UNIVERSITI TEKNOLOGI MARA SELANGOR

PUNCAK PERDANA CAMPUS

FACULTY OF INFORMATION SCIENCE

CDIM262 - BACHELOR OF INFORMATION SCIENCE (HONS)

INFORMATION SYSTEMS MANAGEMENT

ADVANCED WEB DESIGN DEVELOPMENT & CONTENT MANAGEMENT (IMS566)

GROUP ASSIGNMENT

ACADEMIC TEACHING AND LETTER ADMINISTRATION SYSTEM (ATLAS)

PREPARED BY:

MUHAMMAD AMIERUL HAQEEM BIN MOHD FAIZAN	2025302029
MOHAMMAD NURTHASNIF ZAFFRAN BIN ABDUL TALIP	2025367558
IZYAN FARHANA BINTI AHMAD MARTINO	2025189901
MOHAMMAD AKMAL FAIZUDIN BIN SHASHI	2025147747
RUSYDAN BIN JAMAN	2025180301

PREPARED FOR:

DR. MUHAMMAD ASYRAF BIN WAHI ANUAR

DATE OF SUBMISSION:

2nd FEBRUARY 2026

ACKNOWLEDGMENT

All praise and gratitude be to Allah S.W.T. With His boundless grace and mercy, our group has successfully completed and fulfilled the assignment entrusted to us by our lecturer, Dr. Muhammad Asyraf bin Wahi Anuar, for the subject IMS566 Advanced Web Design Development and Content Management.

Alhamdulillah, we have learned many valuable lessons throughout the process of completing this assignment. Both the explicit and implicit contents of the task have deepened our understanding of general knowledge. We have thoroughly studied all the essential information needed to complete the assignment, and as a result, we have gained countless benefits. We also became more efficient in ensuring the quality of our work so that it could be presented in the best possible form to our respected lecturer.

In conclusion, our group would like to express our deepest appreciation and heartfelt thanks to our lecturer, Dr. Muhammad Asyraf, for all the guidance and support given throughout this assignment. We also extend our gratitude to all parties, whether directly or indirectly involved, who contributed to our efforts. May our work serve as an intrinsic motivation for everyone to become individuals of excellence, adorned with noble character and humanity within.

Muhammad Amierul Haqeem bin Mohd Faizan
Group Representative | ATLAS Project Manager

TABLE OF CONTENT

1.0 INTRODUCTION

The Academic Teaching and Letter Administration System (ATLAS) is an online academic administration system that was created by a team of students at Universiti Teknologi MARA (UiTM) to help administer the university at the faculty level and modernize the processes involved. The system is centered on the computerized handling of official academic letters, especially teaching instructions letters that are given to lecturers in certain courses.

The ATLAS will help the administrative employees and faculty offices, namely in the Faculty of Information Science with the provision of a centralized platform in generating, managing, and approving academic letters electronically. This system allows the administrators to effectively be able to issue teaching instruction letters without using physical documentation which saves them time spent on manual work and enhances efficiency in their operation.

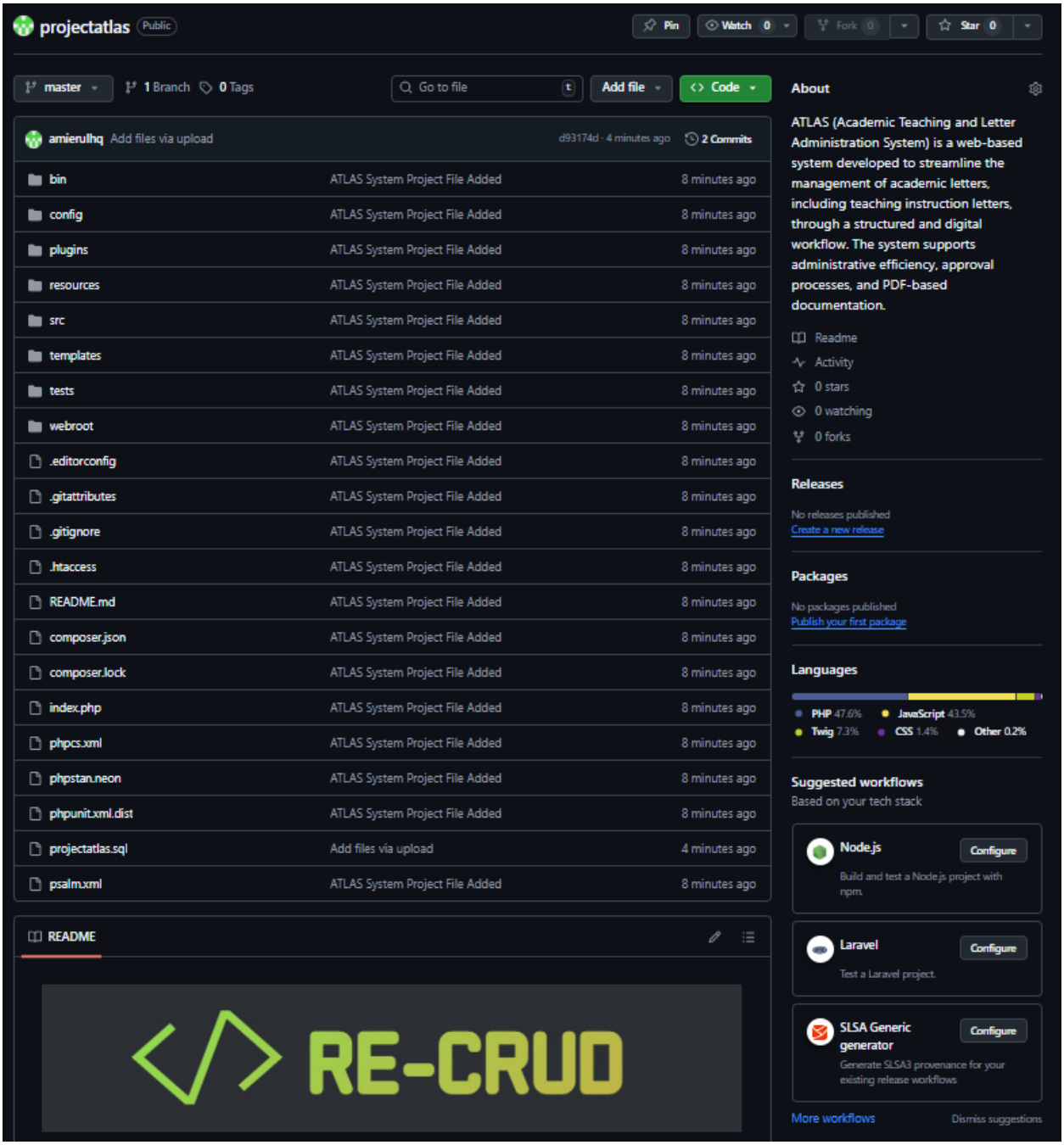
The approved workflow of ATLAS is a major aspect in which the Faculty Dean is the official approval authority of assigned letters. This is to make sure that the letters issued and not just those issued by the faculty are up to standard in terms of consistency and professionalism. On approval, the lecturers can access and download the letters in PDF format as a personal record and refer to them in future.

