PROJECT REPORT

ON

Institution’s Innovation Council (IIC) Web Portal

Project I Submitted in fulfilment of the requirements

for the degree of

B. Tech in CSE (Artificial Intelligence & Machine Learning)

BY

Abhijeet(13030822034)

UNDER THE GUIDANCE OF

Dr. Sudipta Chakrabarty

Assistant Professor

CSE (Artificial Intelligence & Machine Learning) Department

TECHNO MAIN SALT LAKE

EM 4/1 SALT LAKE CITY, SECTOR V
KOLKATA – 700091
West Bengal, India
December, 2025

Techno Main Salt Lake

[Affiliated by Maulana Abul kalam Azad University of Technology (Formerly known as WBUT)]

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

Certificate of Recommendation

This is to certify that Abhijeet have completed their project report on: "Institution's Innovation Council (IIC) Web Portal", under the direct supervision and guidance of Dr. Sudipta Chakrabarty. We are satisfied with their work, which is being presented for the fulfilment of the degree of B. Tech in CSE (Artificial Intelligence & Machine Learning), Maulana Abul Kalam Azad University of Technology) (Formerly known as WBUT), Kolkata – 700064.

Dr. Sudipta Chakrabarty

(Signature of Project Supervisors)

Date:

Dr. Sudipta Chakrabarty

(Signature of Project Coordinator)

Date:

Dr. Soumik Das

(Signature of the HOD)

Date:

Techno Main Salt Lake

[Affiliated by Maulana Abul Kalam Azad University of Technology
(Formerly known as WBUT)]

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

Certificate of Approval

The foregoing **PROJECT I** work is hereby approved as a creditable study of **B.Tech Degree in CSE (Artificial Intelligence & Machine Learning)** and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or any statement made, opinion expressed or conclusion therein but approve this Assignment work only for the purpose for which it is submitted.

Name and Signature of Internal Examiners

Name and Signature of External Examiners

Acknowledgement

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to the teaching fraternity of the Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning) and especially **Dr. Sudipta Chakrabarty** for giving us this opportunity to undertake this project and also for supporting us through the entire project.

We also wish to express our gratitude to the **HOD, Prof. (Dr.) Soumik Das** and all our teachers of the Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning) for their kind-hearted support, guidance, and utmost endeavor to groom and develop our academic skills.

At the end, we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during the effort in shaping this concept till now.

Full Signature of the Candidates (with date)

1. -----------------------------------
   Abhijeet
   (13030822034)

Table of Contents

1. Introduction

Higher education institutions are undergoing a rapid digital transformation, where traditional administrative and academic processes are being replaced by modern computer-based systems. The Institution Innovation Council (IIC), formed under the Ministry of Education's Innovation Cell (MIC) and AICTE, functions as a key driver of innovation by promoting entrepreneurial thinking, fostering ideation, and facilitating student–faculty collaboration.

However, the efficiency of IIC is directly dependent on the quality of its information systems. Many colleges still rely on outdated static websites, which lack dynamic updating capabilities, real-time communication mechanisms, and interactive functionalities. These static systems require manual editing of HTML pages for even minor updates, causing delays and inconsistencies. As a result, crucial information such as notices, event schedules, meeting invitations, and IIC activity reports often becomes outdated or inaccessible.

Students frequently depend on physical notice boards, informal WhatsApp groups, or verbal communication, making the flow of information unreliable. Faculty members struggle to keep their profiles updated and often miss meeting notifications due to the absence of a centralized digital communication channel.

To overcome these limitations, this project proposes the development of a dynamic and database-driven IIC web portal that centralizes all operations, ensures real-time updates, supports role-based access, and enhances transparency across the institution. The proposed platform becomes a digital ecosystem that strengthens institutional communication, promotes innovation culture, and aligns with national goals of digital transformation in education.

The proposed portal aims to:

- Create a central hub for all IIC-related information
- Enable real-time updates without technical intervention
- Enhance communication between students, faculty, and management
- Maintain activity reports, event galleries, minutes of meetings, and achievements

Ultimately, this web portal moves the institution toward a smart digital ecosystem, minimizing manual work, improving transparency, and aligning with the national vision of Digital India and innovation-driven education.

2. Scope of the Project

The project aims to develop a fully dynamic IIC web portal that streamlines communication, event management, and record handling for students, faculty, and administrators.

2.1 Functional Scope

Admin Functionalities

- Manage notices dynamically (create, edit, categorize, archive).
- Schedule meetings with agenda, date, venue, and participants.
- Add/update faculty profiles and IIC member details.
- View and respond to student queries.
- Access analytics dashboard (activities, notices, queries, usage stats).

Faculty Functionalities

- Secure role-based login.
- Receive automatic meeting notifications.
- Maintain professional profile (publications, achievements, patents).
- Access personalized dashboard updates.

Student Functionalities

- View all notices and announcements in real-time.
- Check event details, deadlines, and participate.
- Submit queries digitally for quick resolution.
- Access information from anywhere without physical dependency.

2.2 Technical Scope

- Developed using Django MVT architecture.
- SQLite as backend database.
- Frontend built with HTML, CSS, JavaScript.
- Fully responsive design for mobile and web access.
- GitHub used for version control and collaboration.

The project covers a complete digital platform for managing IIC activities with dedicated features for admins, faculty, and students. It ensures smooth notice updates, event management, profile handling, query resolution, and real-time information access. With Django-based development, responsive UI, and structured storage, the system aims to deliver an efficient, scalable, and user-friendly portal for institutional innovation management.

3. Concepts and Problem Analysis

3.1 Cost Analysis Using COCOMO

COCOMO (Constructive Cost Model) is a widely used software estimation technique that helps to predict the required effort, development time, and cost of a software project based on the estimated size in KLOC (Kilo Lines of Code). For the proposed Institution Innovation Council (IIC) Web Portal, the project size is assumed to be approximately 50 KLOC, considering backend development, UI pages, and database modules.

Cost Estimation Technique:- Effort= a1*(KLOC)a2 PM

Assumptions:

- Estimated Project Size: 50 KLOC
- Development Mode: Organic
- Team Size: 5 Developers
- Cost per Developer per Month: ₹1000
- Project Duration (Academic Consideration): 6 Months
- Technology Used: Django, SQLite, HTML, CSS, JavaScript

Organic mode is suitable because:

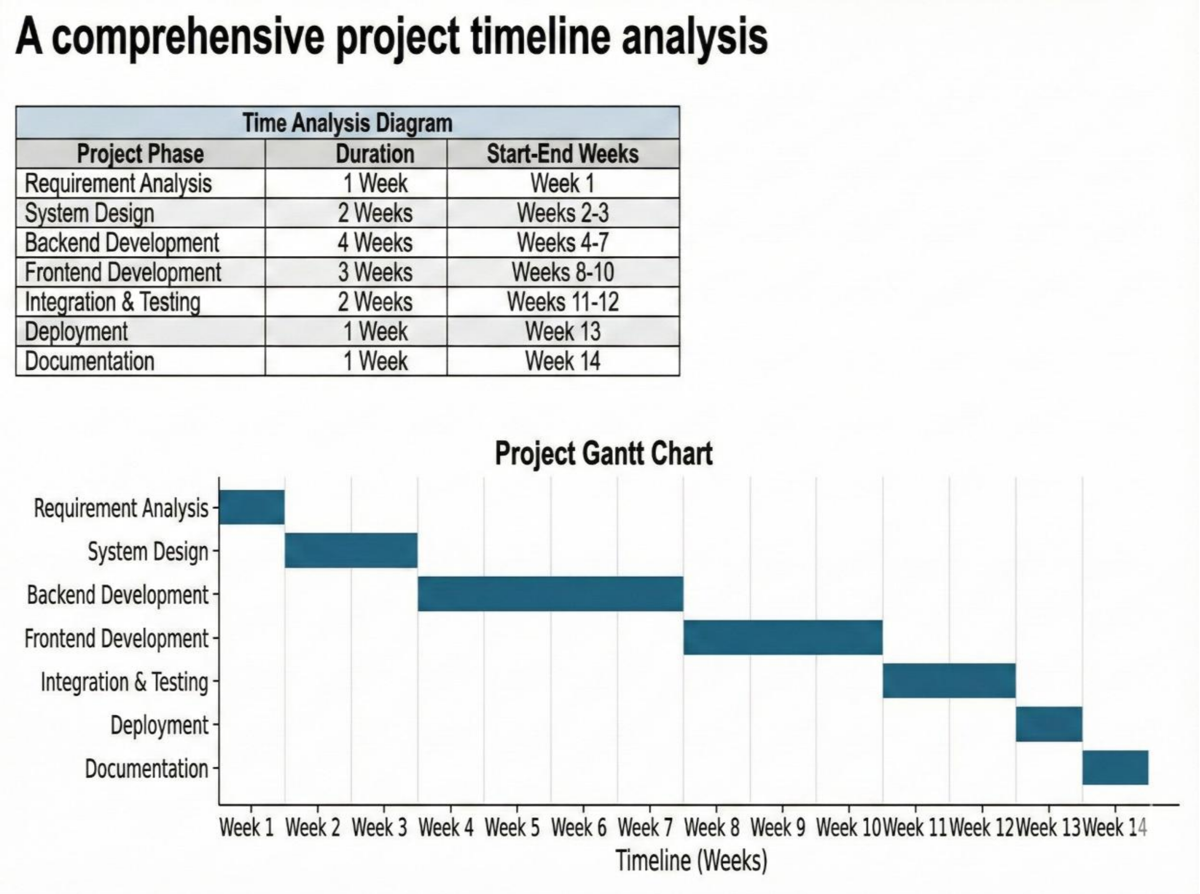
- Project complexity is low to moderate
- Small development team
- Familiar and easy-to-use technology stack

Cost Estimation (Academic Calculation)

Cost = Team Size × Cost per Developer × Duration
Cost = 5 × ₹1000 × 6
Total Development Cost ≈ ₹30,000

Note: Since this is a final-year academic project developed by students, no real salary or financial cost is involved. However, for the purpose of software engineering economics and COCOMO-based estimation, a hypothetical cost of ₹1,000 per developer per month has been assumed. Based on the calculated effort and development time, the notional project cost is approximately ₹30,000, representing the estimated industry-level cost for developing a similar system.

3.2 Time Analysis



A comprehensive project timeline analysis

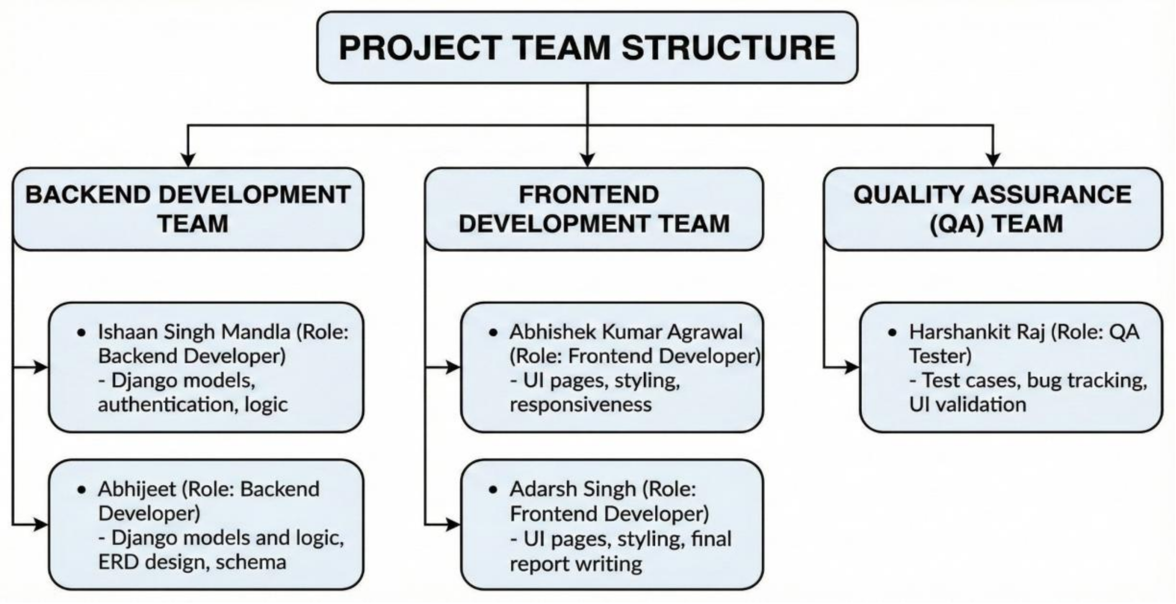
Time Analysis Diagram		
Project Phase	**Duration**	**Start-End Weeks**
Requirement Analysis	1 Week	Week 1
System Design	2 Weeks	Weeks 2-3
Backend Development	4 Weeks	Weeks 4-7
Frontend Development	3 Weeks	Weeks 8-10
Integration & Testing	2 Weeks	Weeks 11-12
Deployment	1 Week	Week 13
Documentation	1 Week	Week 14

Project Gantt Chart

3.3 Team Structure

Roles and Responsibilities

Member	Role	Responsibilities
Ishaan Singh Mandla	Backend Developer	Django models, authentication, logic
Harshankit Raj	QA Tester	Test cases, bug tracking, UI validation
Abhijeet	Backend Developer	Django models and logic, ERD design, schema
Abhishek Kumar Agrawal	Frontend Developer	UI pages, styling, responsiveness
Adarsh Singh	Frontend Developer	UI pages, styling, final report writing


PROJECT TEAM STRUCTURE
BACKEND DEVELOPMENT TEAM
FRONTEND DEVELOPMENT TEAM
QUALITY ASSURANCE (QA) TEAM
• Ishaan Singh Mandla (Role: Backend Developer) - Django models, authentication, logic
• Abhijeet (Role: Backend Developer) - Django models and logic, ERD design, schema
• Abhishek Kumar Agrawal (Role: Frontend Developer) - UI pages, styling, responsiveness
• Adarsh Singh (Role: Frontend Developer) - UI pages, styling, final report writing
• Harshankit Raj (Role: QA Tester) - Test cases, bug tracking, UI validation

3.4 Software Configuration Management (SCM)

Software Configuration Management (SCM) is carried out to ensure that every change made during the development of the IIC Web Portal is controlled, tracked, and documented properly. It helps maintain code integrity, avoid version conflicts, and ensures that the team can work collaboratively without overwriting each other’s work. SCM minimizes risks, maintains quality, and allows smooth development even when multiple developers work simultaneously.

