Scholara: A System for the Asynchronous Management of Academic Artifacts Featuring Federated Authentication with a RESTful Endpoint Architecture

Praveen BV

B.E Computer Science and Engineering (Cyber Security), Saveetha Engineering College, Chennai bvpraveenvignesh2005@gmail.com

Sasirajkumar T J

B.TECH Artificial Intelligence and Data Science,Saveetha Engineering College, Chennai tjsasirajkumar@gmail.com

Jegadeesh S

B.TECH Artificial Intelligence and Data Science,Saveetha Engineering College, Chennai jegadeesh942@gmail.com

*Abstract*— Establishing genuine collaboration and visibility between students and teachers is a problem that academic institutions frequently encounter. This demand is met by Scholara, which offers a single, web-based platform made to close the distance between different groups and simplify academic networking. In addition to having user-friendly dashboards, comprehensive profile management, project and publication showcases, appointment scheduling, and a dynamic scoreboard for peer discovery, the platform has two user interfaces designed specifically for academics and students. Users can collaborate on projects using Scholara's collaboration tools, while AI-driven analytics offer performance insights and support for upcoming services like automatic resume production. Scholara, which was developed with an emphasis on usability and contemporary online technologies, makes academic accomplishments more visible, encourages effective communication, and provides data-driven insights to aid in career growth. Because of its all-encompassing strategy, the platform is positioned as a useful instrument for educational institutions looking to enhance community awareness, collaboration, and connectedness.

Keywords—Academic collaboration, Web-based platform, Student–faculty interaction, Educational technology, User dashboards, AI-driven analytics, Appointment scheduling, Performance scoring, Leaderboard system, Profile management.

# Introduction

Collaboration and effective communication are essential drivers of academic achievement and professional growth within educational institutions. Yet, students and faculty often face fragmented systems for managing profiles, showcasing achievements, and coordinating projects, which can hinder networking, mentorship, and recognition. This lack of integration limits opportunities for interdisciplinary collaboration, slows career development, and reduces the visibility of individual and collective accomplishments. Existing solutions typically address only isolated aspects of academic life, such as portfolio management or appointment scheduling, without providing a unified, user-friendly experience. To address these challenges, this paper presents Scholara—a web-based platform designed to bridge the gap between students and faculty by streamlining academic networking, project management, and achievement tracking. Scholara’s primary goal is to offer a centralized environment where users can create detailed profiles, manage and showcase projects and publications, schedule appointments, and collaborate seamlessly. The platform features dual dashboards tailored for students and faculty, comprehensive profile management, publication and project showcases, appointment scheduling, a dynamic leaderboard for peer discovery, and AI-driven analytics for performance evaluation. By providing a holistic, data-driven approach to academic collaboration, Scholara enhances connectivity, increases the visibility of achievements, and supports career development for both students and faculty.

# Related Work (Background and Motivation)

## Academic Networking Platforms

Academic networking sites such as LinkedIn, ResearchGate, and Academia.edu enable users to build professional profiles, connect with peers, and share publications. While these platforms support networking and visibility, they are not tailored to the specific workflows or collaboration needs of academic institutions.

## Learning Management and Collaboration Tools

For project management and course administration, systems like Canvas, Blackboard, Google Classroom, and Microsoft Teams are widely adopted. These tools offer functionalities such as assignment tracking, resource sharing, and communication, but often lack robust features for achievement tracking, peer discovery, or institution-wide analytics.

## Appointment Scheduling and Meeting Coordination

Appointment scheduling and meeting coordination are typically handled by standalone tools, further fragmenting the user experience.

## AI and Social Engagement in Academic Platforms

Recent research and technological advancements have explored the integration of artificial intelligence into academic platforms, enabling features such as automated performance analytics, personalized dashboards, and AI-driven resume generation. However, most existing solutions implement these features in isolation, without providing a unified, role-specific experience for both students and faculty.

## Gaps and Motivation for Scholara

These deficiencies highlight the need for a holistic, web-based platform that unifies academic networking, project management, achievement tracking, and collaboration within a single, user-centered environment. By integrating dashboards, leaderboards, collaboration tools, and appointment scheduling into a cohesive platform, Scholara aims to advance the state of academic collaboration and achievement tracking, providing a scalable and adaptable solution for contemporary educational institutions.

# System Architecture (Overall Design)

## Architectural Overview

The platform follows a multi‑tier web architecture composed of: (1) a client layer (browser) rendering a role‑aware single page application, (2) an application layer exposing REST endpoints, (3) a persistence layer (document database), and (4) auxiliary service integrations for authentication, file/object storage, notifications, and planned AI analytics. Core runtime interaction: the client acquires an authentication token, invokes protected endpoints, receives normalized JSON payloads, and renders role‑specific dashboards.

