International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

# Tapconnect: An AI-Powered NFC-Based Digital Professional Networking Platform

**Naresh MKS[1], Manoj Kumar J[2], Jelfin Pushparaj S [3], Mrs. R. Vijayalakshmi[4]**

[1,2,3]Student, CSE, Adhiyamaan College of Engineering,
[4]Project Guide & Assistant Professor, CSE, Adhiyamaan College of Engineering

**ABSTRACT**

TapConnect is an AI-powered, NFC-based digital professional networking and skill-matching platform developed to help college students showcase their technical skills, academic projects, achievements, and professional profiles efficiently. Traditional networking methods often require students to share resumes, social media links, and contact information separately, making professional information less convenient to access. TapConnect addresses this challenge by integrating Near Field Communication (NFC) technology, QR codes, and Artificial Intelligence (AI) into a unified digital platform.

The system enables students to create and maintain digital portfolios containing personal details, technical skills, academic projects, certifications, resumes, and professional links. These profiles can be accessed through NFC cards or QR codes, enabling quick and convenient profile sharing. The platform provides dedicated interfaces for students, department administrators, and super administrators to manage profiles, NFC cards, user accounts, and platform activities. AI-powered features integrated through the Google Gemini API support professional summary generation, resume review, project recommendations, skill analysis, and skill matching between students by identifying shared and complementary skills. Activity tracking and analytics also help administrators monitor platform usage and networking interactions.

TapConnect is developed using React and Vite for the frontend, Node.js and Express.js for the backend, and MongoDB for data storage. By combining digital portfolios, contactless profile sharing, and AI-assisted skill analysis, the platform aims to enhance students' digital presence, simplify professional networking, and encourage meaningful academic and career-oriented connections within the college community.

**KEYWORDS:** TapConnect, Near Field Communication (NFC), Artificial Intelligence (AI), Digital Portfolio, Professional Networking, Skill Matching, QR Code, Google Gemini API, Student Profile Management, MongoDB.

## 1.1 OVERVIEW

TapConnect is an AI-powered, NFC-based digital professional networking and skill-matching platform designed to improve professional connectivity among college students. Students develop technical skills, complete academic projects, and earn certifications throughout their academic journey. However, presenting these achievements effectively and connecting with peers who possess complementary skills

can be challenging. Traditional networking methods often involve sharing resumes, multiple online links, and contact details separately, making professional information less accessible and convenient to manage. TapConnect provides a centralized platform that enables students to create and maintain digital professional portfolios. These portfolios include personal information, department details, technical skills, academic projects, certifications, resumes, and professional links such as GitHub, LinkedIn, and personal portfolio websites. The platform uses Near Field Communication (NFC) technology and QR codes to enable quick access to student profiles. This approach simplifies profile sharing and allows users to explore relevant professional information without exchanging printed resumes or manually entering details.

Artificial Intelligence is integrated through the Google Gemini API to provide features such as professional summary generation, resume review, project recommendations, and skill analysis. The skill-matching feature identifies common and complementary skills between students, helping them discover potential collaborators and explore opportunities for knowledge sharing and professional growth. These AI-assisted capabilities make student profiles more informative and support skill discovery within the college community.

The platform also incorporates role-based access for students, department administrators, and super administrators. Students can manage their professional profiles, while department administrators handle student-related activities and NFC card management within their departments. Super administrators oversee platform-wide administration and activities. These features support organized user management, profile administration, and activity monitoring.

TapConnect is developed using React and Vite for the frontend, Node.js and Express.js for backend services, and MongoDB for database management. By combining digital portfolios, contactless profile sharing, QR code access, and AI-assisted skill matching, TapConnect aims to simplify professional networking, strengthen students' digital presence, and encourage meaningful academic and career-oriented connections.

### 1.2 OBJECTIVES

The primary objective of TapConnect is to develop an intelligent and user-friendly digital networking platform that enables college students to showcase their professional profiles and discover meaningful connections through Near Field Communication (NFC) technology, QR codes, and AI-powered skill matching. The platform aims to provide a centralized space where students can present their technical skills, academic projects, certifications, resumes, and professional achievements in an organized manner. Its skill-matching functionality helps identify common and complementary competencies among students, encouraging peer learning and potential collaboration.

The specific objectives are:

1. To develop digital student portfolios: Provide a centralized platform for students to maintain their personal information, technical skills, academic projects, certifications, resumes, and professional links.
2. To implement NFC-based profile sharing: Enable users to access student portfolios by tapping NFC-enabled cards, reducing the need for manual exchange of professional information.
3. To provide QR code accessibility: Allow users to open digital portfolios through QR codes, providing an alternative method of accessing profiles on compatible devices.
4. To integrate AI-powered features: Use the Google Gemini API to generate professional summaries, review resumes, recommend projects, and analyze students' skills.