ATLAS can facilitate the digitization of academic administration by removing the necessity of using real letters and switching to a completely digital form of the process. The system fosters efficiency, accessibility, and sustainability besides increasing the overall management of teaching instructions and formal academic correspondence in the faculty context.

2.0 GITHUB REPOSITORY LINK

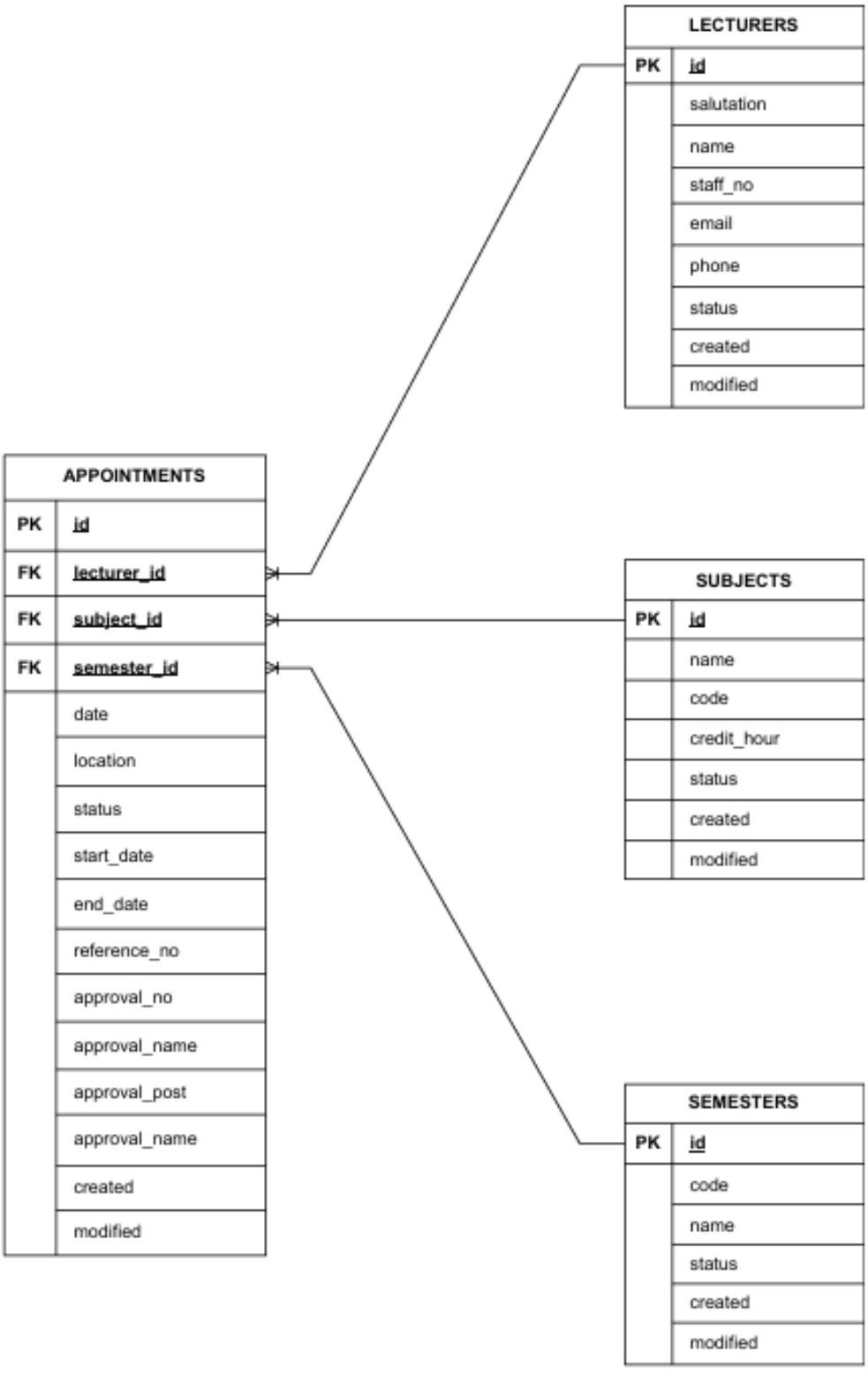
Github Repository Link

https://github.com/amierulhq/projectatlas.git



Overview of Project Atlas GitHub Repository

3.0 ENTITY-RELATIONSHIP DIAGRAM (ERD)



4.0 SYSTEM REQUIREMENTS

This section outlines the software and technical requirements needed to develop and run the Academic Teaching and Letter Administration System (ATLAS). The system is implemented as a web-based application to ensure ease of access for administrative staff and lecturers.