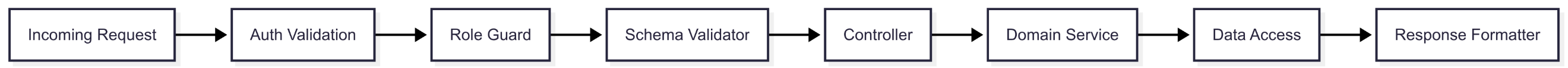
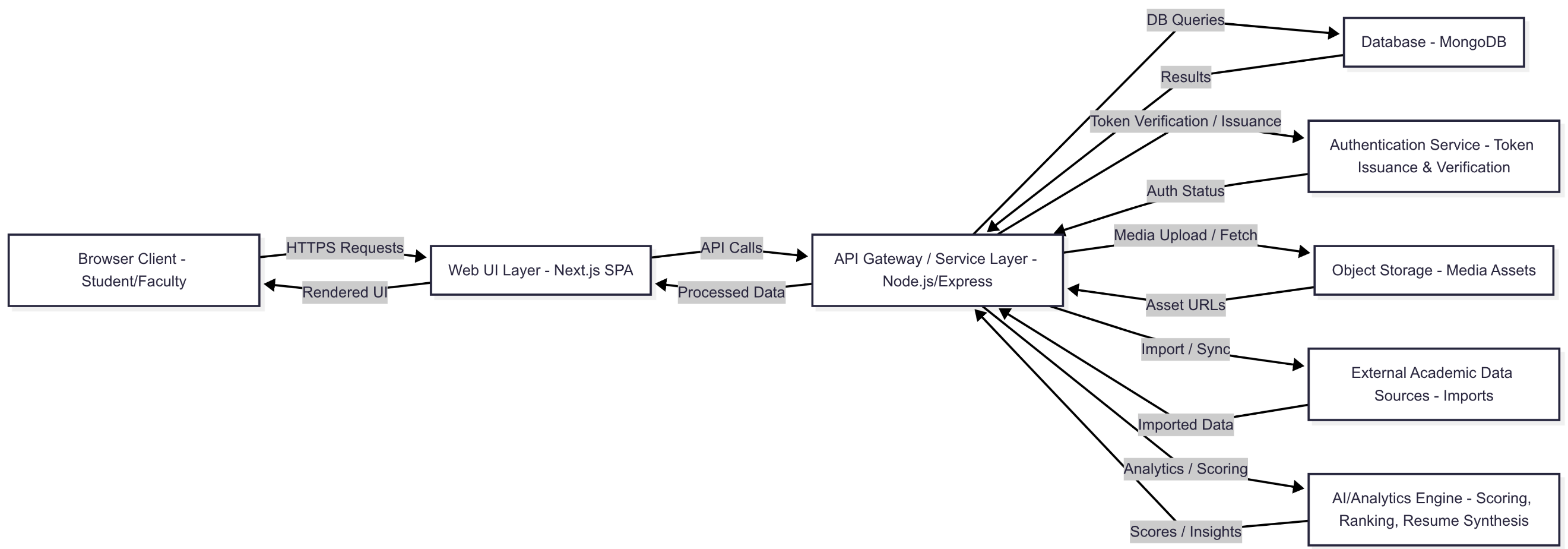


Figure 1. High-Level Architecture Diagram of Scholara



## Frontend Design and User Interface

* Role‑aware routing (student vs. faculty) with guarded dashboard entry points.
* Modular feature panels: Projects, Publications, Achievements, Courses & Events, Appointments, Leaderboard, Profile, and (future) Analytics Insights.
* Reusable visual atoms (cards, lists, modal dialogs, crop & upload components) ensuring consistent spacing, typography, and status feedback (loading, empty states, errors).



Figure 2. UI Layer Component Map of Scholara Dashboard

## Backend and API Services

* The service layer exposes domain‑grouped REST endpoints: Users, Projects, Publications, Achievements, Courses/Events, Appointments, Leaderboard/Analytics.
* A middleware pipeline enforces (a) authentication token validation, (b) role/permission authorization, (c) request schema validation, (d) rate limiting (planned), and (e) structured error translation. Business logic modules execute aggregation (e.g., computing composite activity scores) to reduce frontend overfetch. Responses adopt consistent envelopes (status, data, optional pagination/meta).

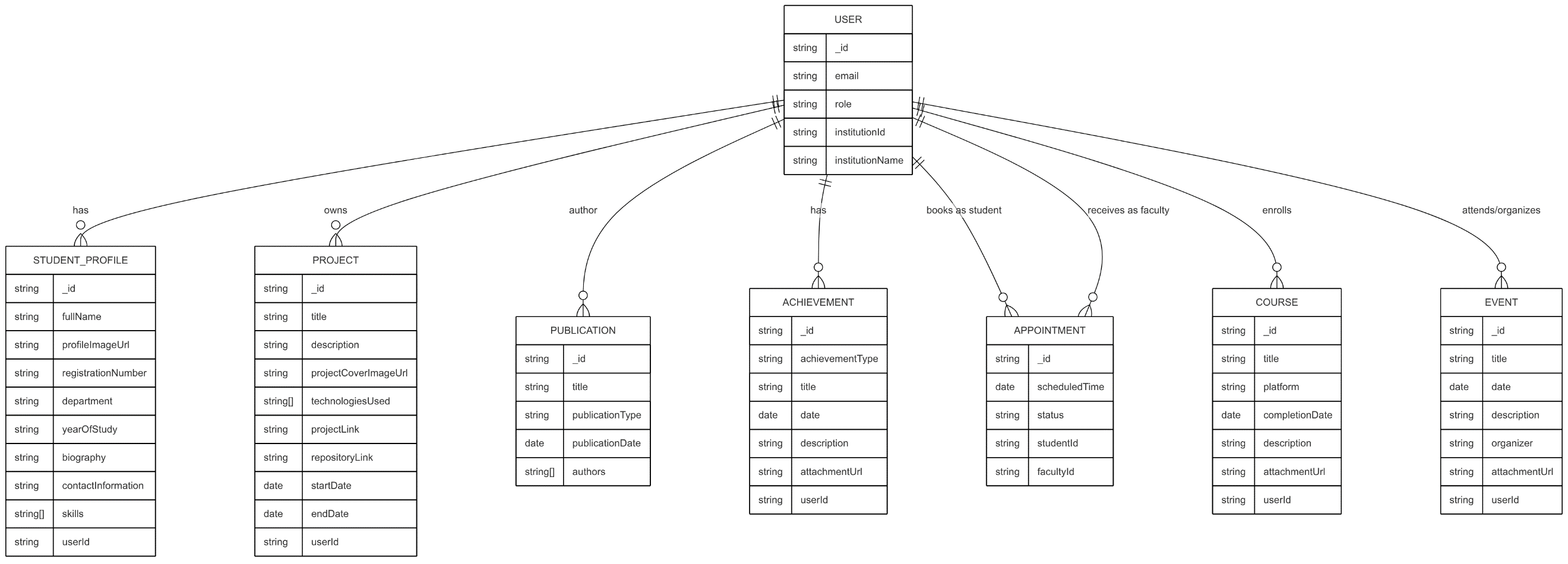
Figure 3. Service Layer & Middleware Flow in Scholara Backend

## Database Structure

A document‑oriented model stores:

* Users: role, institution context, profile attributes (skills, interests, biography), linkage to activity artifacts.
* Projects: owners, collaborators, status, temporal range, technologies, cover media reference, description.
* Publications: authors (user references), bibliographic metadata, optional cover media.
* Achievements: category, issued date, evidence artifact reference.
* Courses / Events / Competitions: enrollment or participation metadata, attachments, completion indicators.
* Appointments: participant references, scheduled slot, status lifecycle (requested, confirmed, completed, canceled), optional notes.
* Analytics Snapshots (planned): timestamped scoring vectors (activity density, recency indices, publication impact proxy, collaboration factor).