Activities Performed

1. Version Control

- Git and GitHub are used to maintain the project repository.
- Source code, documentation, and configuration files are stored and managed centrally.
- Different branches are created for development, testing, and production to avoid conflicts and ensure safe updates.

2. Change Tracking

- Every update or modification is committed with a descriptive message.
- Commit history helps identify who made changes, what was modified, and when.
- Previous versions can be restored easily in case of bugs or design issues.

3. Build Management

- Each feature or module is developed and tested in its branch before merging.
- Continuous testing ensures stability and reduces chances of errors in deployment.
- Only verified builds get merged into the main branch.

4. Release Management

- Project development is divided into milestones (ex: authentication module, notice system, dashboard, etc.).
- Stable versions are released after successful testing of each milestone.
- Documentation is updated before every release to maintain traceability.

3.5 Quality Assurance Plan

Quality Assurance was carried out to ensure that the IIC Web Portal is reliable, secure, user-friendly, and performs smoothly. The QA process focused on detecting issues early, maintaining coding standards, and verifying that all features work as expected.

QA Techniques Used

1. Static Code Review

- Code was reviewed before merging to maintain standards and readability.
- Helped detect logical issues early and ensured cleaner code.
- Code is reviewed by team members before merging into the main branch.

2. Functional Testing

- Each module such as login, notices, queries, and dashboard was tested.
- Ensured that features worked according to requirements under different scenarios.
- Positive and negative test scenarios are evaluated to ensure robustness.

3. Form Validation Testing

- Input fields such as login, event registration, query form, faculty profile forms, etc., are tested for validation.
- Input fields tested for incorrect or incomplete entries.
- Prevented invalid data and improved data accuracy.

4. Security Testing

- Login system, authentication flow, and password protection were verified.
- Basic checks performed against unauthorized access and SQL injection.

5. Performance Testing

- Checked loading speed and response time under moderate usage.
- Ensured smooth functioning without lag.

6. Cross-Browser & Device Testing

- Tested on Chrome, Firefox, Edge, and mobile view.
- Verified responsive layout and consistent UI experience.

3.6 Risk Management

Risk management is an essential part of software development to identify potential issues that could affect the success of the project. The aim is to predict risks early, understand their impact, and apply preventive measures to reduce failures during development and deployment. The following risks were identified for the IIC Web Portal, along with their impact level and mitigation strategies.

Risk Details and Mitigation

1. Technical Risk

- Description: Possibility of server failure, crashes, or database corruption that may lead to data loss or website downtime.
- Impact: High, as the system may become inaccessible and important information may be lost.
- Mitigation: Frequent database backups are scheduled, and code is version-controlled to restore the system quickly in case of failure.

2. Data Risk

- Description: Incorrect or incomplete data entry, especially during faculty profile updates or form submissions.
- Impact: Medium, as inaccurate data affects credibility and system accuracy.
- Mitigation: Input validation is applied on forms, ensuring only correct and complete information is stored in the system.

3. Operational Risk

- Description: Delay in module integration or development stages due to coordination or dependency issues.
- Impact: Medium, which can extend project completion time.
- Mitigation: Weekly review meetings and progress monitoring ensure timely development and smooth integration.

4. Human Risk

- Description: Team member unavailability due to workload, personal issues, or academic commitments.
- Impact: Medium, as it can slow development or halt certain tasks.
- Mitigation: Task distribution plans and redundancy strategies are used so others can continue work if a member is unavailable.

4. Literature Survey

A literature survey helps in understanding existing research, previous implementations, and the technological need behind developing a new system. Various studies and academic papers emphasize that modern educational institutions are moving toward dynamic and interactive web portals instead of static websites. This shift is driven by the increasing requirement for real-time access to information, transparency, and innovation-supportive digital environments.

Research findings suggest that:

- Traditional static websites lack flexibility and require manual updates, making them unsuitable for institutions where notices, events, and reports need frequent posting. Minor content changes demand technical editing, which is time-consuming and inefficient.

- Role-based access control is essential for security and content management. Literature highlights that user authentication, privilege separation (admin, faculty, student), and controlled access help maintain data integrity and prevent unauthorized modifications.

- Dynamic notice and announcement systems improve institutional communication by ensuring instant updates, reducing dependency on physical notice boards, WhatsApp messages, or verbal communication. Real-time broadcasting of events, deadlines, and meetings promotes better participation.

- Digital innovation ecosystems are becoming integral in colleges, especially under initiatives like IIC (Institution Innovation Council) and MIC. Studies indicate that digital platforms promote entrepreneurship, idea submission, collaboration, and documentation of achievements.

- Centralized information management platforms reduce communication gaps between students, faculty, and administration. A single system allows stakeholders to view updates, submit queries, access event details, and maintain records efficiently.

5. Software Requirement Specification

A Software Requirement Specification (SRS) is a comprehensive, structured document that captures, analyzes, and presents all requirements of a software system before development begins. It describes the system's functionality, expected behavior, performance conditions, constraints, interfaces, and the environment in which it will operate. As a formal contract between stakeholders and developers, the SRS ensures clarity and avoids misinterpretations during software development.

The SRS specifies functional requirements, such as modules for notice management, faculty profiles, login authentication, and query handling, along with non-functional requirements such as usability, security, reliability, performance efficiency, and scalability. In addition, it defines hardware/software interfaces, data flow, assumptions, design limitations, and dependency constraints. By documenting requirements early, it helps reduce development risks, prevents rework, saves cost and time, and guides testing, validation, and maintenance activities.

The main objectives of an SRS are to:

- Precisely document user and system requirements in a structured format
- Provide a clear foundation for system design, development, and implementation
- Support testers in generating test cases and validation procedures
- Serve as a reference document throughout the software lifecycle

By defining inputs, outputs, control flow, and business rules clearly, the SRS ensures that the end product meets the expectations of users and institutional needs. It transforms conceptual ideas into technical language, guiding the development team towards building a reliable and efficient software solution.

5.1 Abstract

This project introduces a dynamic, interactive, and database-driven web portal designed to digitize and streamline the operations of the Institution Innovation Council (IIC). Traditional college websites are often static, manually updated, and lack interactivity, leading to delayed communication and poor documentation of events. The proposed system overcomes these limitations by providing real-time updates, automated content management, and a collaborative digital ecosystem for innovation activities.

The platform allows administrators to publish notices instantly, schedule meetings, upload activity reports, and maintain event records without modifying any core code. Faculty members can update professional details, add achievements such as research publications or patents, and receive automated meeting notifications.

Students can access announcements, view event details, and submit queries digitally, eliminating physical barriers and dependency on notice boards or informal communication channels.

Features like role-based access control, centralized database storage, query handling, responsive UI, and automated communication tools contribute to transparency, improved usability, and a smoother workflow within the institution. Overall, this portal enhances digital infrastructure, increases accessibility, and promotes innovation culture.

5.1.1 Introduction

The SRS document functions as the development blueprint of the IIC Web Portal. It clarifies how the system should function, what features must be implemented, and what constraints or environments the software must operate under. By documenting system requirements early, misunderstandings are reduced, planning becomes easier, and development flows systematically toward defined goals.

This section covers the core intention of the system: to digitalize event management, notice distribution, meeting scheduling, documentation, and profile management processes within IIC. It describes how multiple user roles—Admin, Faculty, and Students—will interact with the system according to assigned permissions to ensure controlled and secure access.

The SRS also highlights constraints such as chosen technology stack, security considerations, response speed expectations, compatibility requirements, and resource availability. It mentions system dependencies like frameworks, database servers, internet connectivity, and third-party tools. By establishing all functional and non-functional expectations clearly, the SRS aligns development, testing, and deployment around a shared understanding.

It ultimately ensures synchronized communication among stakeholders, enabling smooth development flow, minimizing rework, and ensuring the final product meets institutional expectations effectively.

5.1.2 Purpose

The primary purpose of the Institution Innovation Council (IIC) Web Portal is to establish a centralized, systematic, and digital platform that manages all innovation and activity-related operations within the institution. In many colleges, IIC-related updates are spread across physical notice boards, WhatsApp groups, and static web pages that must be manually edited for every update. This leads to delayed circulation of information, confusion during events, and loss of important data over time. The proposed system addresses these challenges by bringing all IIC functions under a unified interface that works online, remains accessible 24/7, and updates information in real time.

The system acts as a single-window access point, meaning users no longer need to check multiple platforms for event announcements, meeting schedules, activity reports, or faculty achievements. Notices can be posted instantly by the admin, faculty can maintain and update their professional records anytime, and students can view important updates within seconds. By converting traditional manual procedures into digital workflows, the portal minimizes paperwork, saves time, reduces dependency on physical communication, and enhances operational efficiency.

Another important purpose of this project is to encourage an innovation-driven environment within the institution. When events, competitions, webinars, achievements, and startup-related opportunities are showcased digitally and regularly, students feel more informed and motivated to participate. Increasing visibility naturally increases engagement and contribution from the student community.

From the administrative perspective, the portal offers a structured dashboard where authorized users can post notices, schedule meetings, and manage reports without technical background. This automation eliminates repetitive work and ensures data consistency across updates. The system also incorporates role-based access control, meaning different users (Admin, Faculty, Student) have different levels of authority. This approach maintains transparency and accountability—every activity such as notice upload, event creation, or profile update is traceable to a specific logged-in user, which improves security and responsibility handling.

Overall, the purpose of developing this web portal is to replace scattered communication with a modern and integrated platform that supports seamless information flow, secure data handling, efficient collaboration, and digital documentation of all IIC-related activities. It aims to strengthen institutional innovation management and enhance productivity through technology-driven automation.

5.1.3 Scope

The scope of the project includes designing and implementing a dynamic web-based platform for IIC management, integrating essential operational modules into a single digital system. The system will support notice publishing, event scheduling, faculty profile maintenance, student query handling, and record management seamlessly.

A relational database will store all information such as user details, notices, events, meeting records, and activity logs. A secure role-based authentication system will differentiate features accessible to Admin, Faculty, and Student users. This ensures data privacy, controlled editing rights, and structured content flow.

The automated meeting notification feature will alert faculty members about scheduled meetings, reducing missed communication. The interactive noticeboard will update instantly, replacing manual website edits. Faculty profile management will allow structured maintenance of achievements, while students can submit queries and receive responses digitally.

The portal will offer a responsive and user-friendly interface, ensuring compatibility across desktops, laptops, tablets, and mobile devices. The modern UI design improves usability and encourages wider adoption.

In summary, the scope covers full development, integration, deployment, and testing of a dynamic IIC portal enabling seamless workflows, centralized communication, and improved innovation management within the institution.

The SRS delivers a clearly defined framework for designing, developing, and deploying the IIC Web Portal. It outlines the system expectations, scope, purpose, architecture preferences, and required features in detail. By documenting requirements systematically, the SRS ensures that the development team and institutional stakeholders work toward a shared vision, reducing risks, enhancing planning, and improving project execution. Ultimately, this specification ensures the creation of a reliable, accessible, and scalable digital solution tailored to support the innovation ecosystem of the institution.

In conclusion, the Software Requirement Specification clearly defines the expectations, scope, objectives, and functional boundaries of the IIC Web Portal. It provides a structured foundation for development by outlining the system’s purpose, modules, interactions, and technical requirements. By documenting these elements in detail, the SRS ensures that every stakeholder involved in the project maintains a common understanding of how the portal will operate and evolve. This clarity contributes to effective planning, smooth development, accurate testing, and successful implementation of the system, ultimately resulting in a reliable and efficient digital platform for managing all IIC-related activities.

6. Theoretical Background

The theoretical background provides the conceptual foundation required to understand the development, functioning, and design of the IIC Web Portal. It explains the underlying technologies, models, frameworks, programming languages, architectural approaches, development methodology, and digital transformation theories related to the project. This section ensures that the reader understands the technical logic behind choosing specific tools and approaches used in building the system.