5. To implement intelligent skill matching: Identify common and complementary skills between students to support networking, peer learning, and potential project collaboration.
6. To provide role-based access control: Establish separate functionalities for students, department administrators, and super administrators to ensure appropriate access to platform features.
7. To simplify student and NFC card management: Enable department administrators to create and manage student accounts, assign or replace NFC cards, and manage student records efficiently.
8. To monitor platform activity: Maintain activity records and analytics related to profile views, NFC interactions, downloads, and other relevant platform events.
9. To ensure secure profile management: Implement authentication, authorization, and account-management features to protect student information and restrict access to authorized operations.
10. To improve professional visibility and networking: Help students present their abilities effectively, discover relevant skills among peers, and strengthen their academic and career-oriented connections.

## LITERATURE SURVEY

The literature survey examines existing technologies and approaches related to digital professional networking, Near Field Communication (NFC), QR-code-based information sharing, artificial intelligence, and skill-matching systems.

**M. HEALY ET AL. (2023) – "LINKEDIN AS A PEDAGOGICAL TOOL FOR CAREERS AND EMPLOYABILITY LEARNING: A SCOPING REVIEW OF THE LITERATURE"**

Healy et al. (2023) reviewed the use of LinkedIn as a learning tool for career development and employability in higher education. The study discusses how professional networking platforms help students develop their professional identities, explore career opportunities, and connect with industry professionals. However, students may still need to share profile links manually when networking in person. TapConnect addresses this practical issue by providing a centralized student portfolio accessible through NFC-enabled cards and QR codes.

**NFC FORUM – "NFC TECHNOLOGY" (TECHNICAL RESOURCE)**

The NFC Forum describes Near Field Communication (NFC) as a short-range wireless technology that enables convenient interactions between compatible devices. NFC is used in contactless transactions, access systems, and information-sharing applications. In TapConnect, NFC-enabled cards link users directly to student portfolios through unique public tokens, simplifying the exchange of professional information without requiring manual entry of web addresses.

**DENSO – "QR CODE: HISTORY AND TECHNOLOGY" (TECHNICAL RESOURCE)**

DENSO developed the QR code in 1994 to enable efficient information storage and retrieval. QR codes are two-dimensional codes that can store web addresses and other data for scanning by compatible devices. They provide a convenient alternative to NFC because users can access digital profiles using a smartphone camera. TapConnect incorporates QR codes to provide an additional method of accessing and sharing student portfolios.

**M. TAVAKOLI ET AL. (2022) – "AN AI-BASED OPEN RECOMMENDER SYSTEM FOR PERSONALIZED LABOR MARKET DRIVEN EDUCATION"**

Tavakoli et al. (2022) presented an AI-based recommendation system that connects learners' educational goals with labour-market requirements and relevant learning opportunities. The study demonstrates how recommendation systems can support personalized education and skill development. TapConnect applies

a related concept through AI-powered professional summary generation, resume review, project recommendations, and skill analysis using the Google Gemini API.

**S. ALGARNI AND F. SHELDON (2023) – "SYSTEMATIC REVIEW OF RECOMMENDATION SYSTEMS FOR COURSE SELECTION"**

Algarni and Sheldon (2023) reviewed recommendation systems designed to help students select suitable courses. Their study discusses recommendation techniques and challenges associated with providing relevant suggestions when user information is limited. These findings emphasize the importance of accurate profile information in recommendation systems. TapConnect uses students' listed skills to identify common and complementary competencies, helping students discover potential peers for learning and collaboration.

**S. L. CHANG AND M. K. KABILAN (2022) – "USING SOCIAL MEDIA AS E-PORTFOLIOS TO SUPPORT LEARNING IN HIGHER EDUCATION: A LITERATURE ANALYSIS"**

Chang and Kabilan (2022) examined the use of social media as electronic portfolios in higher education. Their literature analysis discusses how digital portfolios can support learning, reflection, and the presentation of students' achievements. TapConnect follows this approach by allowing students to organize their technical skills, projects, certifications, resumes, and professional links in a centralized digital portfolio. It extends the concept by integrating NFC access, QR codes, and AI-assisted skill matching.

### 2.1 RESEARCH GAP

The existing studies highlight the importance of digital portfolios, professional networking platforms, contactless technologies, and AI-based recommendation systems in improving access to information and supporting professional development. However, students may still need to use separate applications to showcase their achievements, share professional profiles, analyze their skills, and identify suitable peers for collaboration. This can make the networking process less convenient and time-consuming.

TapConnect proposes an integrated solution by bringing these functionalities together in a centralized platform. It enables students to maintain digital portfolios, share profiles through NFC cards and QR codes, and use AI-powered features to improve their professional presentation. In addition, the skill-matching functionality helps students identify shared competencies and complementary technical abilities among their peers. The inclusion of role-based access control and activity monitoring supports organized administration and efficient platform management.

### 2.2 SUMMARY

The literature survey establishes the foundation for TapConnect by examining professional networking, NFC technology, QR codes, AI-based recommendations, and digital portfolios. These technologies offer complementary benefits for professional visibility, convenient profile sharing, and skill discovery. TapConnect combines these capabilities to support student networking and career-oriented development.