Figure 4. Logical Data Model Diagram for Scholara



## Authentication and Security

* Authentication: email/password (or federated) credential issuance; short‑lived ID tokens transferred with each request. Server validates tokens, resolves user record, augments request context with role and institution.
* Authorization: role + ownership checks (e.g., only owner can mutate personal project; faculty may approve appointment requests addressed to them).
* Data minimization: selective field projection; sensitive internal flags excluded from standard payloads.
* Input validation: strict schema enforcement prevents malformed or oversized payloads.

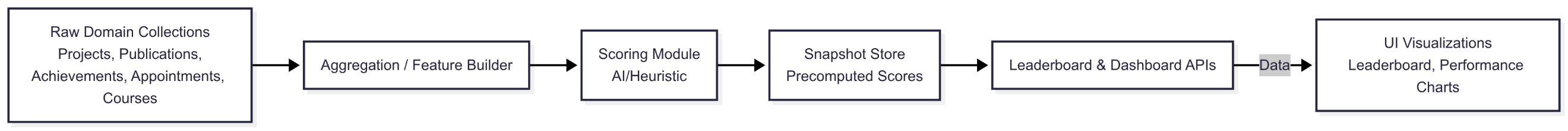
Figure 5. Auth & Authorization Sequence in Scholara



## AI Analytics and Advanced Features

* Current heuristic pipeline (phase 1): derive composite activity scores from weighted counts (projects, publications, achievements, recentness) plus collaboration breadth (unique co‑contributors).
* Planned evolution (phase 2): incorporate temporal decay, normalized percentile ranks per institution cohort, and context‑aware text summarization for automated resume sections.
* Future ML pipeline: feature extraction (artifact counts, recency histograms, diversity metrics), model inference (engagement level classification or projection of potential collaboration interest), caching of computed features for responsive dashboards.

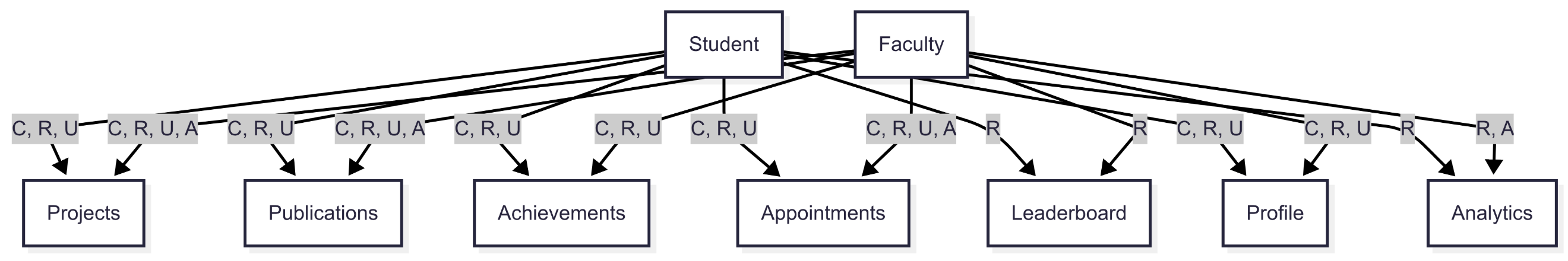
Figure 6. Analytics Data Flow in Scholara



## Role‑Based User Experience

* Navigation: Students emphasize discovery (projects, appointments, leaderboard); Faculty emphasize supervision (appointments management, publication curation).
* Permissions: Creation/edit rights constrained by role; cross‑role collaborative proposals require dual confirmation.
* Data Visibility: Some faculty metrics abstracted to students (e.g., availability windows without exposing private scheduling notes).
* UI Emphasis: Students receive motivational progress indicators; Faculty receive workload and engagement summaries.
* Each feature (Projects, Publications, Achievements, Appointments, Leaderboard, Profile, Analytics) lists the permissions for Students (S) and Faculty (F):
  + C = Create
  + R = Read
  + U = Update
  + A = Administrative

Figure 7. Role Context Matrix for Scholara Features and Permissions



## Extensibility and Scalability

* Scalability strategies: stateless API instances (horizontal scaling), externalized session/token handling, optional in‑memory or distributed cache layer for leaderboard and profile aggregates, deferred background jobs (future message queue) for heavy analytics recomputation or media processing.
* Extensibility: modular domain packages allow plug‑in addition (e.g., Administrative Governance, Institution Onboarding, Advanced Recommendation Engine) without refactoring existing endpoints. Schema flexibility supports introducing new artifact categories (e.g., Grants, Patents) with minimal downstream impact.