SOFTWARE REQUIREMENTS

1. **Local Server Environment**:
   Laragon
2. **Web Server**:
   Apache (bundled with Laragon)
3. **Programming Languages**:
   PHP, HTML, CSS, and JavaScript
4. **Framework**:
   CakePHP (using the Bake tool for scaffolding)
5. **Framework Support**:
   RECRUD Framework (provided by the course lecturer)
6. **Database Management System**:
   MySQL
7. **Database Administration Tool**
   phpMyAdmin
8. **Front-End Framework**:
   Bootstrap
9. **Icon Library**:
   Font Awesome
10. **Code Editor**:
    Visual Studio Code

VERSION COMPATIBILITY

1. **PHP Version**:
   PHP 8.1
2. **CakePHP Version**:
   Compatible with the installed PHP version
3. **MySQL Version**:
   MySQL 5.7
4. **Browser Compatibility**:
   Google Chrome and other modern browsers with JavaScript enabled

DEPENDENCIES AND TOOLS

1. Laragon for local development and testing
2. CakePHP Bake for rapid CRUD generation
3. RECRUD framework to support structured CRUD operations
4. Bootstrap and Font Awesome for responsive and user-friendly interface design
5. Basic JavaScript for client-side interactivity

These system requirements ensure that ATLAS can be developed and executed efficiently in a local environment. The selected tools and frameworks are suitable for academic projects and support the implementation of a structured, maintainable, and user-friendly system.

5.0 USER INTERFACE OVERVIEW

Academic Teaching and Letter Administration System (ATLAS) allows administrative functions to be performed efficiently with the user interface that is user-friendly, intuitive, and easy to use. The system incorporates a clean format and a uniform navigational pattern to make sure that users can acquire major functions with the least input.

In ATLAS, the web-based interface that was built on the Bootstrap tool offers a responsive design capable of being viewed on a variety of screen sizes. Font Awesome icons are incorporated to make the visual elements more understandable and user-friendly. Simple JavaScript is used to enhance interactivity like form validation and notification.

Its primary interface is a navigation menu which enables users to use basic modules of the system, such as Dashboard, Semester Management, Subject Management, Lecturer Management, and Letter Management. The dashboard will give an overview of the key system notifications and tasks to be completed which will enable administrators to act instantly when necessary.

The modules of each management have a CRUD (Create, Read, Update, Delete) structure that is created with CakePHP Bake and the RECRUD system. The forms are made with legible labels and helper information to direct users as they key in the data and chances of making errors are minimized. Records are organized in tables and offered options of viewing, editing or deleting information through action buttons.

The most important module of ATLAS is the Letter Management interface. It enables its users to come up with official academic letters, by choosing the pertinent semester, subject and recipient. Templates of the letters are automatically loaded, and the approval section is set to the Faculty Dean. Once the approval has been made, the users are allowed to create and save letters in PDF format as documents and references.

Overall, the ATLAS user interface is highly usability-oriented, consistent, and clear so that administrative personnel and lecturers could efficiently carry out their duties on the background of digital academic administration.

6.0 FEATURES AND FUNCTIONALITY

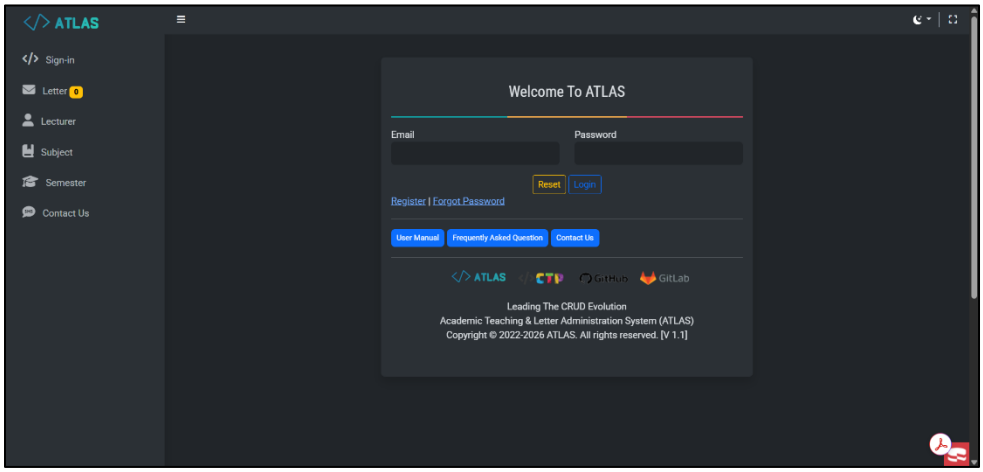


Figure 1: ATLAS Login Interface

SYSTEM LOGIN

The Login portal provides secure entry via email and password authentication, ensuring data protection for administrative records. It includes integrated self-service links for account registration, password recovery, and direct access to system documentation and FAQs.