### 3.1. EXISTING SYSTEM

In the existing approach to professional networking, college students generally depend on conventional methods and separate digital platforms to share their skills, academic achievements, resumes, and professional profiles. Students often use social networking websites, email, messaging applications, and online resume platforms to communicate their professional information. These methods provide basic profile-sharing capabilities but may require users to manage information across multiple platforms.

Traditional networking frequently involves exchanging contact details manually or sharing profile links individually. This process can be time-consuming during college events, technical symposiums, workshops, and placement activities. Identifying students with similar or complementary technical skills also requires manual searching and communication.

Furthermore, existing profile-sharing methods may not provide integrated AI-based resume analysis, professional summary generation, project recommendations, and skill comparison in a single platform. Managing student information and monitoring networking activities across departments can also be challenging without a centralized administrative system.

**Limitations of the Existing System**

- **Fragmented profile information**: Student details, projects, resumes, and professional links may be distributed across multiple platforms.
- **Manual profile sharing**: Exchanging contact information and profile links requires additional effort.
- **Limited skill discovery**: Finding students with relevant or complementary technical skills may involve manual searching.
- **Lack of integrated AI assistance**: Professional profile improvement, resume review, and project recommendations may require separate tools.
- **Limited centralized administration**: Managing student accounts and networking activities across departments can be difficult.
- **Reduced networking efficiency**: Students may find it challenging to discover relevant peers and present their achievements quickly during in-person interactions.

### 3.2. PROPOSED SYSTEM

TapConnect is proposed as an AI-powered, NFC-based digital professional networking and skill-matching platform specifically designed for college students. The system combines digital portfolios, NFC technology, QR-code access, artificial intelligence, and centralized administration to provide an integrated networking experience.

The platform enables students to create and maintain digital portfolios containing their personal information, department, technical skills, academic projects, certifications, resume, and professional links such as GitHub, LinkedIn, and portfolio websites. Each student profile can be accessed through an NFC-enabled card or QR code, allowing users to view professional information conveniently using compatible devices.

TapConnect incorporates AI-powered features through the Google Gemini API. These features include professional summary generation, resume review, project recommendations, and skill analysis. Its skill-matching functionality compares student profiles to identify common and complementary skills, helping users discover potential connections based on their technical abilities.

The system provides three levels of access: Super Admin, Department Admin, and Student. The Super Admin manages department administrators and monitors overall platform activity. Department administrators manage student accounts, assign or replace NFC cards, and view department-level analytics. Students can update their profiles, manage their professional information, and use AI-powered tools to improve their career readiness.

The platform is developed using React and Vite for the frontend, Node.js and Express.js for backend services, and MongoDB for data storage. Authentication, role-based access control, NFC card management, and activity logging support secure and organized platform operations.

**Key Features of the Proposed System**

- Digital portfolio management: Maintains student profiles, skills, projects, certifications, resumes, and professional links.
- NFC-based networking: Provides quick access to digital portfolios through NFC-enabled cards.
- QR-code integration: Offers an alternative way to access student profiles by scanning QR codes.
- AI-powered professional assistance: Generates professional summaries, reviews resumes, recommends projects, and analyzes skills.
- Intelligent skill matching: Identifies common and complementary skills between students.
- Role-based administration: Provides separate functionalities for students, department administrators, and super administrators.
- NFC card management: Supports card assignment, replacement, deactivation, and related administrative operations.
- Activity monitoring and analytics: Records relevant interactions, such as profile views and downloads, to support usage analysis.
- Centralized data management: Maintains student profiles, card records, and relevant platform data in MongoDB.

**ADVANTAGES OF PROPOSED SYSTEM**

- The proposed TapConnect platform offers several advantages over conventional professional networking approaches.
- Faster information sharing: NFC cards and QR codes enable users to access digital portfolios without manually exchanging multiple links or printed documents.
- Improved professional visibility: Students can present their skills, academic projects, certifications, resumes, and achievements through a single digital profile.
- AI-assisted career development: AI-powered tools help students improve their professional summaries, review resumes, analyze skills, and explore project recommendations.
- Efficient skill discovery: The skill-matching feature helps students identify peers with shared interests or complementary technical abilities.
- Centralized student management: Department administrators can manage student accounts and NFC cards through a common administrative interface.
- Role-based access control: Separate permissions for students, department administrators, and super administrators help restrict operations according to user responsibilities.
- Convenient profile accessibility: Support for both NFC and QR codes provides alternative methods of accessing profiles on compatible devices.
- Improved activity tracking: Activity records and analytics help administrators understand platform usage and monitor relevant networking interactions.
- Reduced dependence on printed materials: Digital portfolios reduce the need to distribute printed resumes or physical copies of professional information during networking events.
- Support for academic networking: By combining profile sharing with skill analysis, TapConnect encourages students to discover relevant peers and explore opportunities for learning and project collaboration.
- Overall, TapConnect provides a unified approach to digital profile sharing, AI-assisted skill analysis, and student networking.