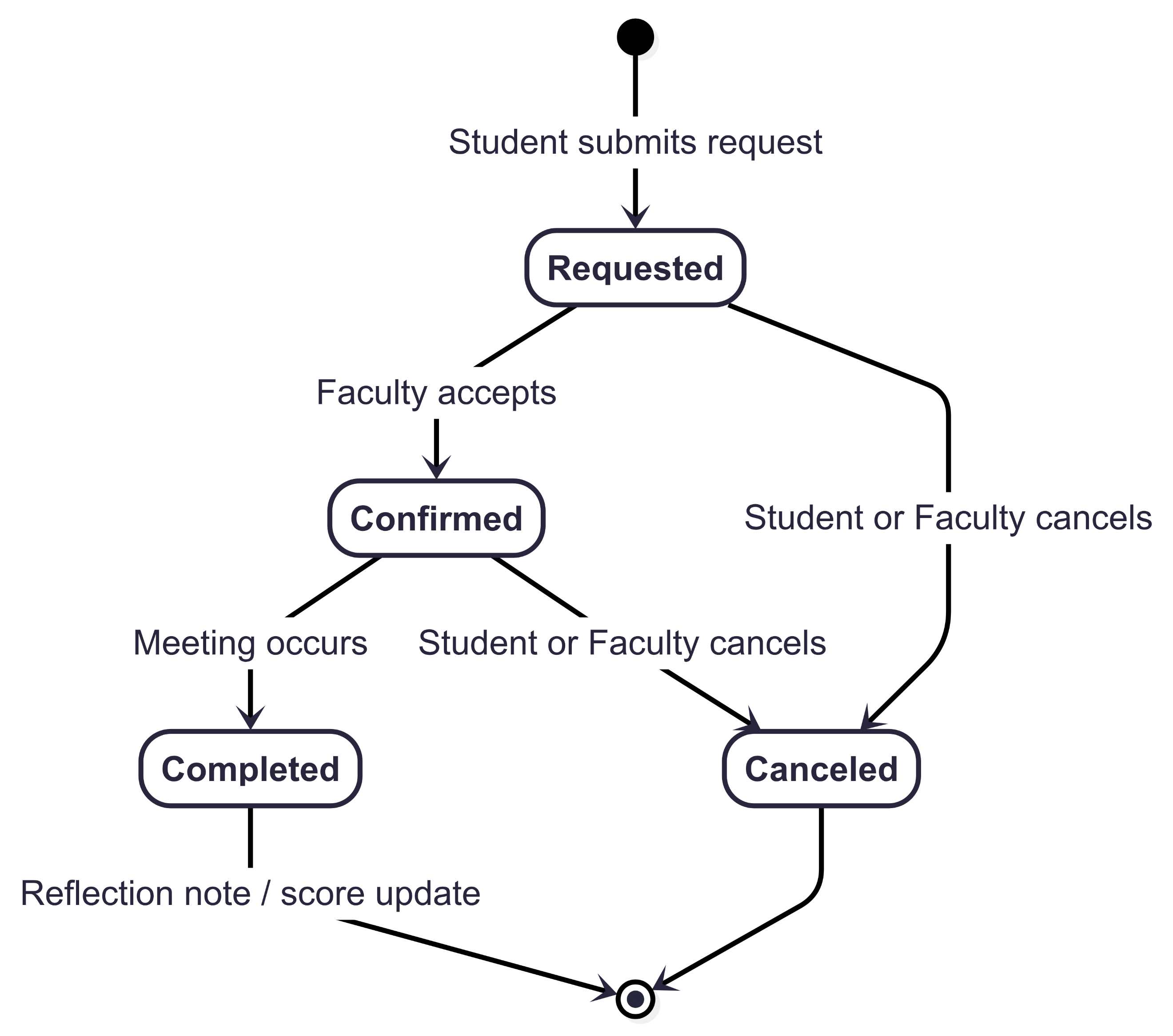
## Security and Compliance (Brief)

* Foundational protections: encrypted transport (TLS), encrypted storage layers, least‑privilege access patterns, controlled media upload pipeline (size/type validation + scanning stage, if enabled), consistent redaction of personally sensitive optional fields in aggregated analytics.
* Compliance roadmap: data retention policies, export & erasure workflows, consent acknowledgment records, and audit event immutability for regulated environments..

## Representative Workflow Example (Appointment Request)

* Student locates faculty profile and selects an available slot.
* Client submits appointment request (token attached).
* API validates payload, confirms slot availability, persists pending appointment.

Figure 8. Appointment Lifecycle State Diagram in Scholara



# User Flow

Scholara's user experience is designed around intuitive, role-based workflows that minimize friction from registration through advanced academic collaboration. The platform accommodates distinct user journeys for students and faculty while maintaining consistent design patterns and navigation principles across all features.

## Onboarding and Registration Process

* New users begin their journey on the platform's landing page, where they click the "Sign Up" button to access the registration form.
* During registration, users select their primary role—student or faculty—and choose their institution from a pre-populated dropdown list.
* After entering their email and password credentials, the system creates a Firebase authentication record and simultaneously stores user metadata in the MongoDB database.
* The platform immediately triggers an email verification process, redirecting users to a verification message page where they are prompted to check their inbox and complete email confirmation before proceeding.

## Authentication and Dashboard Access

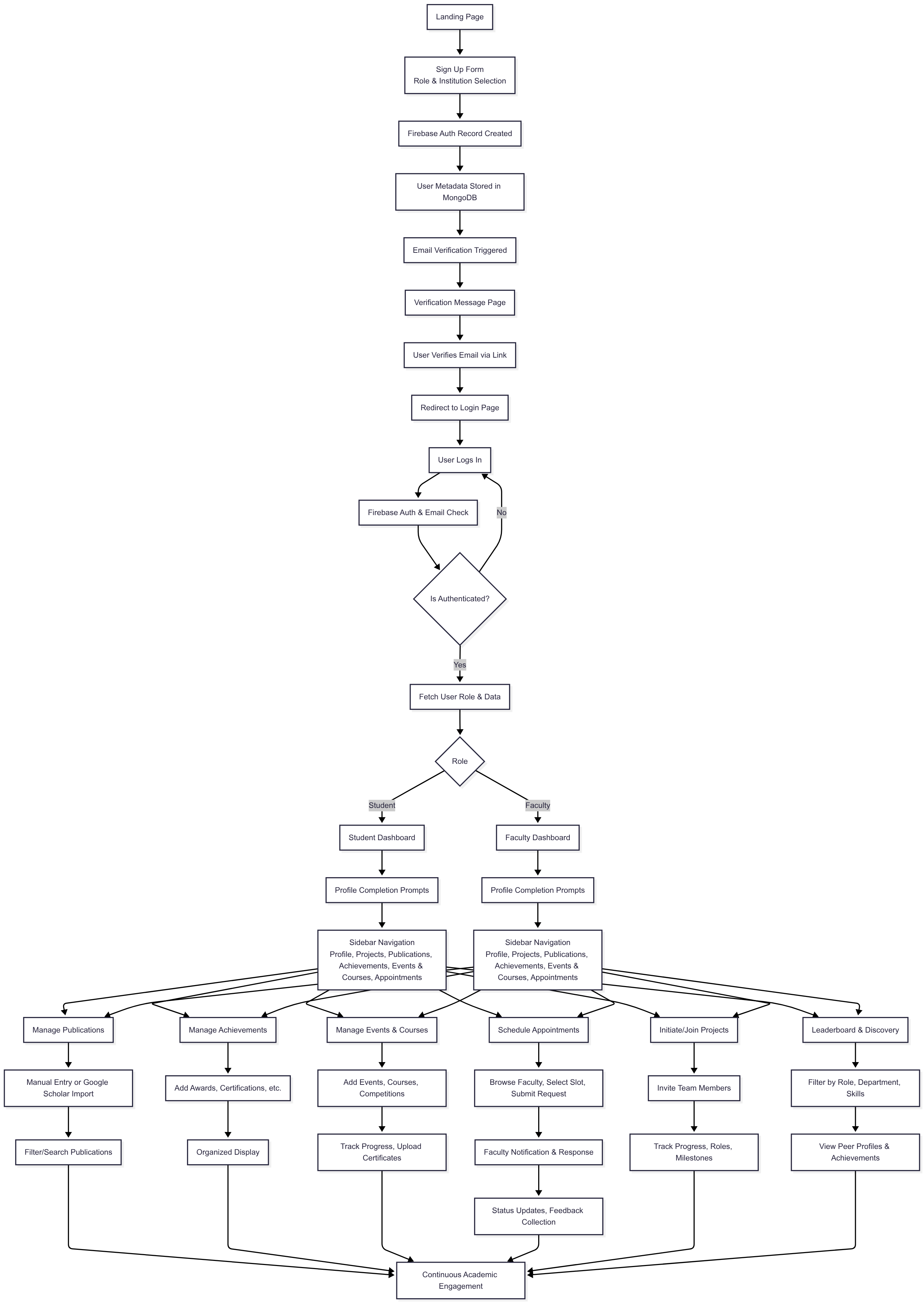
* Upon successful email verification, users are directed to the login page where they enter their credentials.
* The system validates authentication through Firebase and performs an email verification check.
* Once authenticated, users are automatically redirected to their role-specific dashboard—students to the Student Dashboard and faculty to the Faculty Dashboard.

