Introduction.....</b>	<b>3</b>
<b>Scope.....</b>	<b>3</b>
<b>Iteration Management.....</b>	<b>4</b>
<b>System Requirements.....</b>	<b>5</b>
Functional Requirements.....	5
Non-Functional Requirements.....	6
Technical Requirements.....	6
Constraints and Limitations.....	7
<b>Project Summary.....</b>	<b>7</b>
<b>System Design.....</b>	<b>13</b>
1. Architecture Overview.....	13
2. Technologies & Tools Used.....	13
Frontend Frameworks & Libraries.....	13
AWS Amplify (Backend).....	14
Supporting Libraries.....	14
Development Tools.....	14
3. Key Implemented Features.....	15
Student Project Form:.....	15
Admin Dashboard:.....	15
Media Management:.....	15
Preview & Interaction:.....	15
4. Design Considerations.....	15
<b>Project Closeout.....</b>	<b>16</b>
Lessons Learned.....	16
Post-Project Review.....	17
Project Acceptance.....	17
Transition Plan.....	18
Final Remarks.....	18
<b>Deployment and Infrastructure Documentation.....</b>	<b>19</b>
AWS Amplify (Gen 2 & Backend Infrastructure-as-Code).....	19
Next.js (Frontend Framework).....	19
Other Helpful Tools/Docs.....	19
<b>Feature Transition Table.....</b>	<b>20</b>
<b>Closing.....</b>	<b>21</b>

# Introduction

The purpose of this technical report is to document the development and implementation of the UOW Online Innovation Gallery, an online platform created to showcase student innovation at the University of Wollongong. This report outlines the system's technical design, functional requirements, user workflows, and administrative processes to ensure transparency, usability, and security. It serves as a comprehensive reference for stakeholders, demonstrating how the system meets its intended objectives of increasing visibility for student projects, streamlining investor engagement, and fostering a culture of innovation. Additionally, the report details the installation, maintenance, and future scalability of the system, providing a foundation for ongoing improvement and institutional adoption.

## Scope

The UOW Online Innovation Gallery project was undertaken over the course of two academic semesters, providing a generous timeline for iterative development, exploration, and refinement. The project's scope was intentionally broad, granting our team the flexibility to design and build a solution that met key objectives while encouraging creativity and autonomy. Within the overarching mission to develop a student-led innovation gallery, we were guided by two core constraints: alignment with the iAccelerate and University of Wollongong branding, and the creation of a platform that amplifies student intellectual property.

The aim was to build a dynamic, user-friendly web platform that functions similarly to social media platforms like Instagram, where students could upload their projects, describe their work, and tag areas of interest. These tagged projects would then be discoverable by external businesses and investors who could browse the gallery, find innovations aligned with their interests, and initiate contact through a secure form. The vision was to open new pathways for student recognition, career advancement, and commercialization opportunities, all while maintaining a secure and moderated environment under the stewardship of the iAccelerate team.

# Iteration Management

The development lifecycle of the UOW Online Innovation Gallery followed a modified Agile methodology, aligned closely with Scrum principles to support iterative delivery, stakeholder feedback, and adaptive planning. The project was strategically divided into three major milestones: (1) Front-End Prototyping, (2) Full-Stack Development, and (3) Final Polishing and Handover.

During Semester 1, the focus was on project initiation and system planning, involving rigorous scope definition, requirements elicitation, and UI/UX prototyping. Leveraging wireframing tools and front-end frameworks, the student-facing interface was designed to present a functional vision to stakeholders. This design phase prioritized the information architecture and user experience flow, in alignment with UOW and iAccelerate branding guidelines.

As Semester 2 commenced, the development team shifted toward implementation. Development sprints began, encompassing component-based front-end development, admin dashboard architecture, and the integration of form-based data ingestion pipelines. Weekly sprint cycles emulated Agile rituals; featuring sprint planning, stand-ups, retrospectives, and backlog grooming. Bi-weekly stakeholder engagement sessions during Semester 1 transitioned into weekly client interactions in Semester 2, enabling faster feedback loops, iterative refinements, and continuous alignment with stakeholder expectations.

The project execution followed a time-boxed iteration model, with work packages scoped per sprint. Technical debt was tracked and minimized through Git-based version control, while tasks and deliverables were managed via internal task boards. Emphasis was placed on feature prioritization, balancing MVP delivery with stretch goals, and maintaining project velocity without compromising system integrity.