### 3.3. PROPOSED SOLUTION

TapConnect is an AI-powered, NFC-based digital professional networking platform designed to simplify professional profile sharing and skill discovery among college students. The proposed solution integrates NFC technology, QR codes, digital portfolios, artificial intelligence, and centralized administration into a single web-based application.

The system enables students to create professional digital portfolios containing their personal details, technical skills, academic projects, certifications, resumes, and professional links. These portfolios can be accessed through NFC-enabled cards or QR codes, allowing users to view relevant student information quickly and conveniently.

The platform integrates the Google Gemini API to provide AI-powered features such as professional summary generation, resume review, project recommendations, and skill analysis. Its skill-matching functionality compares the skills of two students to identify common competencies and complementary technical abilities. This helps students discover peers with relevant expertise and explore potential opportunities for knowledge sharing and project collaboration.

TapConnect also provides role-based administration to support the organized management of the platform. The Super Admin manages department administrators and monitors overall activity, while Department Admins manage student accounts, NFC card assignments, and department-level information. Students can update their portfolios and access the available AI tools.

The application is developed using React and Vite for the frontend, Node.js and Express.js for backend services, and MongoDB for data storage. Authentication, role-based authorization, NFC card management, and activity logging are incorporated to support secure and efficient operation.

**Major Components of the Proposed Solution**

- **Digital Portfolio Module:** Stores and displays students' professional information, skills, projects, certifications, resumes, and external links.
- **NFC and QR Code Module:** Enables quick access to student portfolios using NFC-enabled cards or scannable QR codes.
- **AI Center Module:** Provides AI-assisted professional summaries, resume feedback, project recommendations, and skill analysis.
- **Skill Matching Module:** Compares student skills and identifies shared.
- **Administration Module:** Supports student account management, department administration, NFC card assignment, and platform monitoring.
- **Analytics and Activity Module:** Records relevant platform interactions and presents activity information to support administrative monitoring.

### 3.4. PROBLEM SOLUTION FIT

Problem–Solution Fit describes how the proposed system addresses the practical problems faced by its intended users. TapConnect is designed to meet the needs of college students who want to present their professional achievements, share their profiles efficiently, and discover peers with relevant technical skills. It also supports administrators who need a centralized method for managing student accounts and NFC cards.

The following table summarizes the major problems and the corresponding solutions provided by TapConnect.

| Identified Problem | Proposed Solution | Expected Benefit |
|---|---|---|
| Difficulty sharing professional information quickly | NFC-enabled cards and QR-code-based portfolio access | Faster and more convenient profile sharing |
| Student information distributed across multiple platforms | Centralized digital student portfolios | Easier access to professional details |
| Difficulty identifying students with relevant technical skills | AI-assisted skill analysis and skill matching | Improved skill discovery and networking |
| Time required to prepare professional summaries | AI-powered professional summary generation | Simplified profile preparation |
| Difficulty identifying improvements in resumes | AI-powered resume review | Useful feedback for improving resume quality |
| Limited guidance for selecting projects | AI-based project recommendations | Support for project exploration and skill development |
| Complex management of student accounts and cards | Role-based administration and NFC card management | More organized administrative operations |
| Difficulty monitoring platform interactions | Activity logs and analytics | Better visibility into platform usage |

.

### 3.5 IDEATION & BRAINSTORMING

Ideation is the process of identifying problems, exploring possible solutions, and developing innovative ideas to create an effective system. The idea behind TapConnect emerged from the need to simplify professional networking among college students and provide a convenient way to showcase their technical skills, academic achievements, and projects.

Students often participate in technical events, workshops, symposiums, and project exhibitions where sharing professional information and identifying peers with relevant skills can be challenging. Traditional methods of exchanging contact details or sharing profile links may require additional time and effort. Furthermore, students may find it difficult to identify peers with complementary technical abilities or obtain guidance on improving their professional profiles.

**BRAINSTORMING**

Brainstorming was conducted to identify the essential features required to make TapConnect useful, practical, and efficient. Different ideas were considered based on student requirements, ease of use, technological feasibility, and administrative needs.

The major ideas identified during brainstorming are listed below:

1. **Digital Professional Profiles:** Create a centralized profile containing students' personal information, technical skills, academic projects, certifications, resumes, and professional links.

2. **NFC-Based Profile Sharing:** Use NFC-enabled cards to provide quick access to digital portfolios through compatible devices.
3. **QR-Code Integration:** Provide QR codes as an alternative method of accessing student profiles without requiring NFC functionality.
4. **AI-Powered Professional Summaries:** Integrate artificial intelligence to help students generate clear and professional profile summaries.
5. **AI-Based Resume Review:** Provide feedback on resumes to help students identify areas that may need improvement.
6. **Project Recommendations:** Suggest relevant project ideas based on students' skills and professional interests.
7. **Intelligent Skill Matching:** Compare student profiles to identify common and complementary technical skills, helping students discover relevant peers.
8. **Role-Based Administration:** Develop separate dashboards for students, department administrators, and super administrators to manage access and platform operations.
9. **NFC Card Management:** Provide administrative functions to assign, replace, and deactivate NFC cards when required.
10. **Activity Tracking and Analytics:** Record relevant profile interactions and present usage information to support administrative monitoring.