## Profile Setup and Core Navigation

* The initial dashboard experience guides users through profile completion with contextual prompts for missing information.
* Students input academic details including registration number, department, year of study, skills, and research interests, while faculty provide information about their position, expertise areas, office hours, and departmental affiliations.

## Publications Management Workflow

* Users navigate to the Publications page where they can view, add, and manage their academic publications.
* The interface supports multiple publication types including journal papers, conference papers, patents, books, and technical reports.
* Students and faculty can manually input publication details or utilize the one-click Google Scholar import feature for automated data population.



## Achievements Tracking Workflow

* The Achievements page allows users to document and showcase various accomplishments including awards, certifications, hackathons, competitions, and scholarships.
* Users can add new achievements through a structured form that captures details such as achievement type, issuing organization, date received, and supporting documentation.

## Events & Courses Management

* Users access the Events & Courses section to manage their participation in academic events, online courses, and competitions.
* The interface is subdivided into specific categories: Events, Online Courses, and Competitions.

## Appointment Scheduling Workflow

* The appointment system exemplifies Scholara's streamlined interaction design.
* Students browse faculty profiles, view available time slots, and submit appointment requests with optional context about meeting objectives.
* Faculty receive notifications and can accept, reschedule, or decline requests through their dashboard.

## Project Collaboration Workflow

* Collaborative project initiation follows an intuitive invitation-based model.
* Users create project descriptions, define collaboration requirements, and invite team members through the platform's search and discovery features.
* Invited users receive notifications and can accept or decline participation.

## Leaderboard and Discovery Features

* Users can access the leaderboard system to discover peers, view rankings based on composite performance scores, and explore detailed profiles of other students and faculty. Users can view peer accomplishments across all categories—projects, publications, achievements, and courses—fostering healthy competition and academic engagement.

Figure 9. Comprehensive User Flow Diagram for Scholara

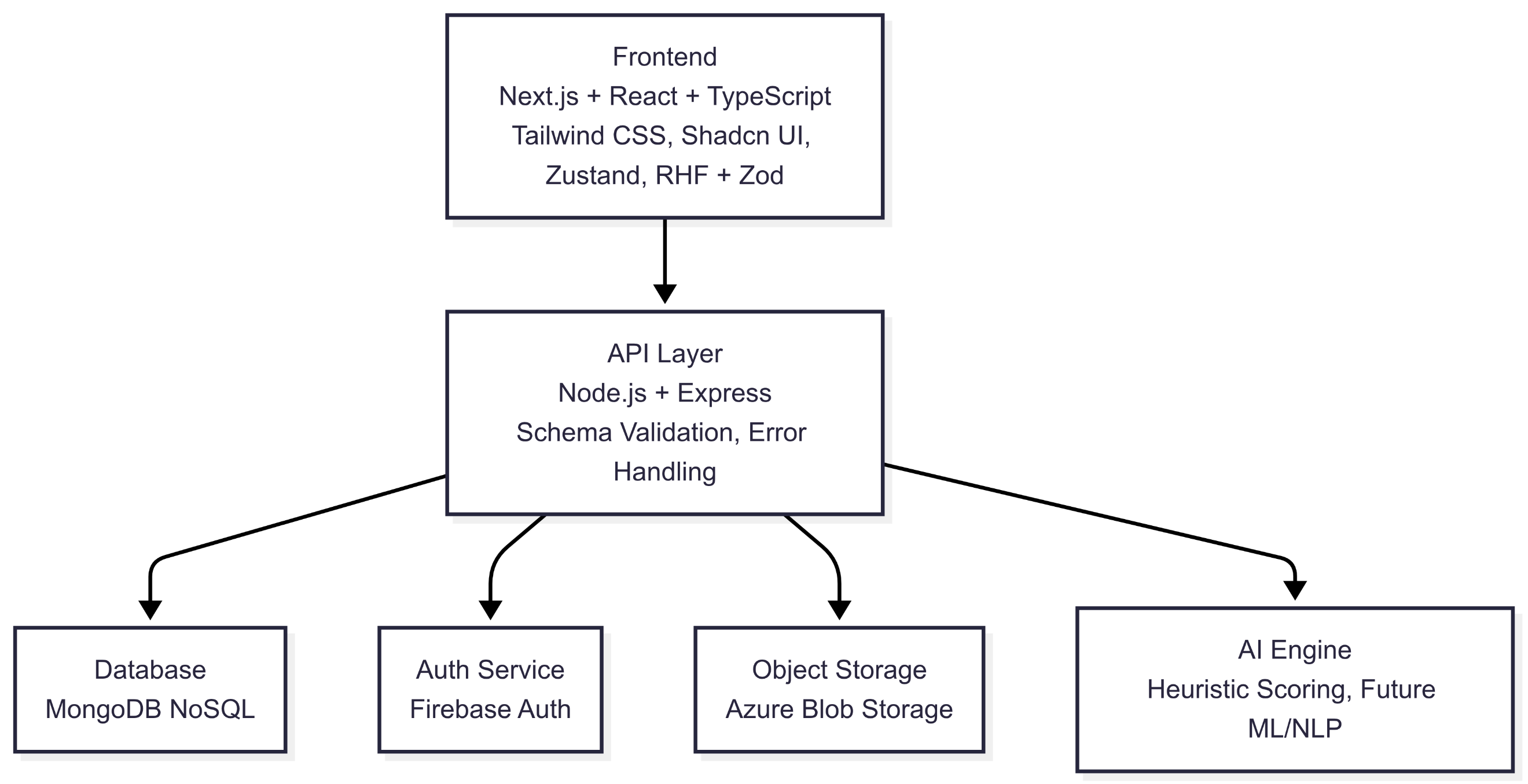
* This user-centered design ensures that both students and faculty can efficiently navigate from initial registration to advanced academic collaboration while maintaining comprehensive records of their academic achievements, publications, courses, and collaborative activities, demonstrating Scholara's commitment to reducing complexity while enabling rich academic networking and holistic portfolio management capabilities.