6.1 Web Development Fundamentals

Web applications function using a client-server architecture, where the browser acts as the client sending requests, and the backend server processes the data and returns the output. The communication happens using HTTP/HTTPS protocols.

Key Concepts:

- Client-Side: Handles user interface and interactions.
- Server-Side: Handles processing, authentication, business logic.
- Request-Response Cycle: Client requests → Server processes → Response sent back.

This architecture ensures remote accessibility and scalability, suitable for institutional systems like IIC portals.

6.2 Frontend Technologies

Frontend is responsible for the look, layout, structure, and interaction of web pages.

HTML (HyperText Markup Language)

- Backbone of web structure
- Defines text, headings, tables, forms, etc.
- Used for notice pages, login forms, event display, profile pages

CSS (Cascading Style Sheets)

- Used for design, theme, color, responsiveness
- Makes UI attractive and readable
- Ensures consistent design across devices

JavaScript

- Adds interactivity (dynamic content display)
- Enhances user experience without page reload

Responsive Design Theory

- Website adapts to mobile, tablet, desktop automatically
- Uses CSS Media Queries & Flex/Grid layouts

A responsive UI is critical for students accessing the portal on phones.

6.3 Backend Technology – Django Framework

Django is a high-level Python web framework used to develop secure and maintainable applications.

Why Django is Used

- Built-in authentication system
- Supports rapid development
- Follows *MVT Architecture*
- Secure (prevents most common cyber attacks)

MVT (Model–View–Template) Architecture

Component	Role in System
Model	Database handling of notices, users, events, queries
View	Business logic and request handling
Template	HTML pages rendered for users

This architecture separates logic from design, making debugging and updates easier.

6.4 Database Concepts

A database is a systematic collection of data that allows efficient storage, retrieval, and management of information. For this project, a relational database model is used, where data is organized in the form of interlinked tables. Each table contains rows (records) and columns (fields), enabling structured data storage and fast query operations. Relational databases are preferred for institutional applications due to their reliability, security, and ease of maintenance.

In the IIC Web Portal, SQLite is chosen as the primary database due to its lightweight nature, server-less architecture, and seamless integration with Django. SQLite stores data locally in a single file, making it suitable for academic and web-based projects where deployment is simple and system resources are limited. In later scalability scenarios, the database can also be migrated to MySQL or PostgreSQL without major architectural changes.

Key Database Roles in the Project

1. Storage of Institutional Data
   The database maintains and organizes various forms of information such as:
   - Notices and announcements
   - Faculty details, achievements, publications

- Meeting schedules and event records
- Student queries and responses

2. Each dataset is stored in a separate table allowing easy management and retrieval.

3. CRUD Operations Support
   The system performs the following basic operations:
   - Create: Adding new notices, meeting entries, faculty profiles
   - Read: Viewing notices, retrieving queries, fetching events
   - Update: Modifying profile details, editing notices, responding to queries
   - Delete: Removing outdated notices or irrelevant records

4. These operations ensure the portal remains dynamic, updated, and relevant at all times.

5. Data Consistency and Integrity
   - Relational constraints maintain linkages between tables (e.g., faculty ID linked with profile data).
   - Validation rules ensure no invalid or duplicate entries are stored.
   - Data integrity prevents corruption or inconsistency during updates.

6. Example: A notice cannot be assigned to a non-existing faculty ID due to relational constraints.

Why SQL Database is Suitable for This Project

- Handles structured institutional information efficiently
- Supports fast query execution and indexing
- Provides transactional safety and ACID compliance
- Easy integration with backend frameworks like Django

- Suitable for multi-user access with controlled permissions

6.5 Authentication & Role-Based Access Control (RBAC)

RBAC restricts features based on user roles:

User Role	Access Privilege
Admin	Full control over notices, faculty, queries, events
Faculty	Profile update + Query response + View information
Student	View notices/events + Submit queries

RBAC improves security, accountability, controlled content editing, and prevents misuse.

6.6 Software Engineering & SDLC Concepts

The system follows Software Development Life Cycle principles:

Phases Used

1. Requirement Collection
2. System Analysis
3. Designing UI/DB architecture
4. Implementation (Coding)
5. Testing (Functional/Performance/Security)

6. Deployment

7. Maintenance & Updates

This stepwise approach ensures smooth development and quality outcome.

6.7 Dynamic Website vs Static Website Theory

Static websites require manual HTML changes for every update.
Dynamic websites retrieve and update information through a database automatically.

Static System	**Dynamic Portal (Proposed System)**
Manual updating required	Automatic notice updates
No database	Database-driven
Not interactive	Interactive UI
Slow information flow	Real-time publishing

Dynamic systems are essential for modern institutional management.

6.8 Digital Communication & Innovation Ecosystem Theory

The IIC portal supports digital transformation in academics, essential for NAAC/IIC ranking and modern workflow.

Benefits include:

- Faster information flow
- Increased student engagement

- Reduced paper usage
- Innovation-friendly culture
- Transparent data management

6.9 Security & Data Protection Theory

Security is crucial since portal holds faculty details, notices, student queries.

Measures used:

- Hashed passwords
- Session-based login
- Input validation
- Protection against SQL injection, CSRF

This ensures reliable and safe usage.

6.10 Version Control Theory (Git & GitHub)

SCM is used to track code updates and avoid conflicts.

- Multiple developers can work together
- Every commit is recorded
- Rollback is possible if errors occur

The theoretical background establishes the foundation for understanding the technological principles, development methods, system architecture, and engineering concepts used to build the IIC Web Portal. It connects theory with implementation, justifying every component — from web technologies to database logic, security models, SDLC methodology, and digital innovation framework. These concepts collectively ensure that the portal is secure, scalable, responsive, and capable of transforming institutional communication into an efficient digital environment.

7. Approach, Methods & Algorithms

The development of the IIC Web Portal follows a structured methodology to ensure reliability, maintainability, and scalability. The approach involves requirement analysis, architectural planning, modular development, database design, and iterative testing. The system is built using Django as the backend framework and SQLite as the database, with HTML, CSS, and JavaScript for the frontend interface. The combination of technologies allows for dynamic content rendering, real-time data management, and secure user authentication.

The goal of the chosen approach is to build a centralized digital platform capable of managing notices, events, faculty profiles, and student queries efficiently. To achieve this, the development process is divided into multiple phases, each involving specific methods and algorithms tailored to the system's requirements.

7.1 Development Approach

The project adopts an incremental and modular development approach, where each feature is developed, tested, and integrated step-by-step. This ensures flexibility in enhancement and easier debugging.

Steps followed:

1. **Requirement Gathering & Analysis**
   - Identifying user needs (admin, faculty, student)
   - Understanding workflow of current system
   - Listing required features: notices, events, meetings, profiles, queries

2. **System Design & Architecture Planning**
   - Designing database structure & entity relationships
   - Creating UI layout and wireframes
   - Planning user access roles and navigation flow

3. **Backend & Frontend Development**
   - Building models, views, templates using Django MVT
   - Designing responsive UI using HTML/CSS/JS
   - Implementing user authentication and session management

4. **Testing & Quality Assurance**
   - Functional testing, validation checks, security testing
   - Fixing bugs and improving performance

5. **Deployment & Maintenance**
    - Hosting application
    - Regular updates and backups

This staged approach ensures smooth development and easier maintainability.

7.2 Methods Used

a) Software Development Model

The Incremental SDLC Approach is chosen, where project modules are implemented in parts:

Increment	Feature Implemented
1st	Login & Authentication System
2nd	Notice & Event Management
3rd	Meeting Scheduler + Faculty Profiles
4th	Query Handling and Dashboard

Advantage:

- Faster outputs, early testing
- Easy to add new features without reworking whole system

b) MVT Architectural Method

Django uses Model–View–Template architecture, which separates business logic, database, and UI.

Component	**Method Used**	**Role**
Model	ORM, SQL	Handles database tables
View	Backend Logic	Processes user request & response
Template	HTML/CSS/JS	Renders final UI to user

Benefits:

- Faster development
- Secure authentication built-in
- Clean code & easy maintenance

7.3 Algorithms Used in Implementation

Although the project is not algorithm-heavy like ML systems, several logic-based algorithms ensure smooth operations.

a) Authentication & Login Algorithm

Input: Username, Password

Process:

- Validate input format
- Check credentials in database
- If match: generate session token
- Redirect to respective dashboard
- Else: show "invalid credentials" message

Output: Authorized access or error notice

Ensures secure login & user verification.

b) Notice/Event Publishing Algorithm

I. Admin creates new notice/event
II. Store details in database
III. Automatically display it on portal homepage
IV. Ordered by latest timestamp
V. Expired events hidden or archived

Enables dynamic content display without code editing.

c) Query Handling Algorithm

I. Student submits query → Stored in DB
II. Faculty/Admin views pending queries
III. Response message saved → Mark as resolved
IV. Student notified of reply

Improves communication & response tracking.

d) Meeting Reminder Algorithm

I. Admin schedules meeting with date & time
II. System stores record in database
III. Email/notification triggered (if enabled)
IV. Faculty sees reminder on dashboard

Reduces missed meetings and improves participation.

e) Faculty Profile Update Algorithm

I. Faculty logs in → Opens profile section
II. Updates achievements/details → Form validation
III. Changes stored in database
IV. Profile displayed dynamically to users

Ensures updated academic records are visible.

7.4 Why This Approach Was Appropriate

- Modular structure simplifies updates and reduces complexity
- Django & SQLite combination ensures security and scalability
- Role-based access protects system from misuse
- Incremental development allows continuous improvement
- Algorithms automate time-consuming manual tasks

The approach, methods, and algorithms used in developing the IIC Web Portal ensure a structured, secure, and efficient solution for managing institutional innovation activities. By adopting Django MVT architecture, role-based control, and CRUD-based database management, the system performs dynamic operations in real-time while maintaining data integrity. The development methodology supports quality assurance, future scalability, and easy maintenance, making the portal a robust platform for academic innovation support.

8. Design/Solution/Methodology

8.1 Data Design

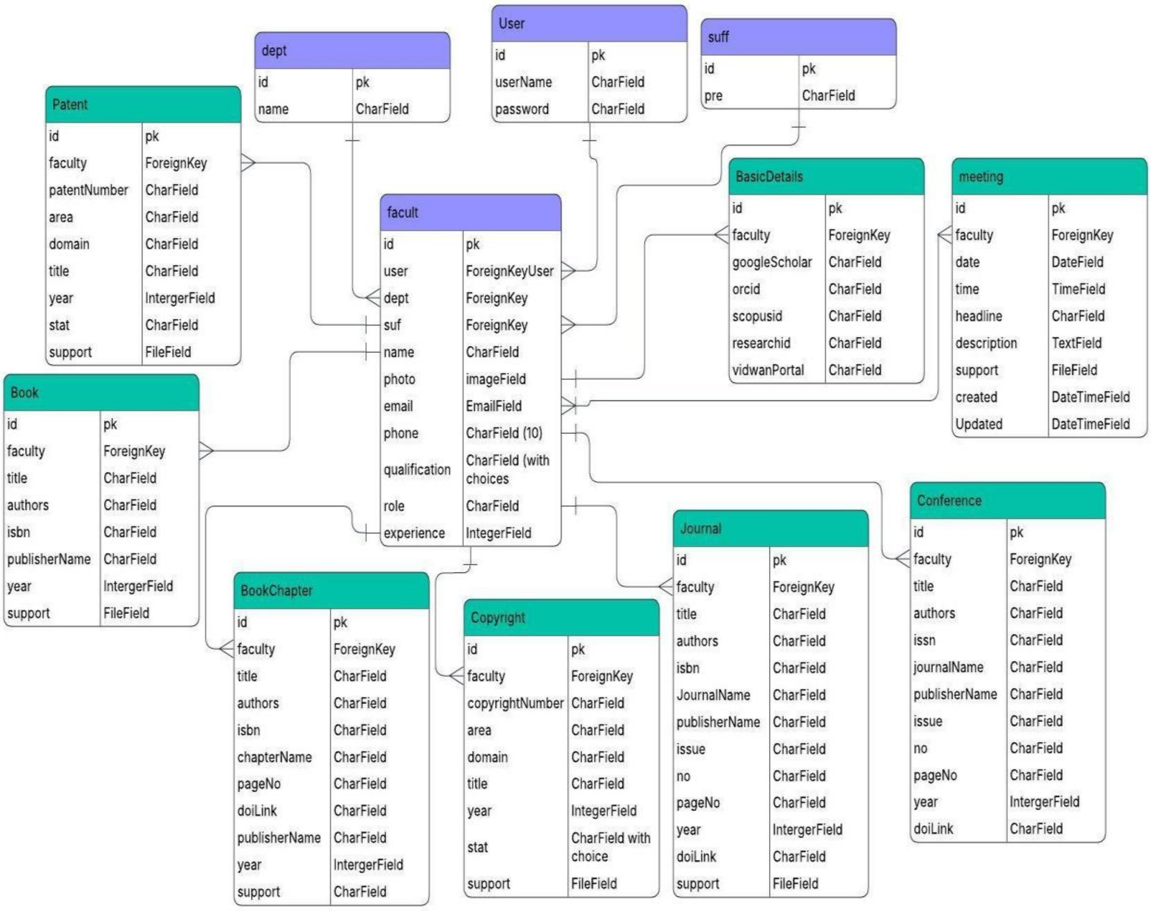
Data design focuses on how information is stored, organized, and accessed within the system. In the IIC Web Portal, a relational database model is used to ensure data integrity, consistency, and reduced redundancy. Each module such as users, achievements, activities, meetings, and gallery is stored in separate tables, which are linked using primary and foreign keys.