**3.6 ARCHITECTURE DESIGN:**

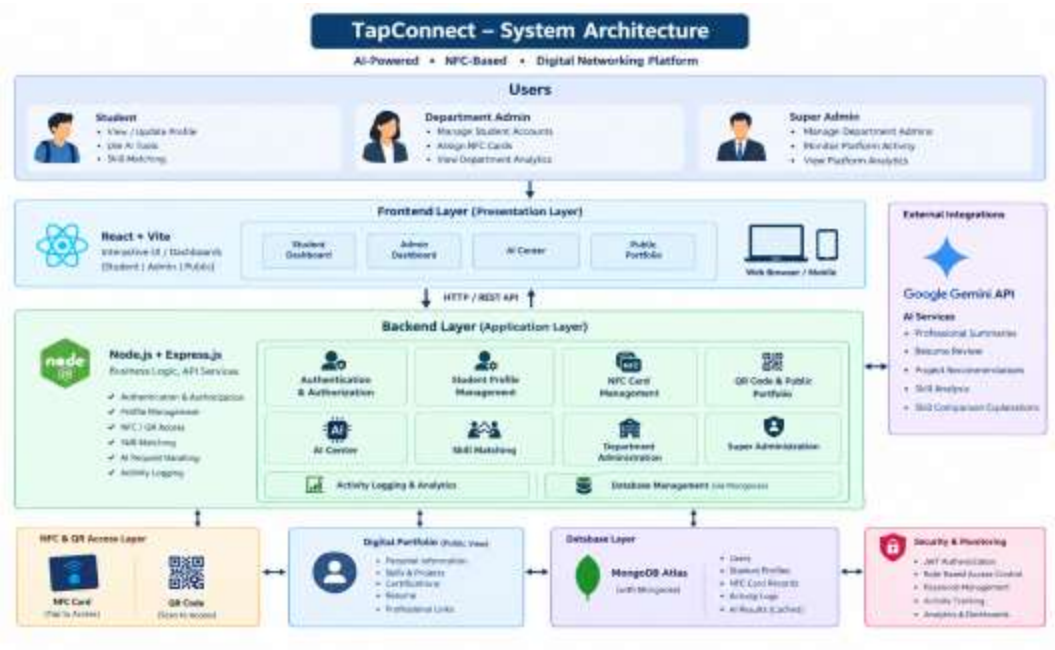



***Figure 3.1: Architectural Design***

The architecture of TapConnect follows a client–server architecture that integrates the frontend interface, backend services, database management, NFC and QR-code-based profile access, and AI-powered functionalities. The architecture is designed to provide a centralized platform for student networking, professional profile management, skill matching, and administrative operations.

The frontend is developed using React and Vite, providing interactive dashboards and user interfaces for students, department administrators, and super administrators. The backend uses Node.js and Express.js to handle HTTP requests, authentication, authorization, business logic, profile management, and

communication with external AI services. MongoDB Atlas serves as the database for storing user profiles, NFC card records, activity logs, and AI-generated results.

The Google Gemini API is integrated to provide AI-powered features, including professional summary generation, resume review, project recommendations, and skill analysis. NFC cards and QR codes direct users to the corresponding public student portfolios. The backend validates requests and retrieves the relevant profile information before presenting it to the user.

Authentication and role-based access control help ensure that users can access only the functionalities permitted by their roles. Activity logging records relevant interactions to support analytics and administrative monitoring.

**Architecture Components**

1. **Presentation Layer:** Provides the user interface through React and Vite, including student dashboards, administrative dashboards, AI tools, and public portfolios.
2. **Application Layer:** Uses Node.js and Express.js to process requests, apply business rules, manage authentication, and coordinate interactions between application components.
3. **Database Layer:** Uses MongoDB Atlas and Mongoose to store and retrieve user information, student profiles, NFC card details, activity logs, and AI results.
4. **AI Integration Layer:** Connects the backend to the Google Gemini API to generate professional summaries, review resumes, recommend projects, and provide skill-analysis explanations.
5. **NFC and QR Access Layer:** Enables profile access through NFC-enabled cards and QR codes that direct users to the relevant digital portfolio.
6. **Security and Monitoring Layer:** Supports JWT-based authentication, role-based authorization, password management, and activity tracking.

### 3.7 MODULE DESCRIPTION

TapConnect is organized into the following major modules to support modular development, maintainability, and efficient operation.

**Authentication and Authorization Module**

This module manages user login, authentication, and access permissions. JWT-based authentication is used to verify authenticated requests, while role-based authorization restricts access according to the user's assigned role.

The module supports student, department administrator, and super administrator accounts. It also provides password-management features, including password changes and password-reset functionality. Students using temporary credentials can be required to change their passwords before proceeding.

**Main functions:**

User login and authentication.