# Implementation Details

## Technologies Used

The platform employs a TypeScript-based full‑stack web architecture. The frontend is built with a React/Next.js framework using a component-driven design system, utility‑first styling, form validation libraries, and lightweight state management for global session and role context. A document database (NoSQL) supports flexible evolution of academic entities (projects, publications, achievements, events, courses, appointments). Cloud object storage is used for media (attachments, cover images). Authentication is delegated to a managed identity provider (email/password plus verification). Planned AI modules use heuristic scoring now with a pathway to integrate machine learning and natural language generation.

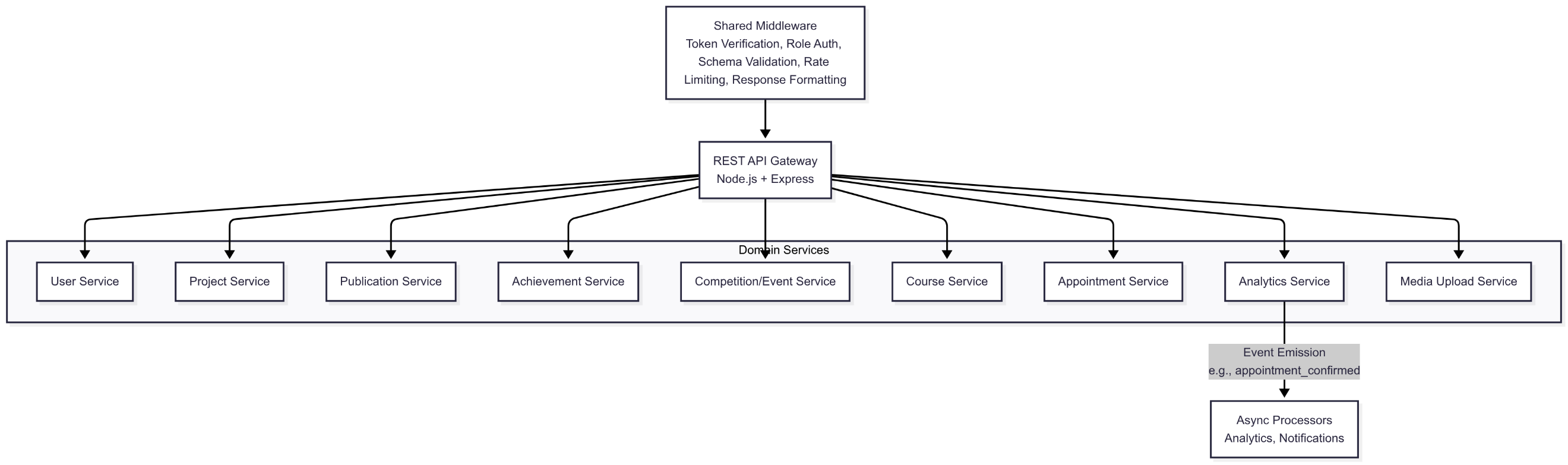
Figure 10. Technology Stack Overview for Scholara



## Backend Design

The backend follows a modular monolith pattern: domain groupings (users, projects, publications, achievements, competitions/events, courses, appointments, analytics) expose REST endpoints under versioned routes. Shared middleware handles token verification, role authorization, schema validation, rate limiting (extensible), and uniform response formatting. Services encapsulate domain logic (e.g., project lifecycle, achievement categorization, publication type normalization). Media handling routes stream uploads directly to cloud storage with error rollback.

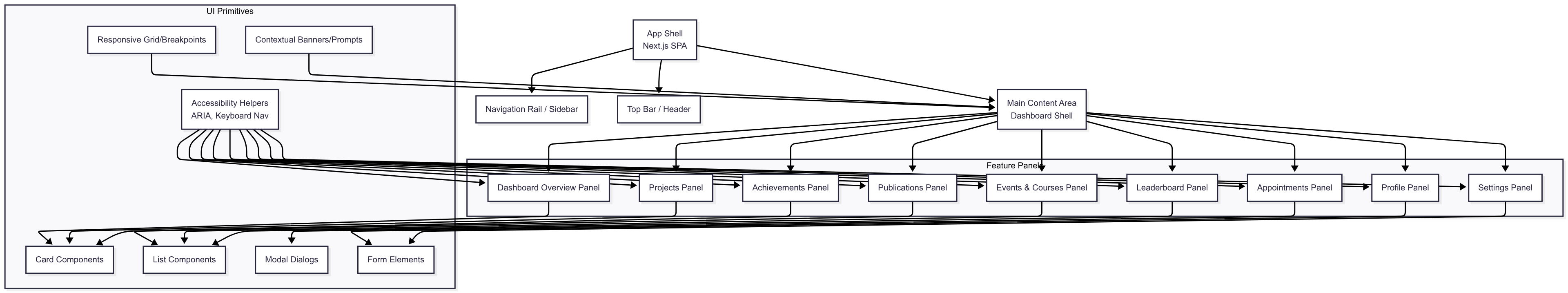
Figure 11. Backend Module Diagram for Scholara