This structured approach ensures secure data storage, easy retrieval, and efficient CRUD (Create, Read, Update, Delete) operations across the system.

8.2 ER Diagram

The Entity Relationship (ER) diagram represents the database structure of the Dynamic Institution Innovation Council (IIC) website. It shows how key entities such as Admin, Faculty, Student, Notices, Meetings, Achievements, Activities, Queries, Gallery, and Certificates are connected and interact within the system.

The ER diagram defines entities, their attributes, and relationships to ensure organized data storage and efficient data management. For example, Admin manages notices and meetings, Faculty maintains profiles and achievements, and Students submit queries and view activities. These relationships help maintain data consistency and support smooth CRUD operations.



dept

id	pk
name	CharField

User

id	pk
userName	CharField
password	CharField

suff

id	pk
pre	CharField

Patent

id	pk
faculty	ForeignKey
patentNumber	CharField
area	CharField
domain	CharField
title	CharField
year	IntergerField
stat	CharField
support	FileField

facult

id	pk
user	ForeignKeyUser
dept	ForeignKey
suf	ForeignKey
name	CharField
photo	imageField
email	EmailField
phone	CharField (10)
qualification	CharField (with choices
role	CharField
experience	IntegerField

BasicDetails

id	pk
faculty	ForeignKey
googleScholar	CharField
orcid	CharField
scopusid	CharField
researchid	CharField
vidwanPortal	CharField

meeting

id	pk
faculty	ForeignKey
date	DateField
time	TimeField
headline	CharField
description	TextField
support	FileField
created	DateTimeField
Updated	DateTimeField

Book

id	pk
faculty	ForeignKey
title	CharField
authors	CharField
isbn	CharField
publisherName	CharField
year	IntergerField
support	FileField

Conference

id	pk
faculty	ForeignKey
title	CharField
authors	CharField
issn	CharField
journalName	CharField
publisherName	CharField
issue	CharField
no	CharField
pageNo	CharField
year	IntergerField
doiLink	CharField

Journal

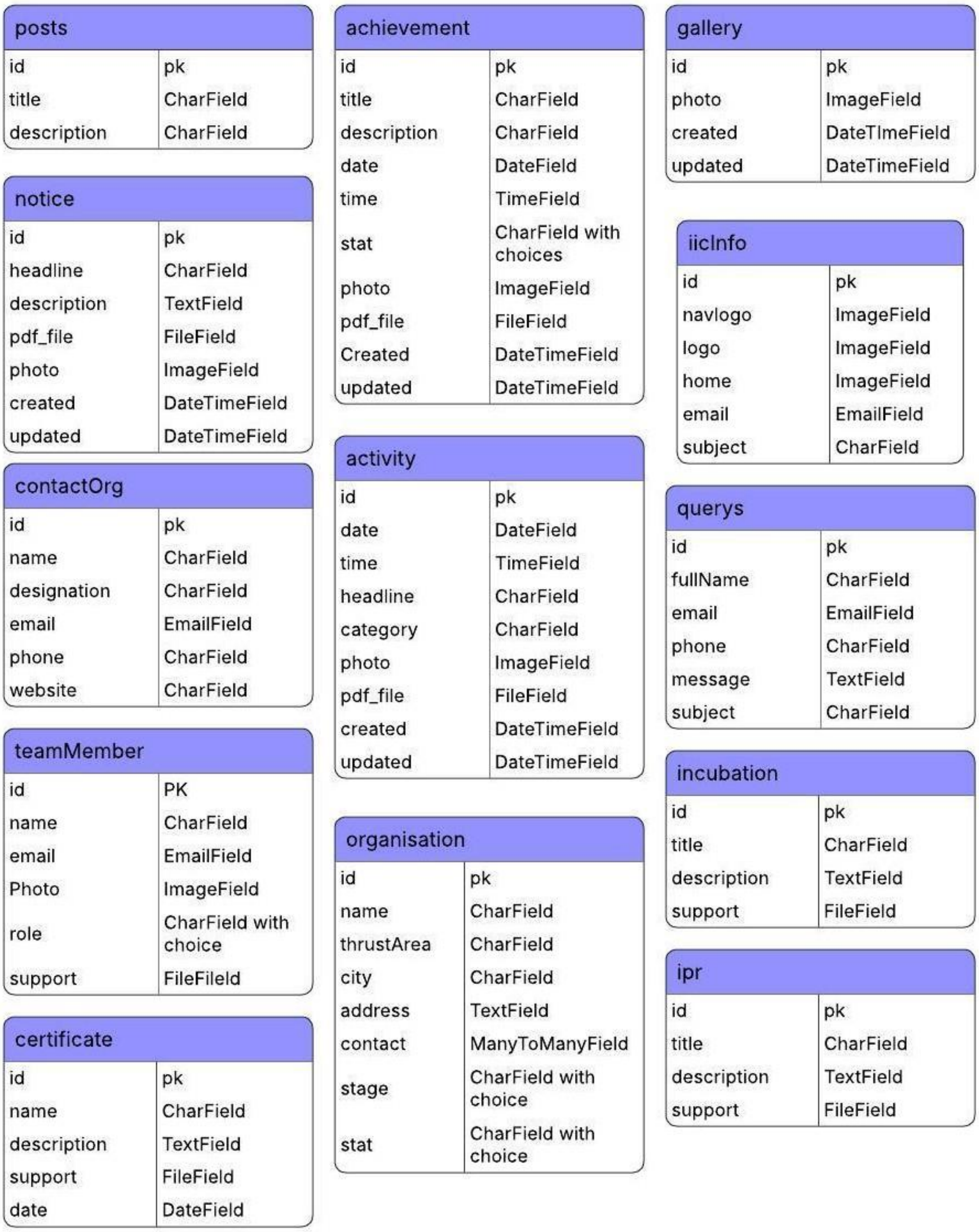
id	pk
faculty	ForeignKey
title	CharField
authors	CharField
isbn	CharField
JournalName	CharField
publisherName	CharField
issue	CharField
no	CharField
pageNo	CharField
year	IntergerField
doiLink	CharField
support	FileField

BookChapter

id	pk
faculty	ForeignKey
title	CharField
authors	CharField
isbn	CharField
chapterName	CharField
pageNo	CharField
doiLink	CharField
publisherName	CharField
year	IntergerField
support	CharField

Copyright

id	pk
faculty	ForeignKey
copyrightNumber	CharField
area	CharField
domain	CharField
title	CharField
year	IntegerField
stat	CharField with choice
support	FileField



posts	
id	pk
title	CharField
description	CharField

notice	
id	pk
headline	CharField
description	TextField
pdf_file	FileField
photo	ImageField
created	DateTimeField
updated	DateTimeField

contactOrg	
id	pk
name	CharField
designation	CharField
email	EmailField
phone	CharField
website	CharField

teamMember	
id	PK
name	CharField
email	EmailField
Photo	ImageField
role	CharField with choice
support	FileFileld

certificate	
id	pk
name	CharField
description	TextField
support	FileField
date	DateField

achievement	
id	pk
title	CharField
description	CharField
date	DateField
time	TimeField
stat	CharField with choices
photo	ImageField
pdf_file	FileField
Created	DateTimeField
updated	DateTimeField

activity	
id	pk
date	DateField
time	TimeField
headline	CharField
category	CharField
photo	ImageField
pdf_file	FileField
created	DateTimeField
updated	DateTimeField

organisation	
id	pk
name	CharField
thrustArea	CharField
city	CharField
address	TextField
contact	ManyToManyField
stage	CharField with choice
stat	CharField with choice

gallery	
id	pk
photo	ImageField
created	DateTImeField
updated	DateTimeField

iicInfo	
id	pk
navlogo	ImageField
logo	ImageField
home	ImageField
email	EmailField
subject	CharField

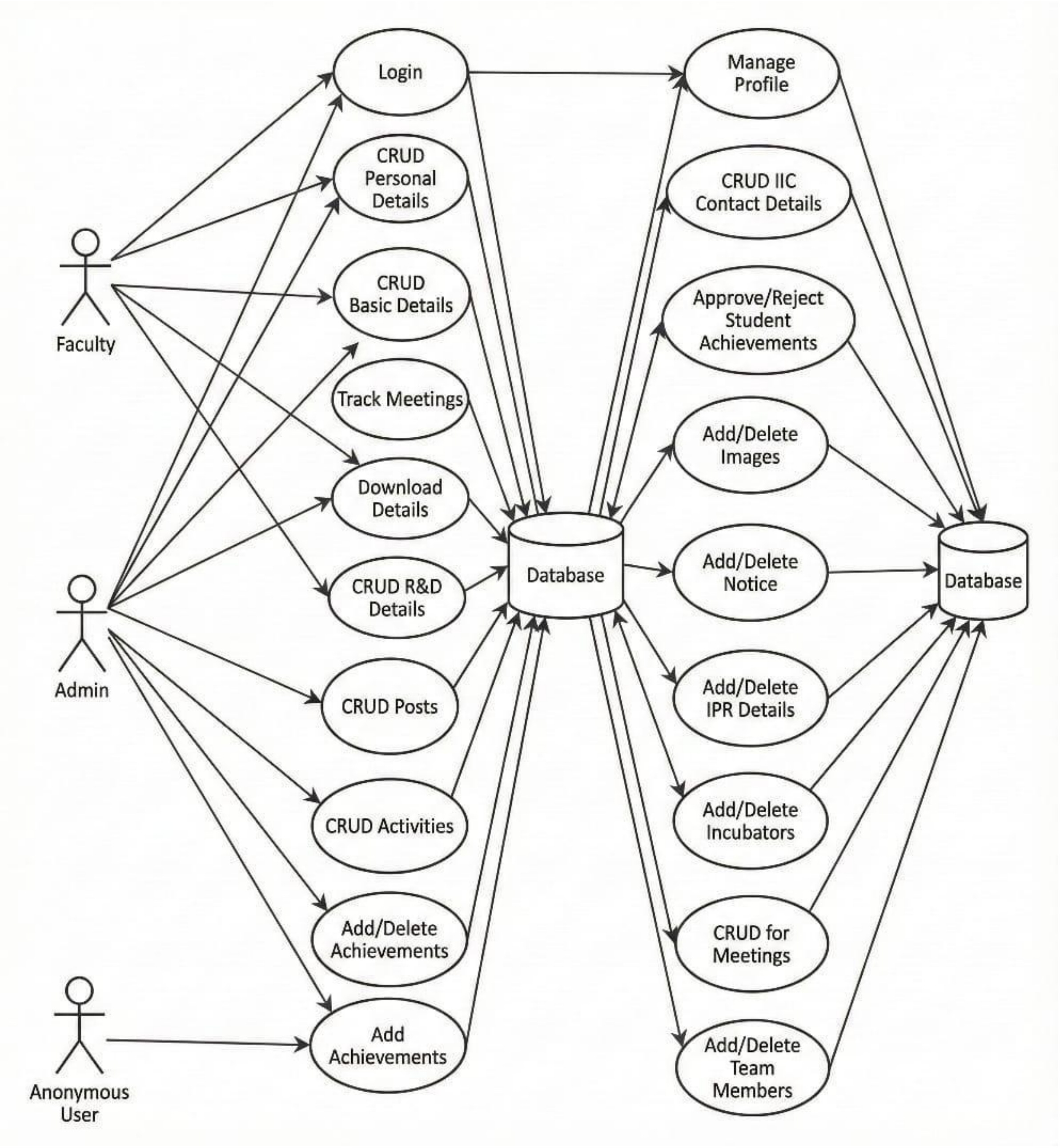
querys	
id	pk
fullName	CharField
email	EmailField
phone	CharField
message	TextField
subject	CharField