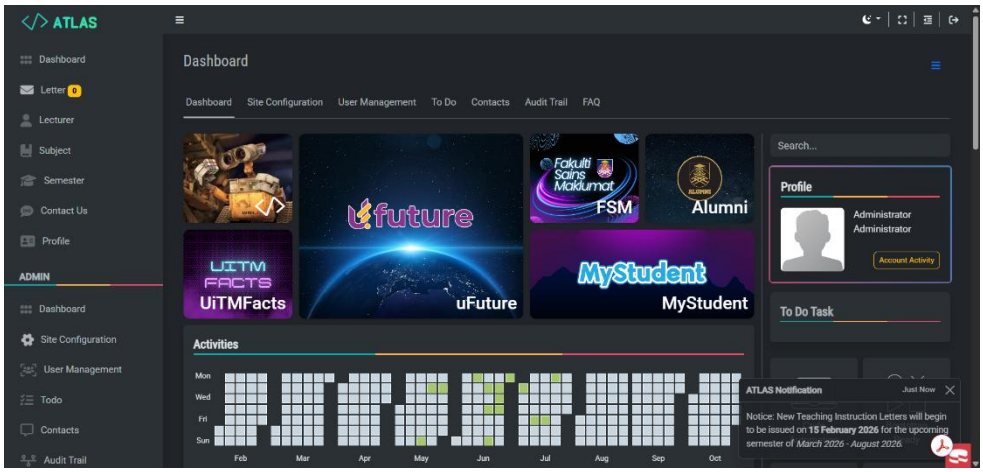


Figure 2: ATLAS Main Dashboard

MAIN DASHBOARD

The Dashboard offers a centralized overview of system activities through a visual activity heatmap and real-time notifications. It provides rapid navigation to core modules, audit trails, and integrated faculty portals like uFuture and MyStudent.

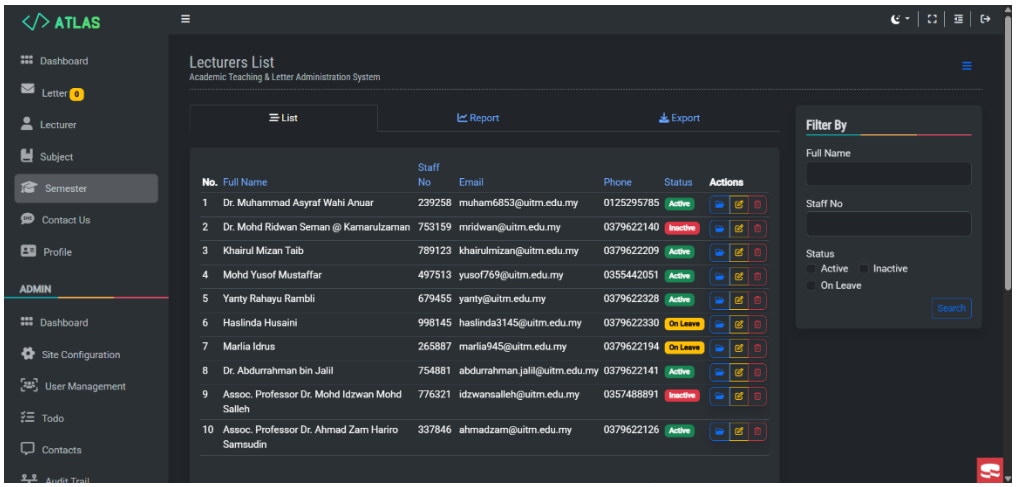


Figure 3: ATLAS Lecturers Management Interface

LECTURER MANAGEMENT

This module enables full CRUD functionality for lecturer records, including staff IDs, contact details, and employment status. Administrators can efficiently edit and update profiles via the actions menu or remove records using the delete tool. The system features a dedicated filter panel that allows precise searches by Full Name, Staff Number, and Status (Active, Inactive, or On Leave) to manage large datasets easily.

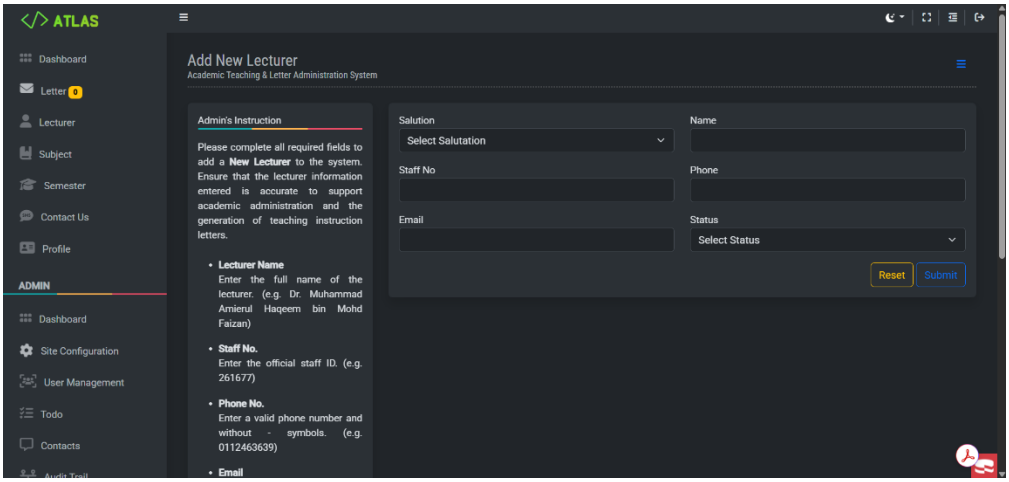


Figure 4: Add New Lecturer Page

ADD NEW LECTURER FORM

The "Add New Lecturer" interface provides a structured form to create and register new staff into the system. It captures essential data including Salutation, Name, Staff ID, and Contact Details, with a dedicated "Admin's Instruction" panel to ensure data accuracy for letter generation.