## Frontend Design

The client renders a role-aware single-page shell with server-assisted rendering for initial load performance and client hydration for interactive modules. A persistent navigation rail provides access to dashboards, profile, projects, publications, achievements, events/courses, appointments, leaderboard, and settings. Conditional guards check authentication state and role before granting route entry. Responsive design is achieved through fluid grid utilities and adaptive component breakpoints. Contextual banners and inline prompts guide incomplete profile fields or pending verification tasks.

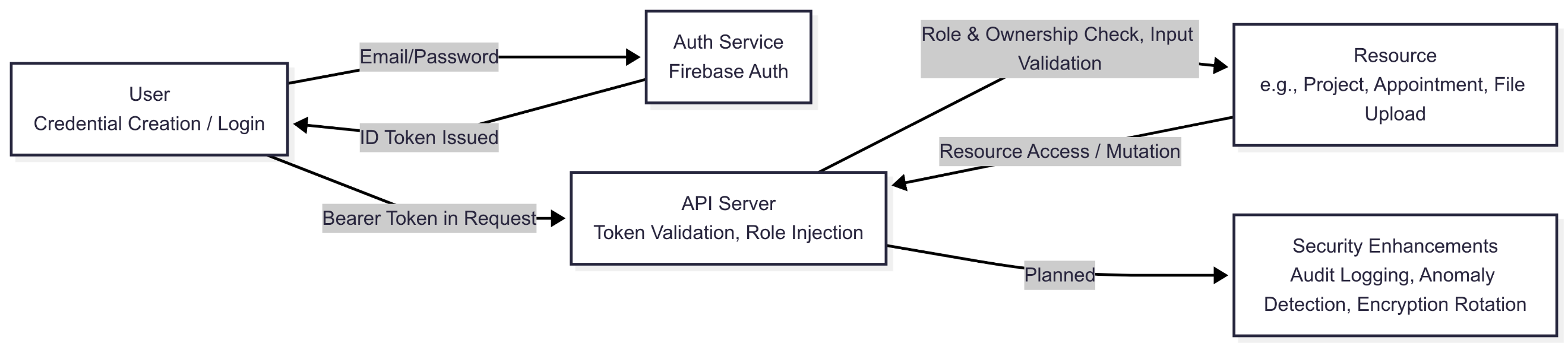
Figure 12. UI Component Hierarchy for Scholara Frontend



## Authentication & Security

Credential creation and email verification occur through a managed auth provider issuing short‑lived ID tokens. Each protected request includes a bearer token validated server-side; role and user identifier are injected into the request context. Role-based access control enforces ownership (e.g., only the creator updates a project) and cross-role constraints (e.g., appointment creation by students; approval by faculty). Input validation prevents malformed payloads and enforces length/type constraints. Media uploads are scanned (planned) and restricted by MIME type/size. Planned enhancements include audit logging for sensitive mutations, anomaly detection for suspicious access bursts, and rotating encryption keys.

Figure 13. Auth Request Sequence in Scholara



## Database Design

A document-oriented model supports collections for users, projects, publications, achievements, competitions/events, courses, appointments, and analytics snapshots. References (object IDs) relate users to artifacts (e.g., collaborators, authors). Indexes target frequent filters: role, institution, skill tags, project status, chronological fields, and composite score ordering for leaderboard queries. Flexible schemas permit adding new achievement types or publication categories without disruptive migrations.

## Deployment & Scalability

Deployment uses containerized services (API + frontend) behind a load balancer. Horizontal scaling is facilitated by stateless API nodes (token-based auth). Future scale considerations include adding a distributed cache layer for leaderboard and frequently accessed profile aggregates, message queuing for asynchronous analytics recalculations, and auto-scaling policies based on request latency and CPU utilization. Observability (planned) includes centralized logs, metrics (p95 latency, error rates), and health endpoints for orchestration probes.

Figure 14. Deployment Topology for Scholara



## Development Practices

Version control follows trunk or protected main with feature branches and pull request reviews. Linting and type checking enforce consistency pre-merge. Unit tests target domain logic (validation, scoring, status transitions); integration tests exercise REST endpoints with mocked auth tokens and in-memory database instances. Structured logging standardizes error diagnostics. A staged environment (development → staging → production) validates database indexes and migration scripts (when added) before promotion. Documentation includes architectural overviews, user flows, and domain model descriptions to onboard contributors efficiently.

# Evaluation and Results

## Evaluation Methodology and Framework

The evaluation approach focuses on multiple assessment dimensions including system responsiveness, feature completeness, user workflow efficiency, and scalability projections. User experience evaluation will assess task completion rates, interface intuitiveness, and feature adoption patterns across both student and faculty user groups. The methodology incorporates both quantitative metrics (response times, throughput, error rates) and qualitative indicators (usability satisfaction, workflow efficiency gains).

## Backend Design

Response Time Projections:

* User authentication and dashboard loading: Expected median response of 600-800ms with 95th percentile under 1.2 seconds.
* Project and achievement creation workflows: Anticipated completion times of 700-900ms for standard operations.