incubation	
id	pk
title	CharField
description	TextField
support	FileField

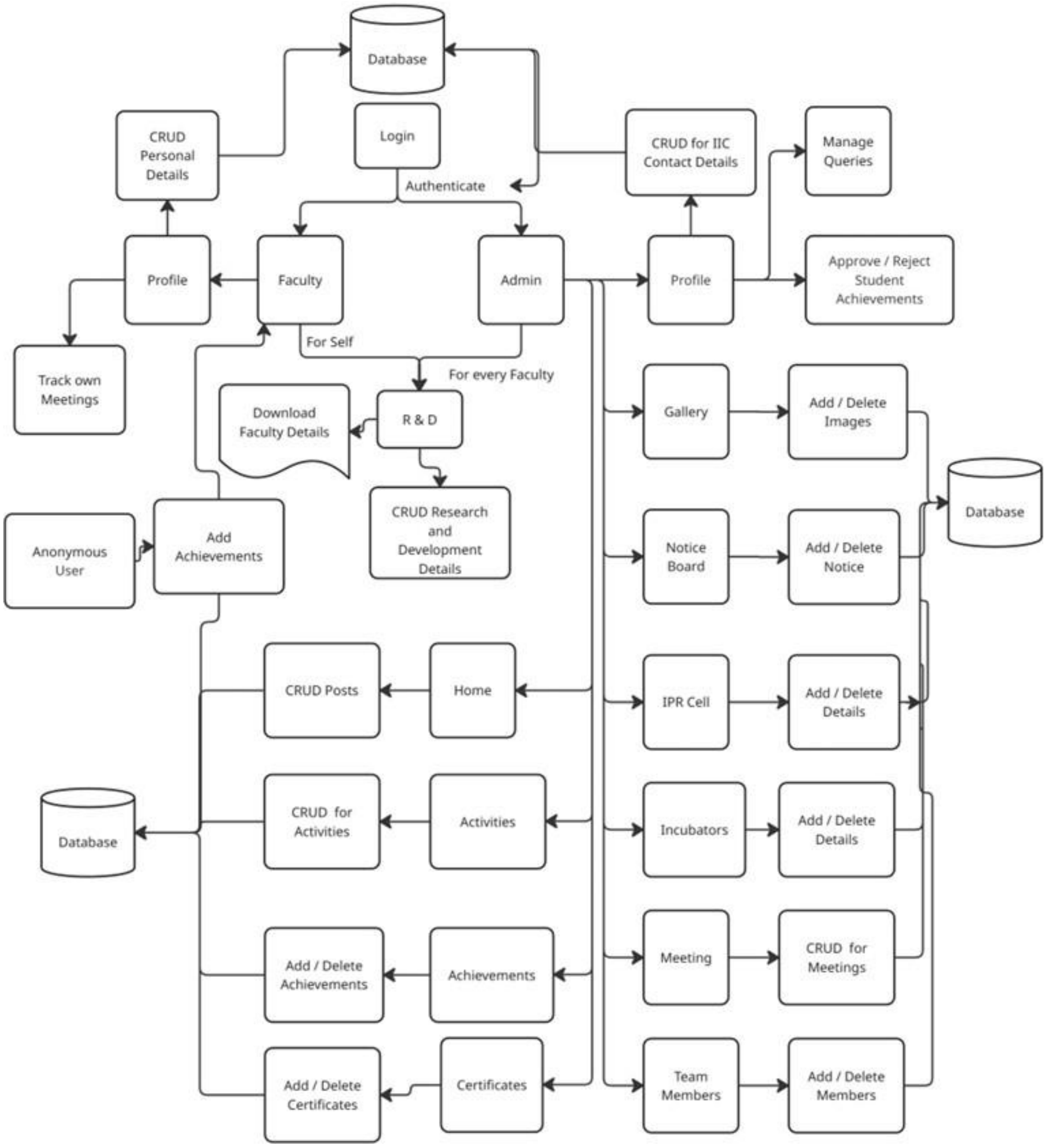
ipr	
id	pk
title	CharField
description	TextField
support	FileField

8.3 UML (Unified Modelling Language)

Use Case Diagram



8.4 Activity Diagram

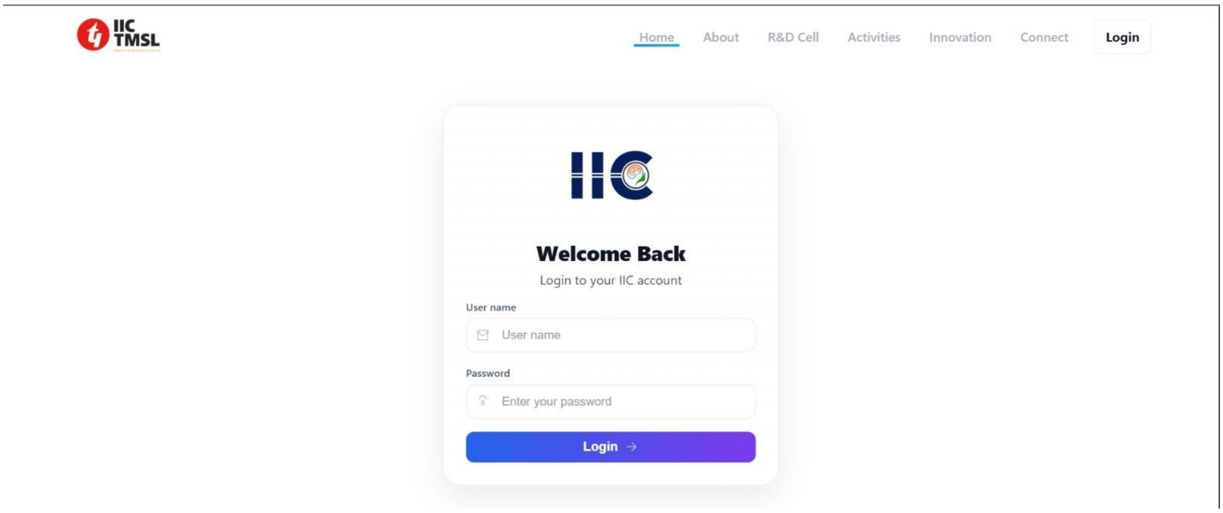


9. Procedural Design

Procedural Design explains the step-by-step working of the Institution Innovation Council (IIC) Web Portal. It shows how users interact with the system and how data is processed, stored, and displayed.

9.1 Faculty in Procedure

1. Faculty enters login credentials.
2. System validates credentials using Django authentication.
3. Login is redirected to Admin dashboard based on role.
4. Invalid login shows an error message.



Welcome Back

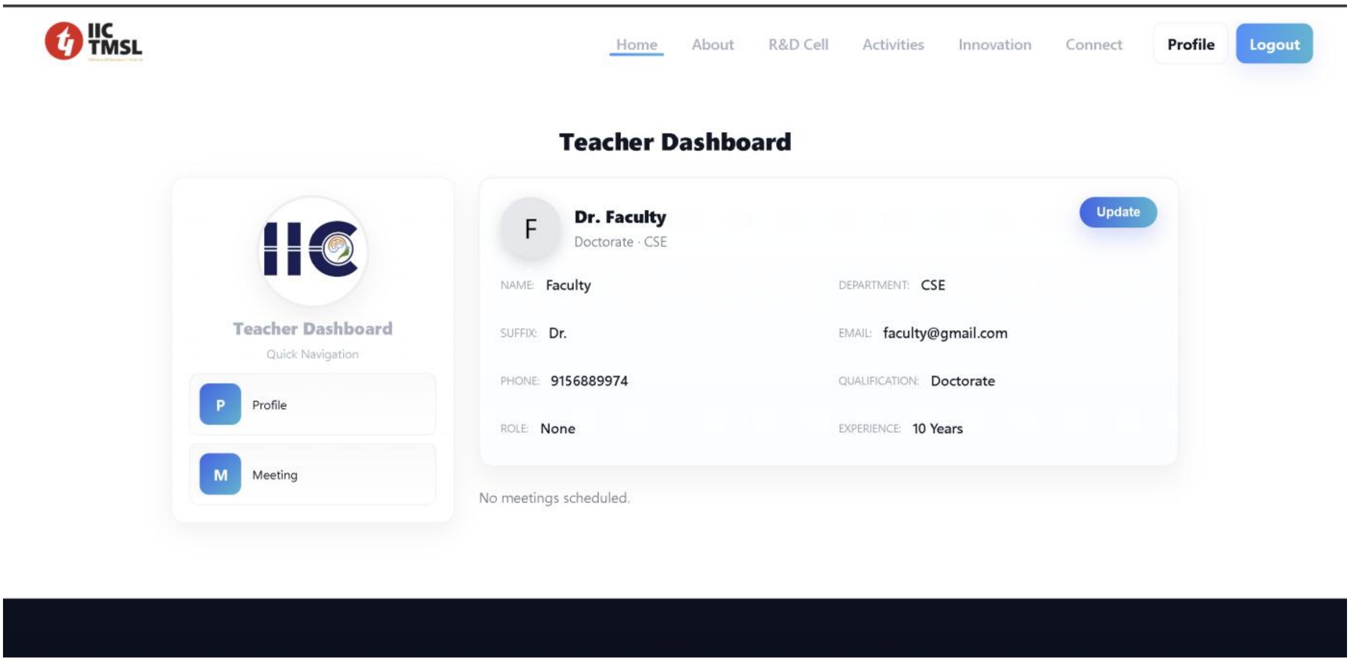
Login to your IIC account

User name: User name

Password: Enter your password

9.2 Dashboard Access Procedure

1. System loads dashboard after login.
2. Modules are displayed according to user role.
3. User selects required module.
4. Data is fetched and displayed dynamically.



Teacher Dashboard

Teacher Dashboard
Quick Navigation

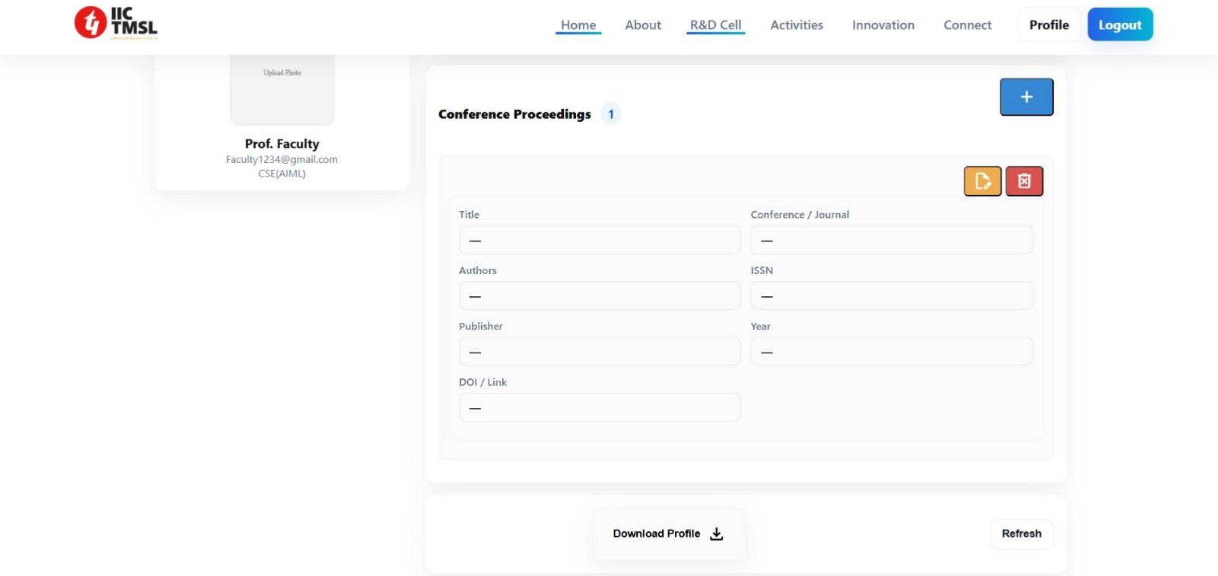
F
Dr. Faculty
Doctorate · CSE

NAME: Faculty
DEPARTMENT: CSE
SUFFIX: Dr.
EMAIL: faculty@gmail.com
PHONE: 9156889974
QUALIFICATION: Doctorate
ROLE: None
EXPERIENCE: 10 Years

No meetings scheduled.

9.3 Content Management (CRUD) Procedure

1. User adds, views, updates, or deletes records.
2. System validates input data.
3. Data is stored or updated in the database.
4. Changes appear instantly on the portal.



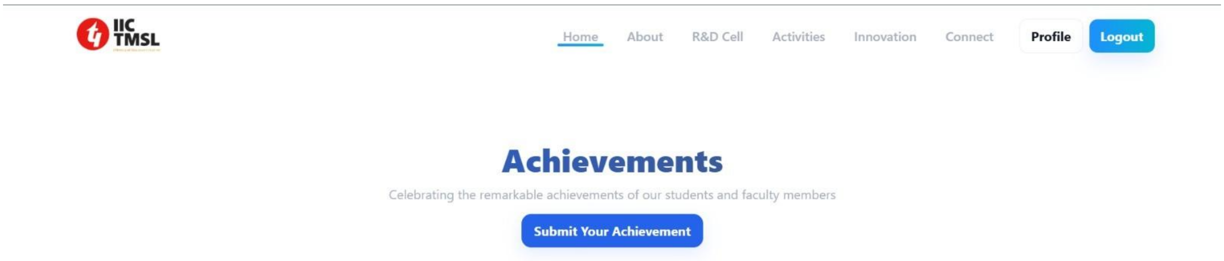
Prof. Faculty
Faculty1234@gmail.com
CSE(AIML)

Conference Proceedings 1

Title: —
Conference / Journal: —
Authors: —
ISSN: —
Publisher: —
Year: —
DOI / Link: —

9.4 Achievement Approval Procedure

1. Achievement is submitted by faculty or student.
2. Admin reviews the submission.
3. Approved achievements are published.