- JWT-based session verification.
- Role-based access control.
- Password changes and password resets.
- Protection of restricted administrative operations.

**Student Profile Management Module**

This module enables students to create, update, and maintain their digital professional portfolios. The portfolio serves as a centralized representation of a student's technical abilities, academic achievements, and professional interests.

Students can manage personal information, department details, technical skills, projects, certifications, resumes, and external professional links.

**Main functions:**

- Create and update student profiles.
- Manage technical skills and project details.
- Add certifications and professional links.
- Maintain resume and portfolio information.
- Display public professional profiles.

**NFC Card Management Module**

This module manages NFC cards associated with student profiles. Department administrators can assign cards, replace existing cards, deactivate cards, and review relevant card history.

Each card uses a unique public token to identify the associated portfolio. The public token is used for profile access rather than exposing the student's MongoDB identifier in the NFC URL.

**Main functions:**

- Assign NFC cards to students.
- Generate or manage unique public tokens.
- Replace and deactivate cards.
- Track card assignment history.
- Connect NFC-based profile access with activity logging.

**QR Code and Public Portfolio Module**

This module enables student portfolios to be accessed through QR codes and public profile URLs. Users can scan a QR code or tap an NFC-enabled card to open the relevant portfolio in a browser.

The public portfolio displays the student's permitted professional information, including skills, projects, certifications, resume, and external links. Relevant interactions, such as profile views and downloads, can be recorded for activity analysis.

**Main functions:**

- Provide public portfolio access.
- Support QR-code-based profile sharing.
- Display student professional information.
- Provide access to permitted resume and project information.
- Record relevant profile interactions.

**AI Center Module**

The AI Center integrates the Google Gemini API to provide AI-assisted professional development features. Requests are handled through the backend, which communicates with the AI service and returns generated results to the frontend.

AI-generated results can be cached in the database to reduce repeated API calls and improve response efficiency.

**Main functions:**

- Generate professional summaries.
- Review resumes and provide improvement suggestions.
- Recommend projects based on student skills.
- Analyze technical skills and identify potential skill gaps.

- Generate explanations for student skill comparisons.

**Skill Matching Module**

The Skill Matching Module compares the skills of two student profiles to identify similarities and complementary competencies. The comparison helps students understand which skills they share and where their technical abilities differ.

The authenticated student's identity is obtained from the authentication context, while the target student is identified through the selected profile or NFC/QR access information. Common and unique skill sets can be determined through backend comparison logic, with AI used to explain the results in natural language.

**Main functions:**

- Retrieve the authenticated student's skills.
- Identify the target student's skills.
- Calculate common skills.
- Identify skills unique to each student.
- Present a readable comparison to support networking and learning.

**Department Administration Module**

This module enables department administrators to manage student information and NFC card assignments within their authorized department. It provides a centralized interface for handling routine administrative operations.

Student accounts can be created individually or through bulk import using supported CSV or XLSX files. Department-level analytics and activity records help administrators monitor relevant operations.

**Main functions:**

- Create and manage student accounts.
- Import student records in bulk.
- Assign, replace, and deactivate NFC cards.
- View department-level activity and analytics.

**Super Administration Module**

The Super Administration Module provides platform-wide administrative capabilities. It enables the super administrator to manage department administrators and monitor activity across the platform.

Access to this module is restricted according to the user's role to prevent unauthorized administrative operations.

**Main functions:**

- Create and manage department administrators.
- Monitor overall platform activity.
- Review platform-wide analytics.
- Support centralized administrative control.
- Maintain separation between department-level and platform-wide permissions.

**Activity Logging and Analytics Module**

This module records relevant events generated during platform usage. Depending on the operation, activity records can include profile views, NFC interactions, downloads, link clicks, and administrative actions.

The stored information supports dashboards, activity timelines, and analytics that help administrators understand platform usage and review relevant events.

**Main functions:**

- Record relevant user activities.
- Track profile interactions and downloads.
- Maintain administrative activity history.
- Present activity timelines and analytics.
- Support monitoring of platform usage.

**Database Management Module**

The database module uses MongoDB Atlas with Mongoose to store and manage the application's structured data. It supports the persistence and retrieval of user accounts, profiles, NFC card information, activity logs, and cached AI results.

The database layer is accessed through backend services so that validation, authorization, and business rules can be applied before data is read or modified.

**Main functions:**

- Store user and student profile information.
- Manage NFC card records and public tokens.
- Maintain activity logs and AI results.
- Retrieve data for dashboards and public portfolios.
- Support consistent and controlled database operations.

### 3.8 MODULE INTERACTION

The modules work together to provide a unified and efficient networking experience for students and administrators. Each module performs a specific function while communicating with other modules through the backend services. This modular approach ensures that profile management, NFC and QR-based access, AI-powered services, and administrative operations function as parts of a single integrated system.

When a student updates their profile, the Student Profile Management Module validates the submitted information and sends the request to the backend. The backend processes the request and stores the updated details in MongoDB, ensuring that the latest profile information is available whenever it is accessed. Similarly, when another user scans an NFC card or QR code, the Public Portfolio Module uses the associated public identifier to retrieve the relevant student profile and display the information permitted for public access.