By May 24th, the core platform met functional requirements, marking the completion of the development milestone. The final phase involved QA testing, UI polishing, and handover documentation. Deliverables included a comprehensive user manual, a technical report, and a scope deviation log detailing proposed but

out-of-scope features. This structured approach ensured a smooth transition to iAccelerate for operational ownership and future scaling.

## **System Requirements**

The UOW Online Innovation Gallery was conceptualized and developed to meet the functional, non-functional, and technical constraints outlined in the SCIT CSIT321 project brief. The system requirements were defined through continuous stakeholder engagement, weekly iteration meetings, and alignment with the broader strategic goals of iAccelerate and the University of Wollongong.

### **Functional Requirements**

User Roles and Access Control:

The system must support two primary user types: Students (content submitters) and Admins (iAccelerate staff overseeing moderation and approvals). Students must be able to sign up using UOW credentials, submit projects, and manage content visibility. Admins must access an internal dashboard to review, approve, or reject submissions and manage project metadata.

Project Submission and Approval Workflow:

Students must be able to upload projects (text, tags, images, embedded videos), preview their gallery pages, and submit them for review. Admins must validate submissions through an approval interface that tracks project status and ensures legitimacy and alignment with UOW branding.

Search and Discovery Mechanism:

The gallery must allow external users (e.g., industry partners, investors) to browse published projects using filters (tags, categories, disciplines). The search system must be dynamic and optimized for ease of use.

Contact and Engagement Pipeline:

Interested external parties must be able to submit a short contact form linked to a specific student project. This form is routed to iAccelerate's administrative backend for validation, ensuring controlled and secure communication with students.

## **Non-Functional Requirements**

### Security and Data Protection:

All user data and content must be stored and processed securely using encrypted AWS services. Role-based access control (RBAC) must be enforced to prevent unauthorized access to student data or backend administrative tools.

### Scalability and Performance:

The platform must be hosted using scalable cloud infrastructure (AWS Amplify, Lambda) to ensure high availability and load handling during exhibitions, public showcases, or simultaneous browsing by external stakeholders.

### Accessibility Compliance:

The interface must conform to WCAG accessibility standards, ensuring the platform is usable by all students, including those with disabilities.

### Maintainability:

Source control using GitHub ensures versioning and team collaboration. The modular React architecture enables future maintainability and ease of feature extension.

## **Technical Requirements**

### Frontend Framework:

The platform is developed using React.js, enabling a responsive and modular component-based UI.

### Backend Infrastructure:

A serverless architecture powered by AWS Lambda and API Gateway handles dynamic backend logic. Project metadata and form submissions are processed securely.

### Data Storage:

AWS S3 and EC3 are used to store student-uploaded files, high-resolution media, and static content. Data is served using Amazon CloudFront for performance optimization.

#### Hosting and Deployment:

The gallery is deployed using AWS Amplify Hosting with integrated CI/CD pipelines for rapid deployment and testing.

#### Administrative Interface:

A custom-built admin dashboard supports content moderation, data validation, and communication review workflows. The dashboard is secured using admin authentication via AWS Cognito.

#### Tooling and Collaboration Environments:

- Version Control & Deployment: GitHub
- Design and Prototyping: Figma, Draw.io
- Communication: Discord, MS Outlook
- Documentation and Reporting: Microsoft Office, Google Docs

## Constraints and Limitations

- Final approval of all published content is to be granted by the iAccelerate Project Leader.
- Hosting and infrastructure are managed and funded by iAccelerate.
- The CMS and backend systems are optimized for AWS Cloud services and may require
- UOW IT involvement for integration with internal university systems.

## Project Summary

The UOW Online Innovation Gallery has successfully met the primary objectives and functional requirements as outlined in the original project brief submitted to iAccelerate and the School of Computing and Information Technology. The project delivers a fully functional digital showcase platform designed to allow students to upload and present their intellectual property (IP) and academic project work. These submissions can be viewed by industry partners, venture capitalists, university departments, and other external stakeholders with the goal of promoting collaboration, investment, and professional opportunities.

## UOW Online Innovation Gallery - Technical Manual

The platform is built using Next.js for the front end, providing modern web development features such as server-side rendering (SSR), file-based routing, and static site generation (SSG), which enhance both performance and SEO. The user interface is styled using Tailwind CSS for utility-first custom design and Ant Design for robust, production-ready UI components. These frameworks were selected to ensure the application is both aesthetically modern and functionally responsive across all devices.