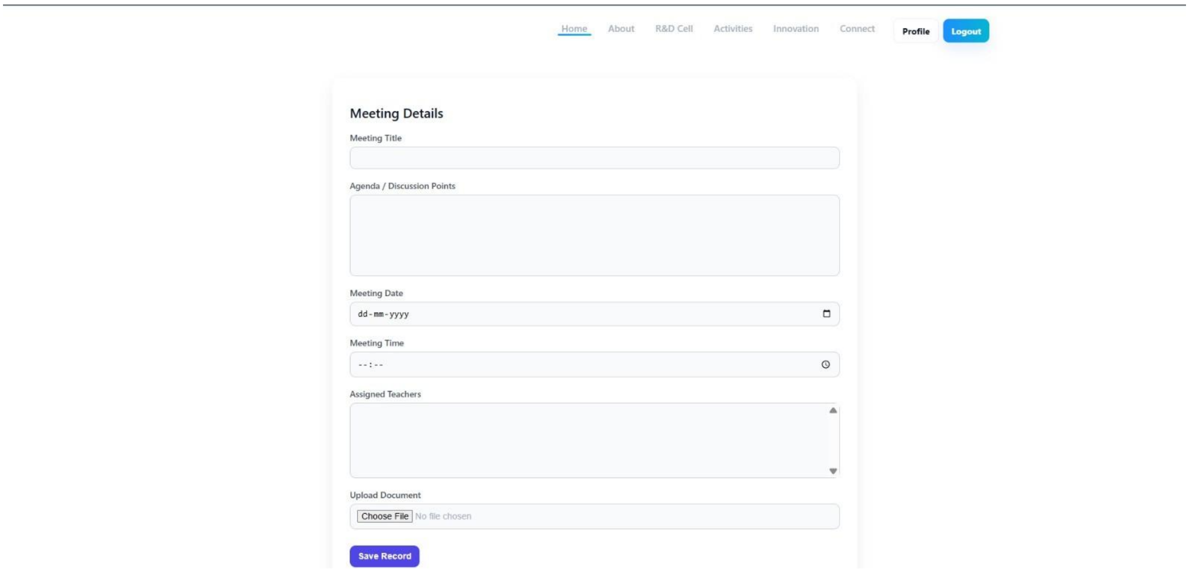


Achievements

Celebrating the remarkable achievements of our students and faculty members

9.5 Meeting Management Procedure

1. Admin schedules meetings.
2. Meeting details are saved in database.
3. Faculty can view meetings on dashboard.



Meeting Details

Meeting Title

Agenda / Discussion Points

Meeting Date

dd-mm-yyyy

Meeting Time

--:--

Assigned Teachers

Upload Document

Choose File No file chosen

10. Coding Standards Followed

To ensure code quality, readability, maintainability, and scalability, standard coding practices were followed throughout the development of the IIC Web Portal. The project follows widely accepted conventions for Python, Django, frontend technologies, and database handling.

10.1 Python & Django Coding Standards

- The project follows PEP-8 guidelines for Python coding, including proper indentation, meaningful variable names, and modular structure.
- Django's Model–View–Template (MVT) architecture is strictly followed to separate database logic, business logic, and presentation.
- Each app/module is organized into files such as `models.py`, `views.py`, `urls.py`, and `templates`, improving clarity and maintainability.
- Reusable code components are used to avoid redundancy and improve efficiency.

10.2 Naming Conventions

- Variables, functions, and methods are named using snake_case (e.g., `add_notice`, `faculty_profile`).
- Class names follow PascalCase (e.g., `FacultyProfile`, `NoticeModel`).
- Database tables and fields are named meaningfully to reflect their purpose.
- URLs are kept clean and readable to improve usability and debugging.

10.3 Frontend Coding Standards

- HTML files are structured properly using semantic tags for better readability.
- CSS is written in a modular manner to maintain consistency in design and layout.
- JavaScript is used only where necessary, mainly for validation and UI interaction.
- Responsive design principles are followed to ensure compatibility across devices.

10.4 Database and CRUD Standards

- All database interactions are performed using Django ORM, avoiding raw SQL wherever possible.
- CRUD operations (Create, Read, Update, Delete) are implemented in a structured and secure manner.

- Data validation is applied at both frontend and backend levels to ensure integrity.
- Foreign key relationships are used to maintain relational consistency between entities.

10.5 Security and Error Handling

- Django’s built-in authentication system is used for login and role management.
- Passwords are stored in encrypted (hashed) form.
- Role-based access control ensures only authorized users can perform sensitive operations.
- Proper exception handling is implemented to avoid system crashes and data loss.

10.6 Version Control Practices

- Git and GitHub are used for version control.
- Meaningful commit messages are written to track changes clearly.
- Code is tested before merging to the main branch to avoid errors.

11. RESULT SET ANALYSIS

Result Set Analysis evaluates the outputs produced by the IIC Web Portal after implementation and testing. It verifies whether the system functions according to the specified requirements and delivers accurate, reliable, and expected results for all user roles. The analysis is performed by executing different modules of the system and observing their behavior, data processing, and outputs.

Step 1: User Authentication Results

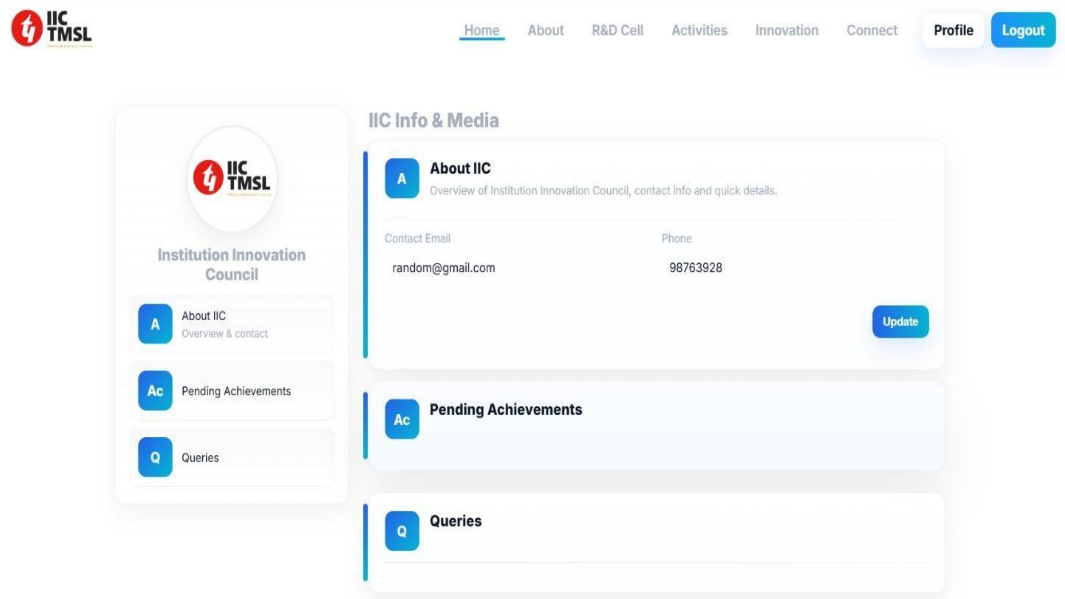
- Users (Admin, Faculty, Student) were tested using valid and invalid credentials.
- The system successfully authenticated valid users and redirected them to role-specific dashboards.
- Invalid login attempts were rejected with appropriate error messages.

Result: Authentication and role-based access worked correctly.

Step 2: Dashboard & Role-Based Access Verification

- Admin, Faculty, and Student dashboards displayed features according to assigned permissions.
- Restricted actions were inaccessible to unauthorized users.
- Session management ensured secure access until logout.

Result: Role-based access control functioned as expected.



IIC Info & Media

A About IIC

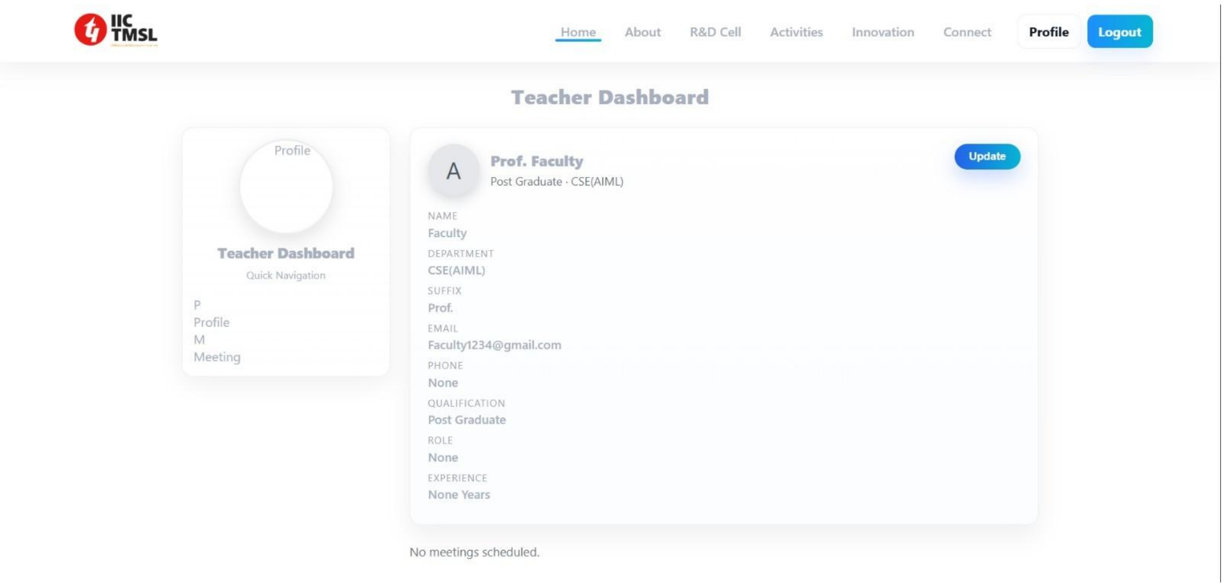
Overview of Institution Innovation Council, contact info and quick details.

Contact Email: random@gmail.com

Phone: 98763928

Ac Pending Achievements

Q Queries



Teacher Dashboard

Profile

Teacher Dashboard

Quick Navigation

A Prof. Faculty

Post Graduate · CSE(AIML)

NAME
Faculty

DEPARTMENT
CSE(AIML)

SUFFIX
Prof.

EMAIL
Faculty1234@gmail.com

PHONE
None

QUALIFICATION
Post Graduate

ROLE
None

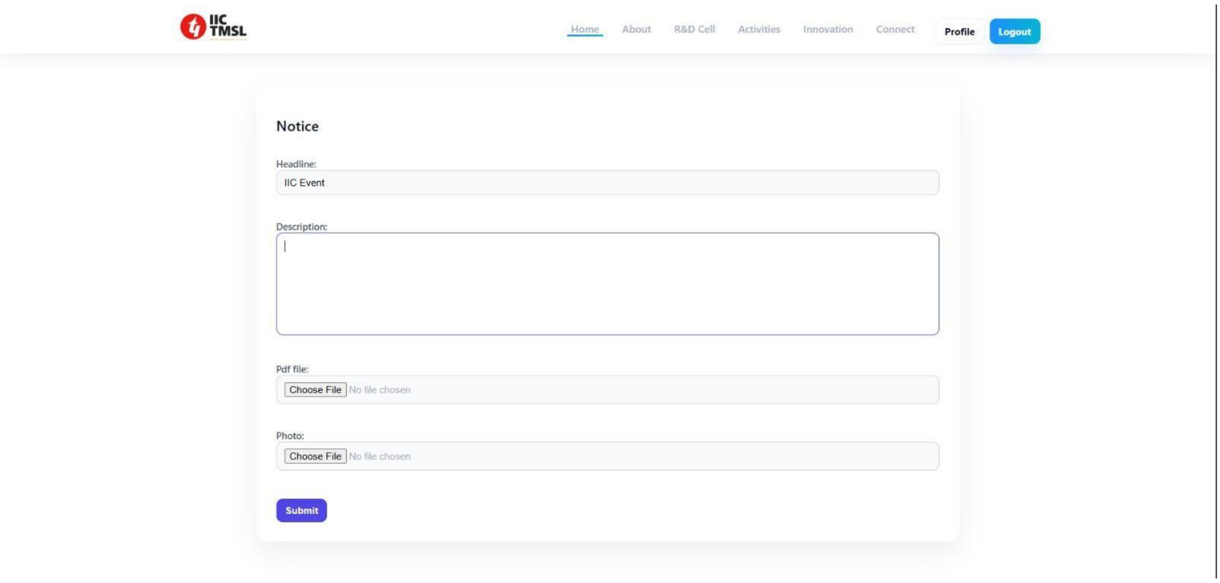
EXPERIENCE
None Years

No meetings scheduled.

Step 3: Notice Management

- Admin successfully added, updated, and deleted notices and posts.
- Newly added notices appeared instantly on the public portal.
- Archived or deleted notices were no longer visible.

Result: Dynamic content update was achieved without manual code changes.