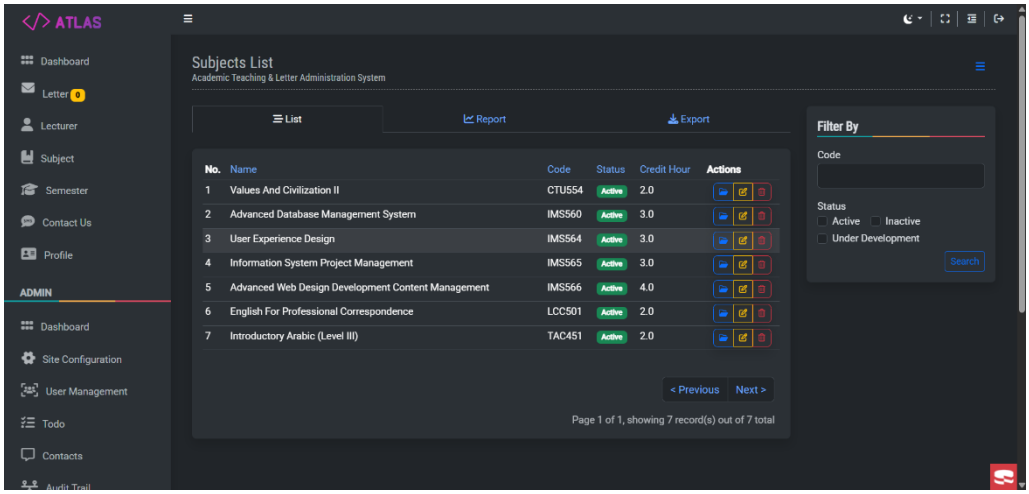


Figure 5: ATLAS Subjects Management

SUBJECT MANAGEMENT

This module manages the faculty catalog, including Subject Codes and Credit Hours. It enables full CRUD functionality and a search filter to sort subjects by Code or Status.

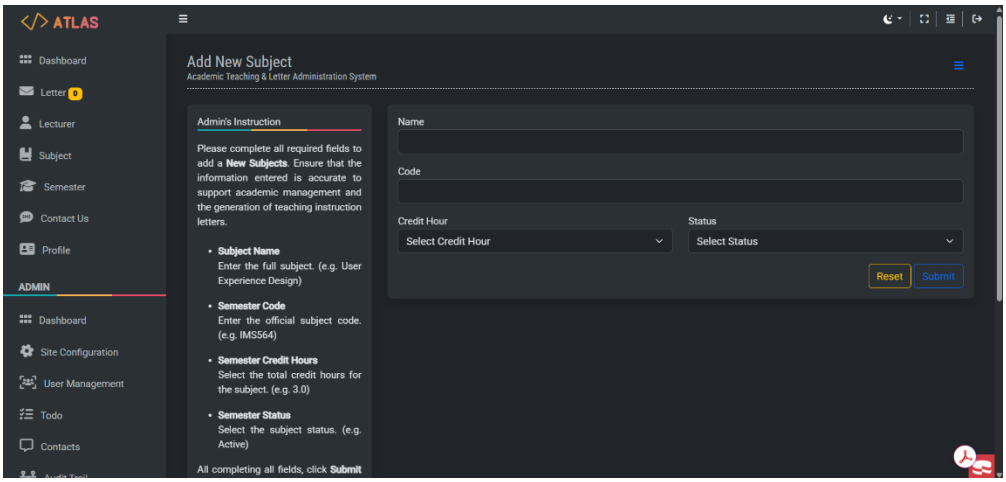


Figure 6: Add New Subject Page

ADD NEW SUBJECT FORM

This interface provides a structured form for the creation of new subject records. It captures critical data such as Subject Name, Code, and Credit Hours, featuring a side-panel for Admin's Instructions to ensure data consistency for academic administration.

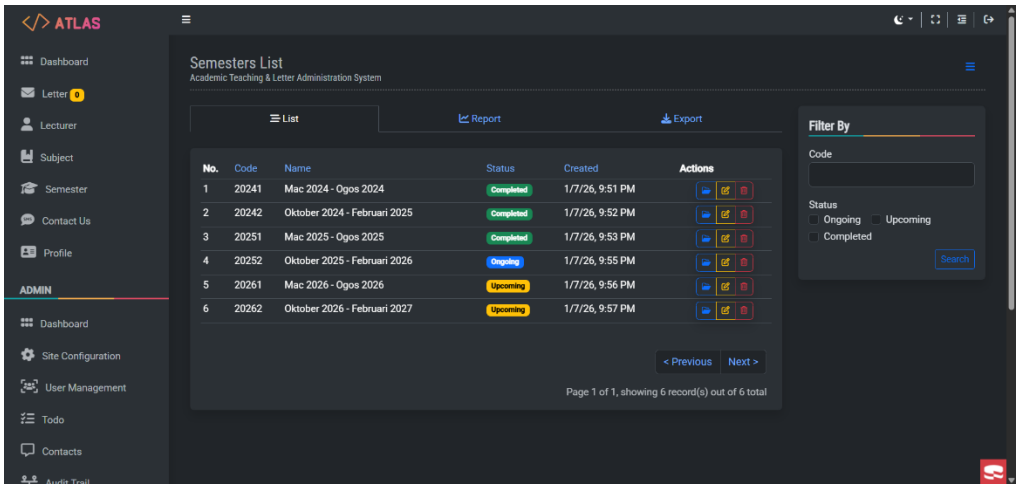


Figure 7: ATLAS Semesters Management

SEMESTER MANAGEMENT

This page manages academic periods, displaying Semester Codes, Names, and Status. It enables full CRUD functionality and includes a Filter By panel to quickly search for records by Code or specific status types like Ongoing, Upcoming, or Completed.