On the backend, the platform leverages AWS Amplify, which provides a serverless cloud infrastructure for hosting, API handling, authentication, and storage. Amplify was instrumental in allowing the team to define backend infrastructure using infrastructure-as-code (IaC) principles. This enabled seamless integration between the front-end application and backend resources such as GraphQL APIs, Lambda functions, and S3 storage buckets. Deployment pipelines were configured to build and release the application directly from a Git repository, ensuring CI/CD best practices.

Minimal scope deviations occurred during the project lifecycle. Some advanced features such as real-time chat functionality and project engagement analytics were deprioritized due to time constraints and were documented in the Scope Deviation Log. These features were acknowledged and approved by stakeholders for potential inclusion in future development cycles to maintain delivery of a stable Minimum Viable Product (MVP) by the project deadline.

A Requirements Traceability Matrix (RTM) has been provided in the appendix to demonstrate how each original requirement was addressed and validated. As reflected in the RTM, all high-priority and critical functional requirements have been implemented successfully. Several minor usability and design enhancements were also incorporated throughout development in response to stakeholder feedback gathered during iterative sprint reviews.

At the time of final submission, the UOW Innovation Gallery is in a production-ready state and has been transitioned to iAccelerate for future maintenance, scalability, and institutional adoption. Comprehensive documentation—including this technical report and a user manual—has been delivered to support future administrative and development efforts.



<b>Requirement ID</b>	<b>Requirement Description</b>	<b>Priority</b>	<b>Implementation Status</b>	<b>Notes / Justification</b>
FR-01	Allow students to submit and upload their project information and IP	High	Implemented	Built using a form-based submission pipeline with Amplify backend and integrated S3 storage.
FR-02	Display student projects in an accessible, browsable online gallery	High	Implemented	Dynamic project rendering via Next.js with SSR and responsive layout using Tailwind CSS + Ant Design.
FR-03	Enable admin moderation and approval of project submissions	High	Implemented	Admin dashboard built to review, edit, and approve/reject content using Amplify APIs.
FR-04	Ensure branding alignment with UOW and iAccelerate	High	Implemented	UI designed using Figma; branding elements implemented

				with Tailwind and Ant Design components.
FR-05	Support authentication and user management	High	Implemented	Authentication handled via AWS Amplify Auth module with user role segregation.
FR-06	Provide a stable and scalable hosting solution	High	Implemented	Hosted using AWS Amplify with CI/CD deployment from GitHub.
FR-07	Ensure accessibility and mobile responsiveness	Medium	Implemented	Fully responsive UI and accessibility standards followed through Tailwind + Ant Design styling.
FR-08	Provide admin dashboard to manage and curate project content	High	Implemented	Built with component-based architecture; allows for real-time content management.

## UOW Online Innovation Gallery - Technical Manual

FR-09	Store rich media (images, videos) associated with student projects	Medium	Implemented	Media files stored securely on AWS S3 with URL-based access and CMS integration.
FR-10	Allow external users (VCs, industry) to view published projects	High	Implemented	Publicly accessible frontend without login; no access to backend features.
FR-11	Enable version-controlled deployment and maintainable codebase	Medium	Implemented	GitHub repository integrated with Amplify for automated CI/CD pipelines.
NFR-01	System should be scalable and cloud-based	High	Implemented	Uses serverless AWS infrastructure (Lambda, S3, Cognito, API Gateway).
NFR-02	System should be secure and reliable	High	Implemented	Authentication, authorization, and secure

# UOW Online Innovation Gallery - Technical Manual

				storage in AWS; HTTPS enforced.
NFR-03	Documentation (Technical + User Manual) must be delivered	High	Implemented	Delivered with final submission; includes user guide and technical report.
NFR-04	Must support deployment via Infrastructure as Code (IaC)	Medium	Implemented	AWS Amplify used with infrastructure definitions in resource files.
DE-01	Real-time messaging between students and VCs	Low	Deferred	Documented in Scope Deviation Log; not implemented due to time/complexity.
DE-02	Advanced analytics on project views and interactions	Low	Deferred	Documented in Scope Deviation Log; marked for future enhancement.

# System Design

The UOW Online Innovation Gallery was designed and developed using a modern web application stack focused on performance, scalability, and maintainability.

This section outlines the system's core architecture, technologies used, and feature-level implementation decisions that support the platform's functionality and user experience.