Notice

Headline:
IIC Event

Description:

Pdf file:
No file chosen

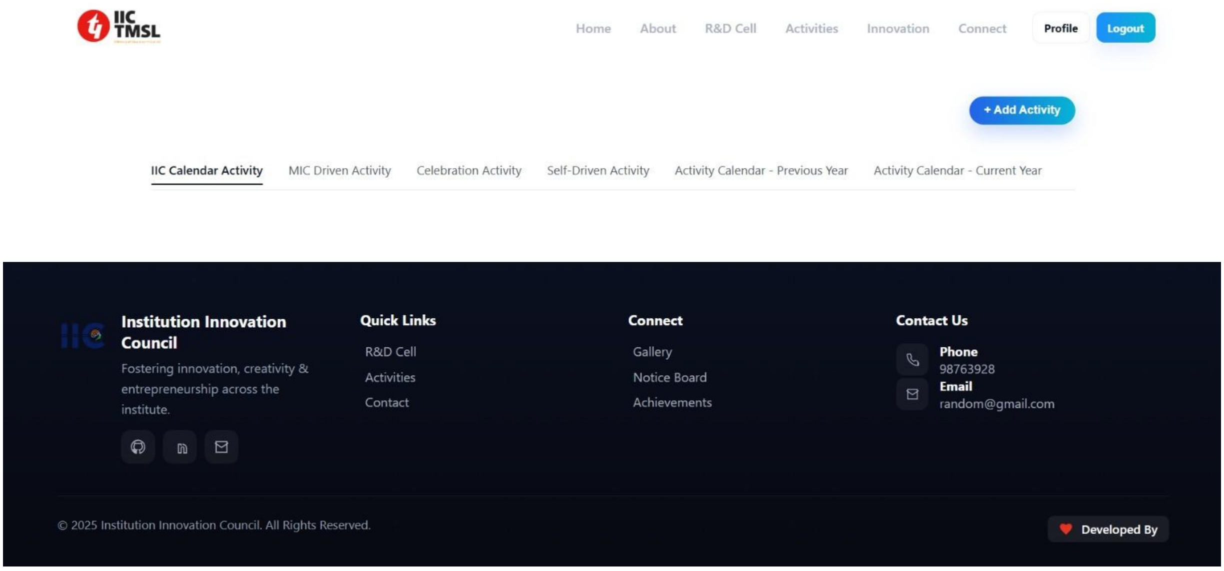
Photo:
No file chosen

Step 4: Activities & Events Display

- Activities and events were created by Admin and stored in the database.
- Event details were displayed correctly with proper date and description.

- Outdated activities were removed or archived successfully.

Result: Event and activity modules worked reliably.



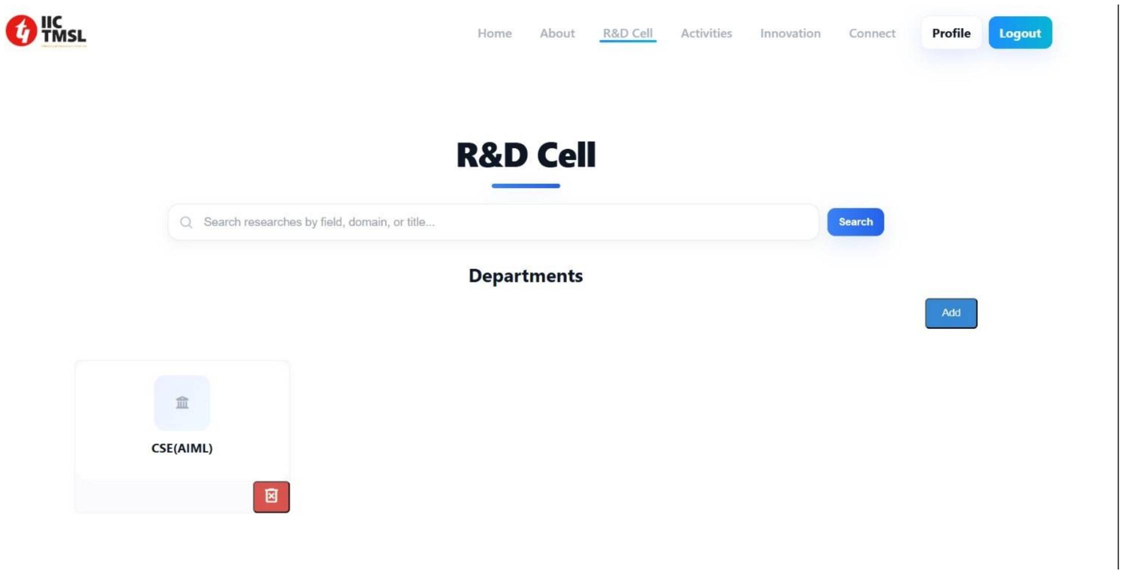
Fostering innovation, creativity & entrepreneurship across the institute.

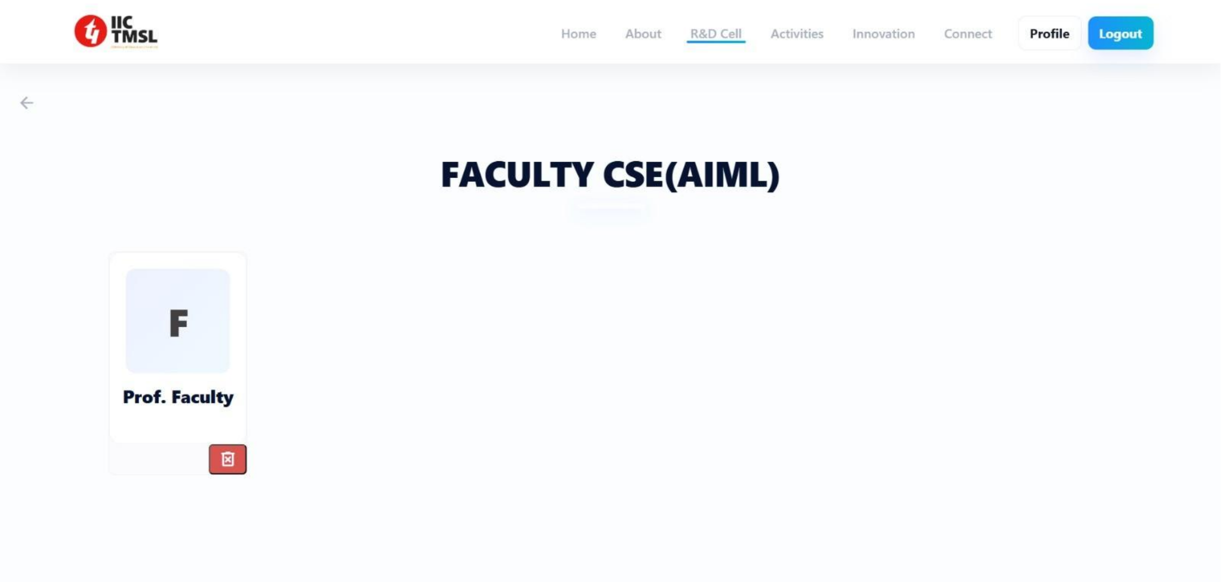
© 2025 Institution Innovation Council. All Rights Reserved.

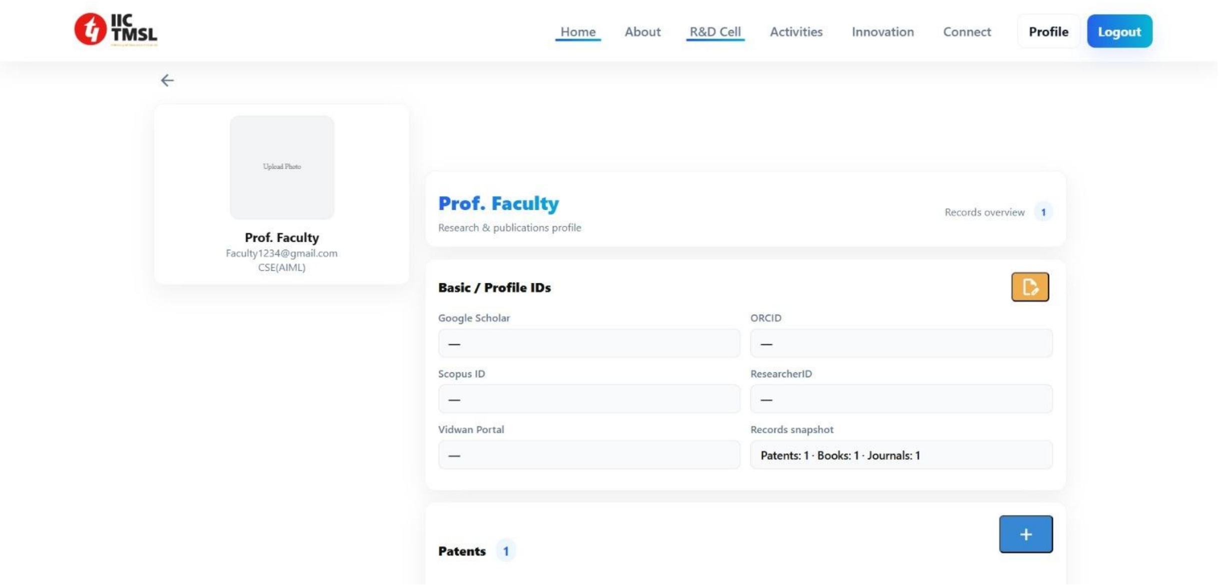
Step 5: Faculty Profile & R&D Data Analysis

- Faculty members updated profile and R&D information.
- Data was stored accurately and retrieved without duplication.
- Admin could view all faculty records in a centralized manner.

Result: Faculty profile and R&D modules performed accurately.







Prof. Faculty
Faculty1234@gmail.com
CSE(AIML)

Prof. Faculty
Research & publications profile

Records overview 1

Basic / Profile IDs

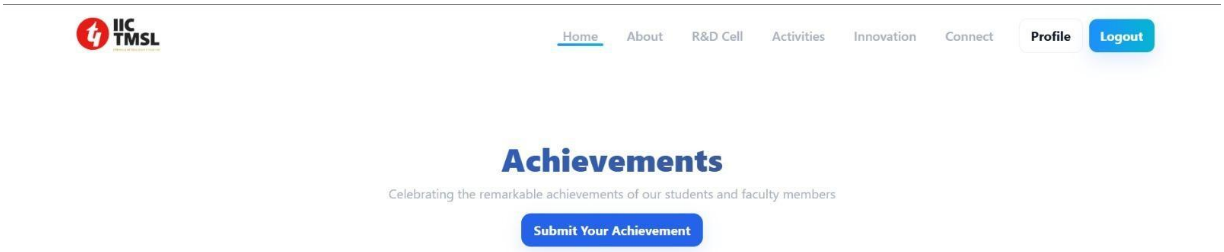
Google Scholar: —
ORCID: —
Scopus ID: —
ResearcherID: —
Vidwan Portal: —
Records snapshot: Patents: 1 · Books: 1 · Journals: 1

Patents 1

Step 6: Achievement & Certificate Approval Results

- Students and faculty submitted achievements successfully.
- Admin approval workflow worked before public display.
- Certificates added by Admin appeared correctly on the portal.

Result: Validation and approval mechanism ensured authenticity.



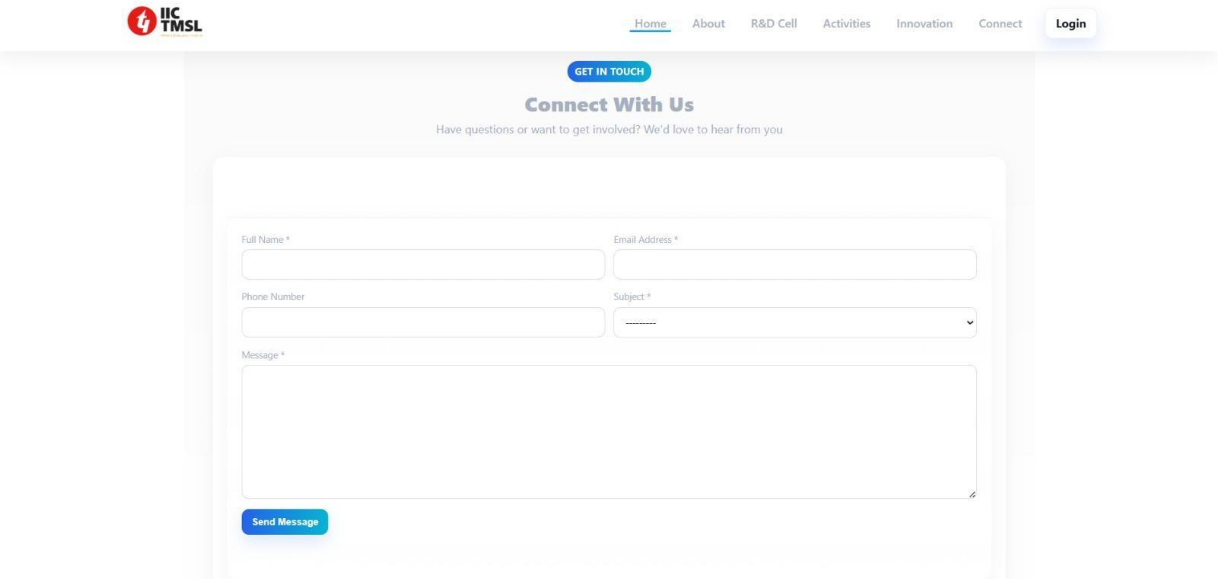
Achievements

Celebrating the remarkable achievements of our students and faculty members

Step 7: Query Submission & Resolution Results

- Students submitted queries through the portal.
- Admin and Faculty viewed and responded to queries.
- Query status changed from pending to resolved after response.