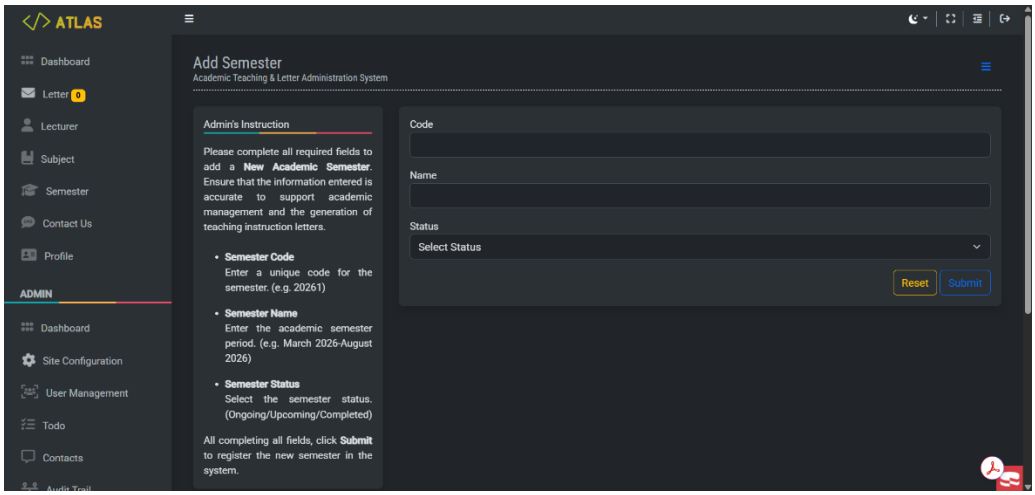


Figure 8: Add New Semester Page

ADD NEW SEMESTER FORM

This interface features a standardized form to register new academic periods into the system. It collects the Semester Code and Name while requiring a Status selection, supported by an integrated instructional guide to ensure data precision for administrative workflows.

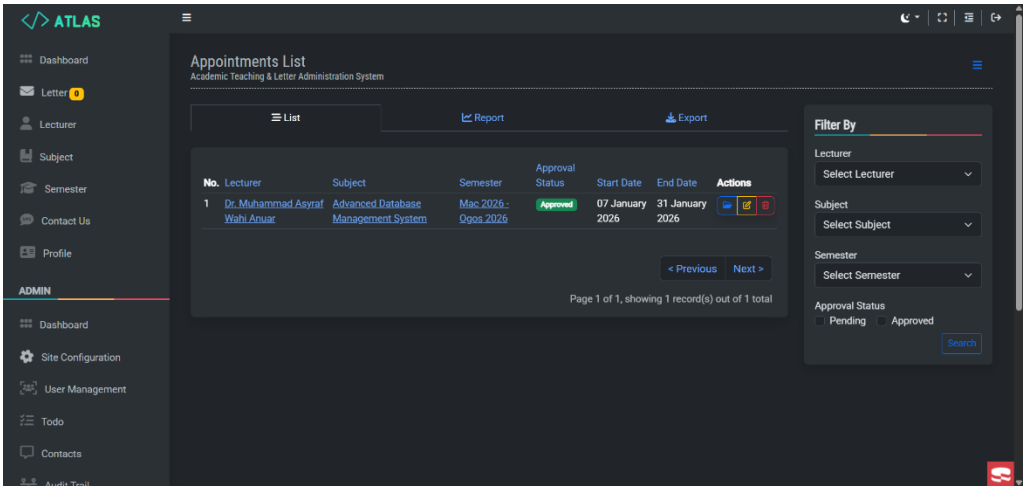


Figure 9: ATLAS Appointments Management

APPOINTMENT MANAGEMENT

This module tracks official lecturer appointments, displaying linked data for Lecturers, Subjects, and Semesters alongside their Approval Status. It supports full CRUD operations and features a Filter By panel to search by specific personnel, academic periods, or status (Pending/Approved).