## 1. Architecture Overview

The system follows a modular full-stack web architecture consisting of:

- A React-based frontend using Next.js for routing, rendering, and page structure.
- A code-defined backend using AWS Amplify Gen 2, which handles authentication, data storage, and serverless logic.
- A CI/CD pipeline linked to GitHub for deployment automation.
- Serverless hosting and resource provisioning via AWS services through Amplify's infrastructure-as-code (IaC) capabilities.

## 2. Technologies & Tools Used

Languages:

- TypeScript – Used throughout the application for strong typing and better IDE support.
- JavaScript – Applied via third-party libraries and for client-side runtime behavior.
- HTML/CSS – Written using JSX, with styling managed through Tailwind CSS classes.
- JSON – Employed in backend schema definitions, API payloads, and local configuration files.

### Frontend Frameworks & Libraries

- Next.js – App Router structure, supporting server-side rendering (SSR), dynamic routing, and improved performance.
- React – Core frontend library, leveraging hooks such as `useState`, `useEffect`, and `useMemo`.

- Tailwind CSS – Enables utility-first styling for fast and consistent design workflows.
- Ant Design (AntD) – Robust component library used for form controls, buttons, inputs, modals, and layouts.
- React Quill – Rich text editor integrated for content-heavy fields like "Background" and "Project Description."

## **AWS Amplify (Backend)**

- Amplify Gen 2 – Backend defined in code and deployed via GitHub integration.
- Amplify Auth – Provides secure authentication, session management, and user registration flows.
- Amplify Storage (S3) – Handles secure file uploads, including previewing and deletion of images.
- Amplify Data (GraphQL API) – Data models used for managing projects, users, and submission workflows.

## **Supporting Libraries**

- lodash.debounce – Prevents excessive API calls during collaborator email search.
- dayjs – Lightweight date library for formatting and bug-fixing date picker behavior.
- antd-img-crop – Provides image cropping functionality during file uploads.
- React Dynamic Import – Dynamically loads components like ReactQuill to avoid SSR-related issues.
- PapaParse – Parses CSV data for structured import of project information.

## **Development Tools**

- VS Code – Main development environment.
- Git & GitHub – Version control, feature branching, and Amplify CI/CD integration.
- Figma – Used during UI/UX design for prototyping layouts and mapping user flow.

- npm & Node.js – Package and environment management for project dependencies.

### **3. Key Implemented Features**

The following functional and non-functional components were designed and implemented to meet core requirements and enhance usability:

#### **Student Project Form:**

- Rich text fields for project description and background via React Quill.
- Form validation, draft saving, and resume-edit features.
- Debounced search field for adding collaborators via email.
- Auto-enabled edit mode when students revisit an existing draft or submission.

#### **Admin Dashboard:**

- Project status management (e.g., Pending, Approved, Rejected).
- Status resets to "PendingEdit" when students update an approved project.
- Role-based control – Admin-specific actions (approve, reject) hidden from student UI.

#### **Media Management:**

- Image uploads using Amplify Storage (S3) with preview thumbnails (Ant Design's picture-card layout).
- Image cropping prior to submission.
- Secure deletion from storage on update or rejection.

#### **Preview & Interaction:**

- Modal-based project preview replicating the final public layout.
- Optimized data fetching to reduce unnecessary re-renders and improve form responsiveness.

### **4. Design Considerations**

- Scalability: The system's cloud-native architecture enables easy scaling through AWS Amplify's built-in support for serverless resources.

- **Maintainability:** Infrastructure as code (IaC) allows for version-controlled backend resources and repeatable deployments.
- **Security:** Amplify Auth provides robust authentication and fine-grained access control for different user roles.
- **Performance:** Next.js SSR and static generation contribute to faster page loads and improved SEO.

## Project Closeout

The completion of the UOW Online Innovation Gallery marks a significant milestone for our team, iAccelerate, and the School of Computing and Information Technology. This section outlines the key reflections from the project lifecycle, project acceptance, knowledge transfer activities, and the transition plan for future development and maintenance.

## Lessons Learned

Throughout the development of the Innovation Gallery, the team has gained valuable insights into both technical execution and collaborative project delivery. Key lessons include:

- **Consistent Communication is Critical:** Maintaining a regular communication cadence—especially through weekly meetings with our project supervisor—proved to be one of the most effective elements in ensuring project alignment. These meetings allowed for iterative feedback, clearer direction, and early identification of any blockers.
- **Stakeholder Involvement Drives Progress:** Actively involving stakeholders and presenting tangible progress boosted both team morale and stakeholder confidence. It also fostered a shared sense of ownership and accountability.
- **Adaptability and Change Management:** In the evolving landscape of software development, it became clear that staying current with modern tools and practices (e.g., adopting AWS Amplify Gen 2 and Next.js) helped us build a resilient and scalable solution. Our ability to adapt quickly to change requests and prioritize stability while still accommodating evolving requirements played a major role in the project's success.