The Activity Logging and Analytics Module records relevant events, such as profile views, NFC interactions, and other supported platform activities. The recorded information can be used to monitor usage patterns and provide administrators with a better understanding of how the platform is being used. This supports activity tracking and helps evaluate the effectiveness of the networking features.

The frontend, developed using React and Vite, provides the user interface through which students and administrators interact with the system. The Node.js and Express.js backend manages requests, applies business logic, communicates with MongoDB, and coordinates AI service integration. MongoDB stores application data, while the Google Gemini API provides AI-generated assistance. This separation of responsibilities helps keep the system organized and makes individual modules easier to develop and maintain.

Overall, the interaction between these modules enables TapConnect to combine digital portfolio management, contactless profile sharing, AI-assisted professional development, student skill matching, and administrative monitoring within one platform. The modular architecture improves maintainability,

supports future enhancements, and allows additional features to be introduced with minimal disruption to existing functionality. It also provides a structured foundation for expanding the platform to support more students, additional networking features, and future academic or career-oriented requirements.

## 4. HARDWARE AND SOFTWARE REQUIREMENTS

The development and execution of TapConnect require suitable hardware and software resources to support the frontend interface, backend services, database operations, NFC and QR-code-based profile access, and AI-powered functionalities. The following requirements describe the recommended environment for developing and using the system.

### 4.1 HARDWARE REQUIREMENTS

The hardware requirements specify the minimum recommended resources needed to develop, run, and test the TapConnect application.

| S. No. | Hardware Component | Minimum Requirement |
|---|---|---|
| 1 | Processor | Intel Core i3 or equivalent |
| 2 | RAM | 4 GB |
| 3 | Storage | 10 GB of available space |
| 4 | Display | Standard monitor or laptop display |
| 5 | Network | Internet connection |
| 6 | Smartphone | Android or iOS device with a web browser |
| 7 | NFC Device | NFC-enabled smartphone or NFC reader for tap-based testing |
| 8 | QR Code Scanner | Smartphone camera or QR-scanning application |

A computer with higher processing power and at least 8 GB of RAM is recommended for smoother development and simultaneous execution of the frontend, backend, and development tools.

An NFC-enabled smartphone is required to test physical NFC card interactions. However, the public portfolio can also be accessed through a QR code or a direct web link. Therefore, an NFC-capable device is not necessary for every user.

### 4.2 SOFTWARE REQUIREMENTS

The software requirements specify the frameworks, programming languages, services, and development tools used to implement and operate TapConnect.

| S. No. | Software Component | Technology Used |
|---|---|---|
| 1 | Operating System | Windows, Linux, or macOS |
| 2 | Frontend Framework | React.js |
| 3 | Frontend Build Tool | Vite |
| 4 | Styling Framework | Tailwind CSS |
| 5 | Programming Language | JavaScript |
| 6 | Backend Runtime | Node.js |
| 7 | Backend Framework | Express.js |

| | | |
|---|---|---|
| 8 | Database | MongoDB Atlas |
| 9 | Database ODM | Mongoose |
| 10 | AI Integration | Google Gemini API |
| 11 | Authentication | JSON Web Token (JWT) |
| 12 | API Communication | HTTP / REST API |
| 13 | Development Environment | Visual Studio Code or equivalent editor |
| 14 | Version Control | Git |
| 15 | Web Browser | Google Chrome, Microsoft Edge, or Mozilla Firefox |
| 16 | Email Service | Nodemailer with a configured SMTP service |

### 4.3 FUNCTIONAL SOFTWARE REQUIREMENTS

The following software components support the major functionalities of TapConnect:

- **Frontend Environment:** React and Vite provide the user interface for student dashboards, administrative dashboards, the AI Center, and public portfolios.
- **Backend Environment:** Node.js and Express.js handle API requests, authentication, authorization, profile management, NFC card operations, and activity logging.
- **Database Environment:** MongoDB Atlas stores user information, student profiles, NFC card records, activity logs, and cached AI results.
- **Artificial Intelligence Service:** The Google Gemini API supports professional summary generation, resume review, project recommendations, and skill analysis.
- **Authentication Mechanism:** JWT-based authentication and role-based access control protect restricted application features.
- **Networking Technology:** NFC cards and QR codes provide access to public digital portfolios through compatible devices and web browsers.
- **Email Functionality:** Nodemailer and a configured email service support initial credential delivery and password-reset workflows.

### 4.4 NETWORK AND CONNECTIVITY REQUIREMENTS

TapConnect requires an internet connection for communication between the frontend, backend, MongoDB Atlas, and Google Gemini API. Internet connectivity is also required for accessing public student portfolios and using cloud-hosted application services.

NFC-enabled devices must support the relevant NFC functionality to perform physical tap-based profile access. QR-code access requires a compatible camera or scanning application. Both methods ultimately direct users to the web application.