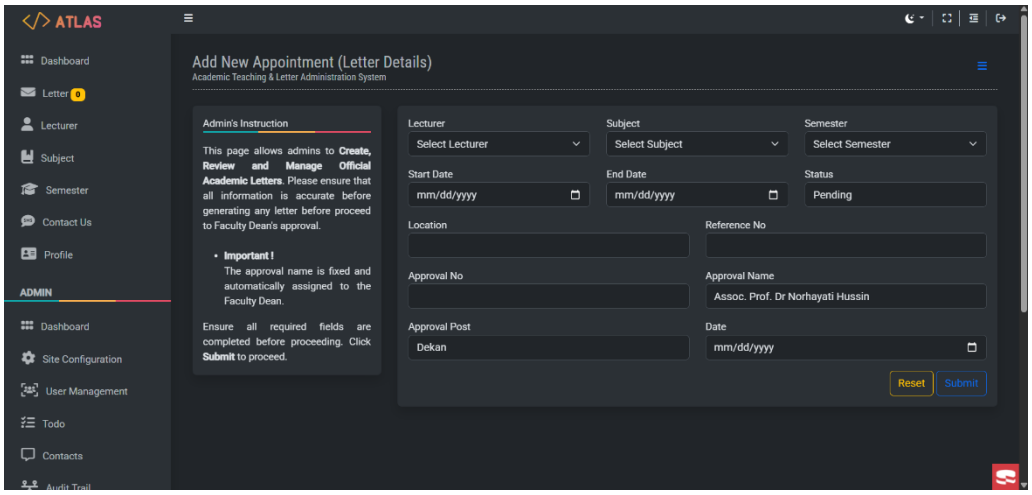


Figure 10: ATLAS Add New Appointment

ADD NEW APPOINTMENT FORM

This interface facilitates the creation and review of official academic letters. It captures specific details including Start/End Dates, Reference Numbers, and Locations, while automatically assigning the Faculty Dean as the fixed approval authority to ensure administrative consistency.

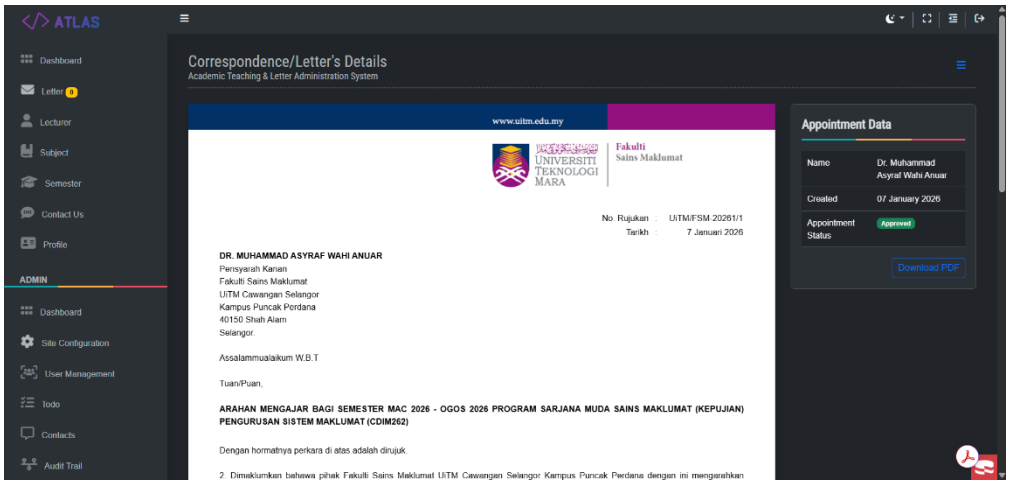


Figure 11: ATLAS Correspondence View Page

LETTER'S PREVIEW AND DETAILS

The Letter Details page provides a real-time document preview of generated teaching instruction letters for final review. It features an automated Approval Status tracker and a Download PDF function, allowing administrators to export official, faculty-branded letters once they are approved by the Faculty Dean.

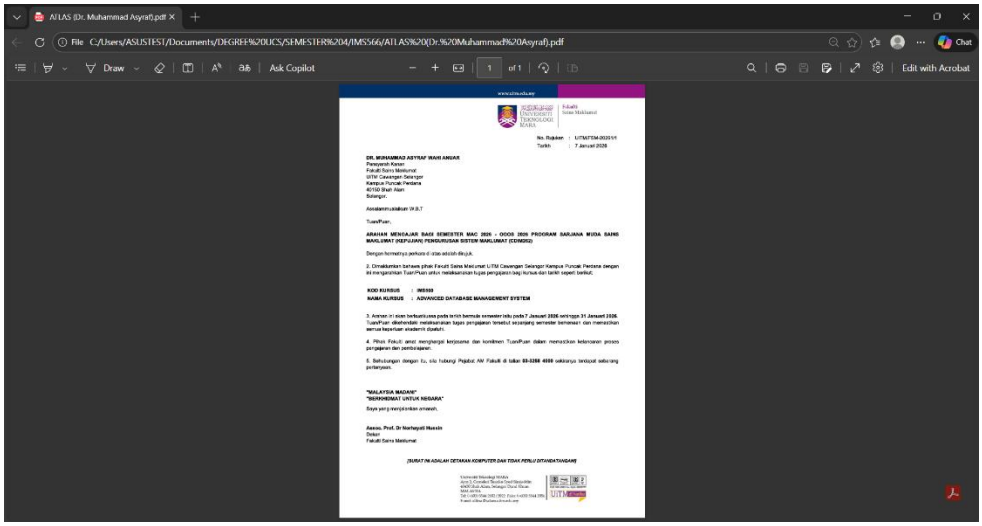


Figure 12: ATLAS Final Generated PDF Document

PDF OUTCOME

The PDF outcome represents the final official document generated by the system, formatted with official faculty letterheads and standardized academic layouts. This document serves as a formal Teaching Instruction Letter, containing all validated data including subject codes, appointment dates, and the digital signature of the Faculty Dean.

1. **Standardized Branding**:
   Includes official university logos and contact information for professional correspondence.
2. **Data Integrity**:
   Automatically pull accurate record details to eliminate manual typing errors.
3. **Accessibility**:
   The file is exported in a universal PDF format, making it easy to distribute to lecturers via email or printed for physical records.