## Post-Project Review

At the time of final submission, the team is proud of the system that has been developed. The UOW Innovation Gallery fulfills its original purpose as a centralized, branded showcase platform for UOW student innovation, supporting industry engagement and long-term digital visibility of capstone and research work.

Key reflections from the post-project review include:

- **Team Growth:** All team members demonstrated strong development, both technically and professionally, through real-world collaboration with an institutional partner.
- **Technology Adoption:** The integration of Next.js, AWS Amplify, and modern UI libraries such as Tailwind CSS and Ant Design helped ensure a high-quality Minimum Viable Product (MVP) with maintainable code and scalable architecture.
- **Project Management Awareness:** We gained a deeper understanding of agile methodology, scope control, stakeholder prioritization, and how proactive change management directly contributes to project success.

## Project Acceptance

We have ensured that Mr. John Kerr, iAccelerate Program and Community Manager and our primary project stakeholder, is satisfied with the final product. The application meets all critical and high-priority functional requirements outlined in the original brief. As part of formal acceptance:

- A final handover meeting and demonstration has been scheduled for Tuesday, 3rd of June, where we will walk through the completed platform, provide documentation, and address any final queries.
- All administrative and technical documentation has been compiled, reviewed, and submitted, including:
  - This Technical Report
  - The User Manual
  - Source Code and Deployment Instructions
  - Requirements Traceability Matrix (RTM)
  - Scope Deviation Log

## Transition Plan

To ensure a smooth transition of ownership and future scalability of the platform, the following activities and resources are included as part of the project handover:

- Feature Transition Table: A documented list of planned features that were deferred or marked out-of-scope during MVP development (e.g., real-time messaging, advanced analytics). This enables the iAccelerate development team or future student groups to pick up where we left off and extend the platform.
- Deployment and Infrastructure Documentation: The project leverages AWS Amplify's infrastructure-as-code (IaC) capabilities, allowing future developers to replicate, update, or scale backend services easily using the same GitHub repository.
  - Further supporting documents that explain how to **Deploy the Application on AWS**, and how to **Create an Admin User in AWS Amplify** are bundled with this report under "DeployingapponAWS.pdf" and "CreateanAdminUserinAWSAmplifyConsol.pdf".

## Final Remarks

This project has provided a valuable opportunity for the team to apply software engineering and project management concepts in a real-world environment. The successful delivery of the UOW Innovation Gallery reflects not only technical achievement but also strong collaboration between students, faculty, and institutional stakeholders. We are confident that the platform will continue to evolve and serve as a flagship example of innovation at the University of Wollongong.

We look forward to seeing how iAccelerate and future students continue to grow and enhance the gallery, and we remain grateful for the mentorship, feedback, and support received throughout the project.

# Deployment and Infrastructure Documentation

## AWS Amplify (Gen 2 & Backend Infrastructure-as-Code)

1. Amplify Gen 2 Overview  
<https://docs.amplify.aws/gen2/>
2. Getting Started with Amplify Gen 2 (Infrastructure-as-Code)  
<https://docs.amplify.aws/gen2/start/>
3. Define Backend Resources (GraphQL, Auth, Storage, etc.)  
<https://docs.amplify.aws/gen2/define-backend/>
4. Amplify Hosting (CI/CD via GitHub)  
<https://docs.amplify.aws/hosting/github/>
5. Deploy Fullstack Next.js App with Amplify  
<https://docs.amplify.aws/gen2/fullstack-apps/nextjs/>
6. Amplify Auth (User Authentication)  
<https://docs.amplify.aws/javascript/build-a-backend/auth/overview/>
7. Amplify Storage (S3 File Uploads)  
<https://docs.amplify.aws/javascript/build-a-backend/storage/overview/>
8. Amplify Data (GraphQL API / Models)  
<https://docs.amplify.aws/javascript/build-a-backend/data/overview/>

## Next.js (Frontend Framework)

1. Next.js Documentation (Official)  
<https://nextjs.org/docs>
2. Next.js App Router (New Architecture)  
<https://nextjs.org/docs/app>
3. Next.js Deployment with Amplify  
<https://docs.amplify.aws/gen2/fullstack-apps/nextjs/>