Result: Query management system improved communication efficiency.



Connect With Us

Have questions or want to get involved? We'd love to hear from you

Full Name *

Email Address *

Phone Number

Subject *

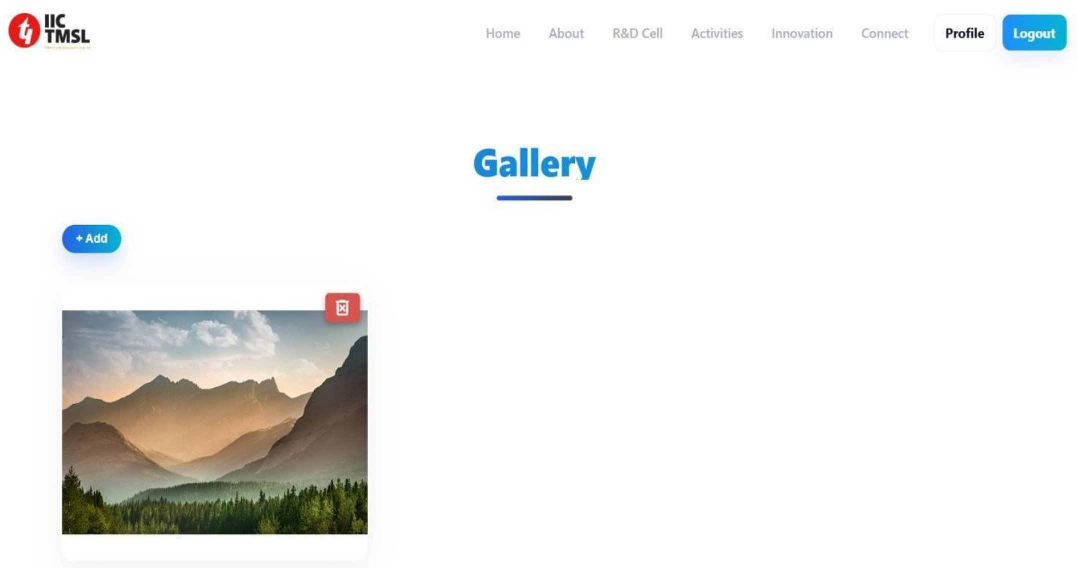
Message *

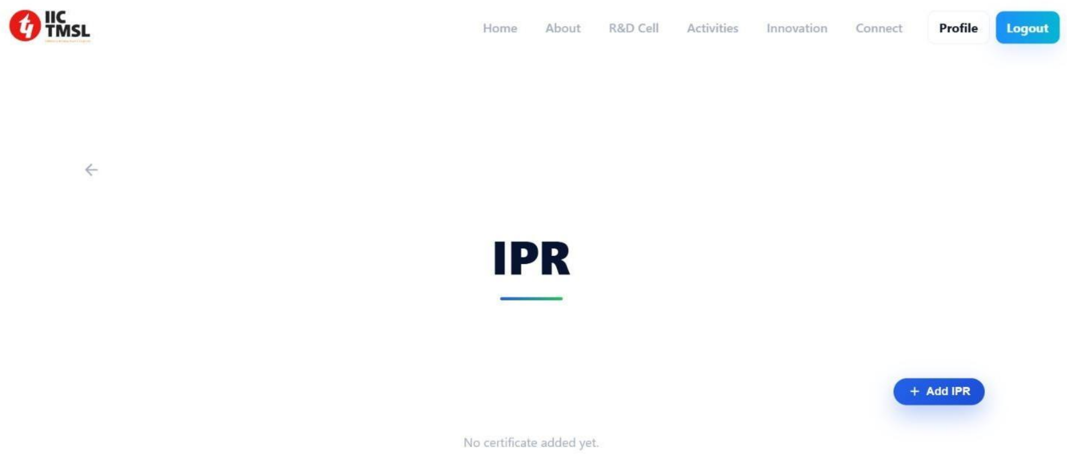
Step 8: Gallery, IPR & Incubator Output Analysis

- Admin uploaded event images to the gallery successfully.

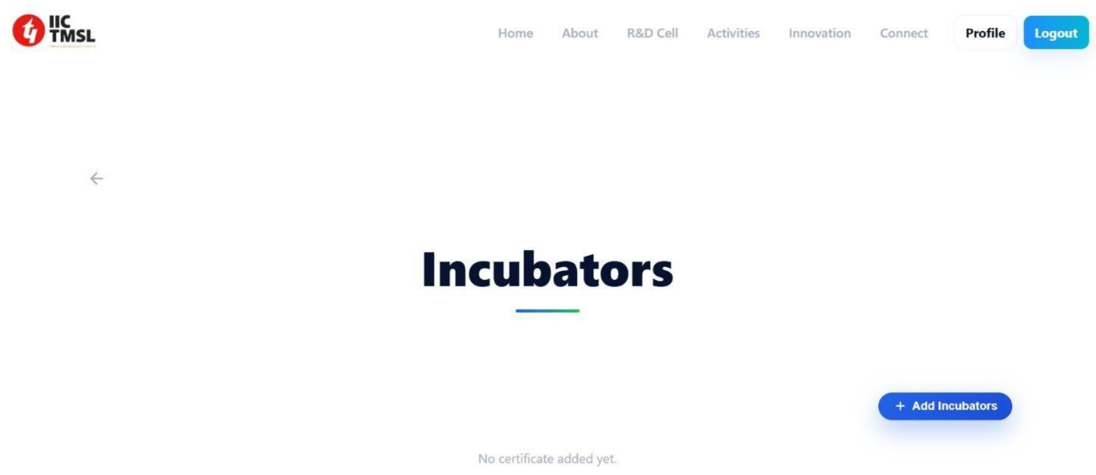
- IPR and Incubator data was updated and displayed correctly.
- Old or incorrect content was removed without affecting other data.

Result: Content management modules functioned smoothly.





No certificate added yet.



Incubators

No certificate added yet.

Step 9: Database Performance & Integrity Check

- All CRUD operations executed successfully.
- No data inconsistency or loss was observed.
- Relational links between entities remained intact.

Result: Database ensured data integrity and reliability.

Step 10: UI & Responsiveness Results

- Portal was tested on desktop and mobile devices.
- UI adapted properly to different screen sizes.
- Result: Responsive and accessible design achieved.

11. Testing

Testing in the IIC Web Portal project was performed to ensure that every module works correctly according to the defined requirements and system flow. Since the application handles multiple user roles such as Admin, Faculty, and Students, testing focused on verifying role-based access, data accuracy, and smooth interaction with the database. Each feature—login, notice board, achievements, meetings, gallery, and queries—was tested individually and in integration to ensure error-free functionality. Testing helped identify logical errors, input validation issues, and access control problems before final deployment.

11.1 Test Case

A test case represents a specific scenario used to validate a particular function of the IIC Web Portal. Each test case includes test inputs, expected results, and actual outcomes after execution. For this project, multiple test cases were designed for critical modules such as authentication, notice creation, faculty profile updates, achievement approval, and query resolution.

Test Case: 1 (Login Functionality)

This test case verifies the user authentication mechanism.

- Input: Valid Admin/Faculty login credentials
- Expected Output: Successful login and redirection to the respective dashboard
- Result: The system correctly authenticated users and prevented unauthorized access

Similarly, other test cases ensured that only Admin could approve achievements, faculty could update R&D details, and students could submit queries.

12. Cyclomatic Complexity Measures

Cyclomatic Complexity is a widely used software metric that measures the logical complexity of a program by analyzing its control flow. It represents the number of independent execution paths through a program’s source code. This metric helps developers and testers understand how complex a module is and how much effort is required to test and maintain it. A higher cyclomatic complexity indicates more decision points and increased risk of errors, whereas a lower value suggests simpler, more readable, and more maintainable code.

In the IIC Web Portal, cyclomatic complexity was considered during the design and development of core modules such as user authentication, role-based access control, achievement approval workflows, and content management operations. These modules involve conditional logic such as user role checks, validation rules, and approval status handling. Measuring and controlling cyclomatic complexity ensured that the system remained stable, easy to test, and adaptable for future enhancements.

Calculation of Cyclomatic Complexity

Cyclomatic Complexity can be calculated using the formula:

$$V(G)=E-N+2P$$

Where:

- E = Number of edges in the control flow graph
- N = Number of nodes
- P = Number of connected components (usually 1 for a single module)

Alternatively, it can be estimated by counting the number of decision points (`if`, `else`, `elif`, `for`, `while`) in a function and adding one. This value indicates how many test cases are required to achieve complete path coverage.

13. Future Scope

Although the current system meets all academic and functional requirements, several enhancements can be introduced in the future to improve scalability, usability, and functionality:

1. **Email and SMS Notification System**
   Automatic notifications for meetings, events, and approvals can be integrated to enhance real-time communication.

2. **Advanced Analytics and Reporting**
   Dashboards with graphical reports on activities, achievements, and participation can help administrators in decision-making.

3. **Cloud Deployment and Scalability**
   Migrating the system to cloud platforms can improve performance, data backup, and accessibility.

4. **Mobile Application Integration**
   A dedicated mobile app can be developed for easier access by students and faculty.

5. **Enhanced Security Features**
   Two-factor authentication and audit logs can further strengthen data security.

Conclusion

The Dynamic Institution Innovation Council (IIC) Web Portal successfully achieves its objective of providing a centralized, secure, and efficient digital platform for managing innovation-related activities within an academic institution. The project effectively replaces traditional static websites and fragmented communication methods with a dynamic, database-driven system that supports real-time updates, structured workflows, and improved accessibility for students, faculty members, and administrators.

Through the implementation of role-based access control, the system ensures that different users—Admin, Faculty, and Students—can interact with the portal according to their responsibilities and permissions. Key modules such as notice management, event and meeting scheduling, faculty profile and R&D management, achievement approval, certificate handling, query resolution, and gallery management were successfully designed, implemented, and tested. These modules work in coordination with a centralized database, ensuring data consistency, integrity, and easy retrieval of institutional records.

The project also demonstrates the effective application of core software engineering principles, including requirement analysis, system design, modular development, database management, coding standards, testing, and quality assurance. Metrics such as cyclomatic complexity were considered to maintain code simplicity and improve maintainability. As a result, the system remains stable, scalable, and easy to enhance in the future.

Overall, the IIC Web Portal improves institutional communication, reduces manual workload, enhances transparency, and supports the promotion of innovation and entrepreneurship within the campus. The project provides a strong foundation for future digital transformation initiatives and serves as a practical, real-world application of modern web development and software engineering concepts.

REFERENCES

The following references were used during the analysis, design, and development of the Dynamic Institution Innovation Council (IIC) website and were helpful in comprehending digital transformation in higher education, role-based web systems, and contemporary web development techniques.

1. Ministry of Education's Innovation Cell (MIC). (2023). *Institution's Innovation Council (IIC) – Official Guidelines*. Retrieved from https://www.mic.gov.in/iic

2. Sharma, A., & Bansal, P. (2021). "Design and Development of Dynamic Web Portals Using Django Framework." *International Journal of Computer Applications*, 182(45), 10–15.

3. Kumar, M., Gupta, S., & Verma, R. (2020). "Digital Transformation in Higher Education Institutions: A Framework for Enhancing Campus Information Systems." *IEEE Access*, 8, 123456–123465.

4. Das, S., Roy, M., & Sen, A. (2020). "Role-Based Access Control Models for Secure Web Applications." *Journal of Information Security and Applications*, 54, 102–112.

5. Patel, D., & Mahato, N. (2019). "A Comparative Study of Static vs Dynamic Websites in Academic Institutions." *International Journal of Advanced Research in Computer Science*, 10(4), 45–50.

6. Django Software Foundation. (n.d.). *Django Documentation*. Retrieved from https://docs.djangoproject.com

7. Python Software Foundation. (n.d.). *Python Official Documentation*. Retrieved from https://www.python.org/doc/

8. W3Schools. (n.d.). *HTML, CSS, and JavaScript Tutorials*. Retrieved from https://www.w3schools.com

9. SQLite Consortium. (n.d.). *SQLite Documentation*. Retrieved from https://www.sqlite.org/docs.html

10. GitHub. (n.d.). *Version Control and Collaboration Documentation*. Retrieved from https://docs.github.com

Final Document

Techno India University, West Bengal

Document Details

Submission ID

trn:oid:::13909:124569741

Submission Date

Dec 15, 2025, 11:17 PM GMT+5:30

Download Date

Dec 15, 2025, 11:19 PM GMT+5:30

File Name

Final Document_Harsankit.pdf

File Size

8.2 MB

53 Pages

7,544 Words

48,241 Characters

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Cited Text

Match Groups

45 Not Cited or Quoted 8%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
1%	Publications
6%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.