7.0 WORKFLOW OF FORM

ATLAS SYSTEM WORKFLOW
The following workflow outlines the sequential process for generating an official appointment letter within the ATLAS system.
Phase 1: Data Preparation (CRUD)
Before an appointment can be created, all prerequisite data must be registered in the system:
1. **Lecturer Registration**: Add lecturer profiles including Staff ID and contact details. 2. **Subject Setup**: Register subjects with their respective codes and credit hours. 3. **Semester Configuration**: Define the academic period (e.g., Mac 2026 - Ogos 2026) and set its status.
Phase 2: Appointment Creation
Once the data is ready, the administrator links the components together:
1. **Form Completion**: Select the specific **Lecturer**, **Subject**, and **Semester** from the dropdown menus. 2. **Letter Details**: Input the Reference Number, Start/End dates, and Location. 3. **Submission**: Click **Submit** to save the record; the status is initially set to **Pending**.

Phase 3: Review and Approval

The system transitions from data entry to formal documentation:

1. **List Monitoring**:
   Use the **Filter By** panel in the Appointment List to find "Pending" records.
2. **Detail Review**:
   Open the **Correspondence Details** to preview the letter layout and data accuracy.
3. **Final Approval**:
   Once verified, the status is updated to **Approved**.

Phase 4: Output Generation

The final step involves distributing the formal document:

1. **PDF Conversion**:
   Click the **Download PDF** button within the letter preview.
2. **Official Document**:
   The system generates a branded PDF containing Dean’s signature and official letterhead for distribution.

8.0 TEAM ROLES & CONTRIBUTION

PROJECT MANAGER AND LEAD DEVELOPER
As the primary architect and lead of the ATLAS project, this role ensures the overall success of the system.
1. **Strategic Planning:** Formulates the comprehensive project roadmap from the initial phase through to completion.
2. **System Architecture:** Acts as the primary visionary and designer of the system's core concepts and features.
3. **Technical Lead:** Takes full charge of the system's codebase and technical implementation.

SYSTEM ANALYST
This role focuses on operational integrity and supporting project leadership to maintain high system standards.
1. **Administrative Support:** Assists the Project Manager in overseeing daily progress and project milestones.
2. **Workflow Validation:** Performs rigorous quality checks on the system's workflow to ensure logical consistency.
3. **Error Detection:** Identifies and corrects overlooked technical or logic errors within the system's processes.

UI/UX DESIGNER
Responsible for the visual identity and user interaction, ensuring the system is both functional and aesthetically pleasing.
1. **Visual Strategy:** Evaluates and selects color palettes and the precise positioning of all interface elements.
2. **Feature Enhancement:** Suggests additional interface improvements to optimize the user experience.
3. **Design Evaluation:** Ensures every element on the dashboard and form aligns with modern design standards.

DATABASE ADMINISTRATOR
Manages the backbone of the system, ensuring that all academic data is stored securely and retrieved efficiently.
1. **Structural Integrity:** Verifies that the Entity Relationship Diagram (ERD) is perfectly synchronized with the PHPMyAdmin implementation.
2. **Data Management:** Ensures the database functions optimally and that all system inputs are accurately recorded.
3. **Troubleshooting:** Acts as the primary point of contact to resolve any technical issues involving database connectivity or queries.

DOCUMENTATION SPECIALIST
Ensuring the project is professionally presented through comprehensive reporting and clear instructional guides.
1. **Technical Reporting:** Authors the formal system reports and project documentation.
2. **Manual Development:** Organizes and structures the User Manual for end-users and administrators.
3. **Final Quality Assurance:** Conducts a final audit of the project report to ensure completeness, professional formatting, and accuracy.

9.0 CONTACT INFORMATION (SUPPORT)

For any questions regarding the contents of this report or to request further data, please reach out to the project leads:

PRIMARY POINT OF CONTACT	
MUHAMMAD AMIERUL HAQEEM (PROJECT MANAGER)	
Email: mdamierul10@gmail.com	Phone: 0102463637

SECONDARY CONTACT	
MOHAMMAD NURTHASNIF ZAFFRAN (SYSTEM ANALYST)	
Email: m.zaffran18@gmail.com	Phone: 0149717565

10.0 CONCLUSION & REFLECTION

Therefore, the Academic Teaching and letter Administration System (ATLAS) can be successfully used to demonstrate the primary goal of digitising and simplifying the administration of academic letters in a faculty setting. The system offers a structured and effective platform on which an administrator can control semesters, subjects, lecturers and official academic letters especially instructional teaching letters. ATLAS decreases reliance on hardcopy documents and promotes a more standardized and sustainable administrative procedure by using online workflow.

With the creation of ATLAS, an integration of CakePHP, the RECRUD framework, and auxiliary tools, including Laragon, phpMyAdmin, Bootstrap, and Font Awesome allowed showing how it is possible to address the actual administrative issues in an academic environment with the help of modern web technologies. Inclusion of a fixed approval authority by the Faculty Dean will provide consistency, accountability and adherence to faculty governance and the possibility of lecturers downloading letters in PDF format is an additive to accessibility and maintenance records.

Reflectively, this project has been a good practical knowledge on system development especially how to put to practice CRUD operations, structured frameworks, and user-friendly interfaces. The process of development also contributed to the realization of how systems work, database management and the significance of the compatibility of the system functionality and the user requirements. The difficulties to be overcome during development, including data consistency and the integration of various frameworks, were dealt with by testing and refinement.

Overall, ATLAS is an effective system of study that can be used in academic institutions to illustrate the role of digitisation in the administrative system. The acquired knowledge and skills in this project make it a firmer ground on web application development and system analysis, as well as point to the possible positive improvement and extension of the system in the real academic setting.