## Other Helpful Tools/Docs

- Tailwind CSS Documentation  
<https://tailwindcss.com/docs>
- Ant Design Documentation  
<https://ant.design/components/overview/>
- React Quill (Rich Text Editor)  
<https://github.com/zenoamaro/react-quill>

- AWS Amplify GitHub Repo Examples  
<https://github.com/aws-amplify/amplify-example-fullstack>

## Feature Transition Table

Feature	Workload	Proposed Solution
Items per table when browsing data on admin end	Medium	Add pagination and dropdown control for "items per page" using Ant Design Table component props ( <code>pagination.pageSizeOptions</code> ) and Redux or Context state management.
Statistics on signups for the admins to view	Medium	Integrate Amplify Data queries to fetch aggregated user signup data and display them in an admin dashboard using charts (AntD/Chart.js).
Sign up data (when they signed up, how old is the account, etc.) on admin side	Medium	Add <code>createdAt</code> and <code>updatedAt</code> fields to the user model (if not already present) and show in user details modal or table view.
Project upload date and more project statistics on admin side	Medium	Add <code>createdAt</code> to project records and implement views showing total uploads, status breakdown, and timeline charts.
Compulsory field to add reason for rejection for businesses on admin side	Low	Modify reject modal to include a required textarea using Ant Design form validation rules ( <code>required: true</code> ). Store the reason in the project model or activity log.
Email notification system of project updates for user side	High	Use AWS SES or third-party service (e.g., SendGrid) integrated with Lambda functions to trigger email notifications on status changes.
Data disclaimers and local law page (data integrity, privacy)	Low	Add a static page ( <code>/disclaimer</code> ) using Next.js routing. Content can be written in markdown or rich text, referencing UOW's privacy and integrity policies.
Change "Sign in / Sign up" to "Student Sign in / Sign up"	Very Low	Update button labels and route titles in the header and auth pages.
Rename "Connect with the Students" to	Very Low	Rename all references and component labels accordingly in

"Business Enquiry Form"		both UI and backend labels if stored.
Admins can put tags on businesses to detail reasons for connecting (e.g., buying IP, hiring)	Medium	Add a new multi-select tag field to the business profile model. Populate via admin dashboard with predefined tags.
Ways to avoid duplicate projects	Medium	Implement a project title uniqueness check during submission or compare project summaries using string similarity libraries ( <a href="#">string-similarity</a> , <a href="#">fuse.js</a> ) and flag possible duplicates.
Rename "Move Project" to "Evaluate Client" on Admin end	Very Low	Update button/component labels and any backend references used in status transition logic.
Clients need a verification tag to move further down the pipeline on admin side	Medium	Add a Boolean <a href="#">isVerified</a> or tag field to client model. Restrict pipeline actions until verified.
Rename "In Discussion" to "Approved/Verified Client" on admin end	Very Low	Update label references in status dropdowns and displayed UI text.
Image on the Projects page not displaying correctly on mobile	Medium	Review Tailwind responsiveness classes; update layout for mobile-first compatibility

## Closing

The UOW Online Innovation Gallery project marks a significant milestone in the University of Wollongong's commitment to innovation, student empowerment, and industry collaboration. Developed through the combined efforts of the student development team, iAccelerate, and the School of Computing and Information Technology, the system delivers a scalable, modern, and secure platform for showcasing student-led innovation to external stakeholders, including venture capitalists, business partners, and university departments.

Throughout this project, the team has gained invaluable experience in full-stack web development, project management, stakeholder engagement, and agile software delivery. Leveraging industry-standard tools such as Next.js, Tailwind CSS, Ant Design, and AWS Amplify, we have ensured the platform is both technically sound and user-friendly.

From the initial concept to a fully functional deployment, this project has demonstrated the importance of iterative development, consistent stakeholder communication, and adaptive planning. Key features such as secure user

## UOW Online Innovation Gallery - Technical Manual

submissions, project moderation, role-based access, and responsive UI components were delivered as per the original brief, with additional refinements made in response to feedback during the sprint cycles.

With the final handover to iAccelerate, we are confident the platform will serve as a robust foundation for future enhancements. The user manual, technical documentation, and transition materials (including the Feature Transition Table) have been prepared to ensure seamless continuation of the project beyond our involvement.

We are grateful for the opportunity to contribute to the University's innovation ecosystem and look forward to seeing the UOW Online Innovation Gallery grow in reach and impact in the years to come.