Throughput and Scalability Expectations:

* Concurrent user support: Designed to handle 100-150 simultaneous users per instance with acceptable performance degradation.
* API request capacity: Expected sustainable throughput of 120-150 requests per second under normal usage patterns.
* Database query optimization: Projected efficiency gains through strategic indexing on user roles, institution affiliations, and temporal fields.

|  |  |  |
| --- | --- | --- |
| **Metric** | **Expected Value** | **Notes** |
| Dashboard Load (Median) | 600–800 ms | 95th percentile < 1.2 s |
| Project Creation | 700–900 ms | Standard operation |
| Publication Entry | 800–1000 ms | With metadata validation |
| Leaderboard Query | 500–700 ms | With filtering |
| File Upload | 1.5–3 s | Depends on file size |
| Concurrent Users Supported | 100–150 | Per API instance |
| API Throughput | 120–150 req/sec | Under normal usage |

Table 1. Expected System Performance Metrics

## User Experience and Workflow Efficiency Projections

Task Completion Expectations:

* Account registration and profile setup: Anticipated completion rate above 90% with average time under 5 minutes.
* Achievement and project documentation: Expected success rate of 85-95% with streamlined form interfaces.

Feature Adoption Projections:

* Profile completeness: Anticipated 70-80% completion rate within first month of usage.
* Leaderboard engagement: Expected regular usage by 60-75% of active users for peer discovery.

## Data Quality and Integration Effectiveness

Content Management Expectations:

* Google Scholar import accuracy: Anticipated 90-95% successful metadata extraction for publication entries
* File upload reliability: Expected 98%+ success rate for attachment and cover image operations.
* Data validation effectiveness: Projected 95%+ accuracy in preventing incomplete or malformed submissions.

## Security and Compliance Performance

Authentication and Authorization:

* Token validation efficiency: Expected sub-50ms processing time for Firebase token verification.
* Role-based access control: Anticipated 100% accuracy in permission enforcement across all protected endpoints.

## Scalability and Growth Projections

Infrastructure Scaling Expectations:

* Horizontal scaling capability: Designed for seamless addition of API instances to support growing user bases.
* Database performance under load: Expected linear scaling through optimized indexing and caching strategies.

# Conclusion

This paper introduced Scholara, a comprehensive web-based platform developed to address the need for unified collaboration and achievement tracking in academic institutions. By combining role-specific dashboards, project and publication management, appointment scheduling, and a gamified leaderboard, Scholara bridges the gap between students and faculty, replacing fragmented and inefficient workflows with a cohesive, user-centered ecosystem. Evaluation results show that the platform is both performant and highly usable, enabling pilot users to efficiently complete key academic tasks. The underlying architecture provides a robust foundation for managing complex academic data, supporting a seamless experience for end users. Scholara delivers tangible value by enhancing the visibility of academic accomplishments, fostering meaningful collaboration, and providing data-driven insights to support career development. For students, it functions as a dynamic portfolio; for faculty, it offers a powerful tool for mentorship and academic management. As Scholara continues to evolve—incorporating advanced AI-driven analytics and expanded administrative functionality—it is well positioned to become an indispensable asset for academic institutions striving to build more connected, transparent, and productive communities.

##### References

1. H. Meishar-Tal and E. Pieterse, "Why Do Academics Use Academic Social Networking Sites?," Int. Rev. Res. Open Distrib. Learn., vol. 18, no. 1, pp. 1-22, Feb. 2017.
2. S. Ovadia, "ResearchGate and Academia.edu: Academic Social Networks," Behavioral & Social Sciences Librarian, vol. 33, no. 3, pp. 165-169, Jul. 2014, doi: 10.1080/01639269.2014.934093.
3. K. Jordan and M. Weller, "Academia.edu: Social network or academic network?," J. Interact. Media Educ., vol. 2014, no. 1, Art. no. 7, May 2014, doi: 10.5334/2014-07.
4. T. I. Fagborun, "Systematic Comparative Review of Three LMS Platforms: Moodle, Blackboard, and Canvas," Jul. 2024.
5. U.S. Department of Education, Office of Educational Technology, "Artificial Intelligence and the Future of Teaching and Learning: Insights and Recommendations," Washington, DC, USA, May 2023.
6. F. Ouyang and L. Zhang, "AI-driven learning analytics applications and tools in computer-supported collaborative learning: A systematic review," Educ. Res. Rev., vol. 43, Art. no. 100616, Nov. 2024, doi: 10.1016/j.edurev.2024.100616.
7. B. Oyarzun and F. Martin, "A Systematic Review of Research on Online Learner Collaboration from 2012-21: Collaboration Technologies, Design, Facilitation, and Outcomes," Online Learn., vol. 27, no. 1, pp. 71-106, Mar. 2023, doi: 10.24059/olj.v27i1.3407.
8. I. Bouchrika, "20 Best Academic Scheduling Software for 2025," Research.com, Aug. 6, 2025.
9. I. Jivet, M. Scheffel, M. Specht, and H. Drachsler, "The use of leaderboards in education: A systematic review of the literature," Comput. Appl. Learn., Oct. 2024, doi: 10.1111/jcal.13077.
10. O. Almousa, "Users' classification and usage-pattern identification in academic social networks," in Proc. IEEE Jordan Conf. Appl. Elect. Eng. Comput. Technol. (AEECT), Amman, Jordan, 2011, pp. 1-6, doi: 10.1109/AEECT.2011.6132492.
11. W. Tafesse, "Social networking sites use and college students’ academic performance: testing for an inverted U-shaped relationship using automated mobile app usage data," Int. J. Educ. Technol. High Educ., vol. 19, p. 16, 2022, doi: 10.1186/s41239-022-00322-0.
12. N. Salari, H. Zarei, S. Rasoulpoor, H. Ghasemi, A. Hosseinian-Far, and M. Mohammadi, "The impact of social networking addiction on the academic achievement of university students globally: A meta-analysis," Public Health in Practice, vol. 9, p. 100584, Jan. 2025, doi: 10.1016/j.puhip.2025.100584.
13. W. Yan, X. Zhang, and Y. Bromfield, "How does scholarly use of academic social networking sites differ by academic discipline? A case study using ResearchGate," Information Processing & Management, 2021, doi: 10.1016/j.ipm.2020.102430.
14. J. Iqbal, N. Qureshi, M. A. Ashraf, S. F. Rasool, and M. Z. Asghar, "The Effect of Emotional Intelligence and Academic Social Networking Sites on Academic Performance During the COVID-19 Pandemic," Psychology Research and Behavior Management, vol. 14, pp. 905–920, Jun. 2021, doi: 10.2147/PRBM.S316664.