### 5.1 IMPLEMENTATION

The implementation of TapConnect involves developing an integrated web application that combines digital portfolio management, NFC-based profile sharing, QR-code access, AI-powered professional assistance, and student skill matching. The system follows a client–server architecture in which the frontend communicates with backend services through REST APIs, while MongoDB Atlas manages persistent application data.

The frontend is developed using React and Vite to provide an interactive and responsive user interface. The backend is implemented using Node.js and Express.js to handle authentication, business logic, profile management, NFC card operations, AI requests, and activity tracking. MongoDB Atlas is used to store user accounts, student profiles, NFC card records, activity logs, and AI-generated results.

The Google Gemini API is integrated to provide intelligent features that support students in improving their professional profiles and analyzing their skills. The application also incorporates role-based access control to distinguish the permissions of students, department administrators, and super administrators. The implementation is divided into several modules, each responsible for a specific set of functionalities.

**User Authentication and Access Control**

The authentication module provides secure access to the platform. Users log in using their credentials, and the backend verifies their identity before granting access to protected resources. JWT-based authentication is used to validate requests, while role-based authorization controls access to student, department administrator, and super administrator functionalities.

The module also supports password changes and password-reset workflows. Where temporary credentials are used, students can be required to change their passwords before accessing the remaining application features.

**Student Profile Implementation**

The student profile module allows students to create and maintain their digital professional portfolios. The information may include personal details, department, technical skills, academic projects, certifications, resume, GitHub, LinkedIn, and other professional links.

The backend validates profile-related requests and stores the relevant information in MongoDB. The frontend retrieves and displays the stored information through the student dashboard and public portfolio interface.

**NFC Card and QR Code Implementation**

The NFC module associates physical NFC cards with student portfolios. Each active NFC card uses a unique public token to identify the corresponding profile. When a compatible device taps the card, the browser opens the associated public portfolio.

QR codes provide an alternative method of accessing student profiles. Users can scan the code using a compatible smartphone camera or QR-scanning application. Both access methods are intended to make professional information available without requiring users to exchange profile links manually.

The backend processes profile-access requests and records relevant interactions for activity tracking and analytics.

**AI Center Implementation**

The AI Center integrates the Google Gemini API through backend services. It provides features designed to assist students in professional profile development and career preparation.

The main AI-powered functionalities include:

- **Professional Summary Generation:** Generates a professional summary based on relevant student profile information.
- **Resume Review:** Analyzes submitted resume content and provides suggestions for improvement.
- **Project Recommendations:** Suggests project ideas based on the student's skills and related information.
- **Skill Analysis:** Examines listed skills and identifies potential areas for development.

- **Skill Comparison Explanations:** Produces readable explanations of common and complementary skills between student profiles.

AI-generated results can be cached in MongoDB to reduce repeated requests and improve response efficiency. AI outputs are intended to assist users and should be reviewed before being used in professional applications.

**Skill Matching Implementation**

The skill-matching module compares the skills of two student profiles. The authenticated user's identity is obtained from the authentication context, while the target student is identified from the selected profile or NFC/QR access information.

The backend compares the two skill sets to identify common skills and competencies unique to each student. AI-generated explanations can then help users understand the comparison. This approach supports skill discovery and helps students identify peers with related or complementary technical abilities.

**Administrative Implementation**

TapConnect provides separate administrative interfaces according to user roles. The **Department Admin Module** supports the creation and management of student accounts, bulk student import using supported CSV or XLSX files, NFC card assignment and replacement, card deactivation, and department-level activity monitoring.

The **Super Admin Module** supports the management of department administrators and platform-wide activity monitoring. Role-based authorization helps ensure that administrative operations remain within the appropriate access boundaries.

**Activity Logging and Analytics**

The activity logging module records relevant interactions, including profile views, NFC-related activity, downloads, professional link clicks, and administrative operations. The recorded information can be used to display activity timelines and analytics in the appropriate dashboards. These features help administrators understand platform usage and review relevant activities. Access to activity information is controlled according to the user's role and permissions.

**5.2 SOFTWARE DESCRIPTION**

The software technologies used in TapConnect are selected to support frontend development, backend processing, data persistence, authentication, AI integration, and profile accessibility.

**React.js**

React.js is a JavaScript library used to build interactive user interfaces. In TapConnect, React is used to develop student dashboards, administrative dashboards, the AI Center, and public portfolio pages. Its component-based structure supports reusable interface elements and simplifies the maintenance of different application sections.

**Vite**

Vite is a frontend development and build tool used with React. It provides a development server and build process for the application. In TapConnect, Vite supports frontend development and generates the production-ready frontend assets.

**Tailwind CSS**

Tailwind CSS is a utility-first CSS framework used to style the application interface. It supports consistent layouts, responsive dashboards, forms, cards, navigation elements, and other interface components.

**Node.js**

Node.js is a JavaScript runtime used to execute the backend application. It enables TapConnect to process requests, run application logic, communicate with the database, and integrate external services.

**Express.js**

Express.js is a web framework for Node.js used to create backend routes and REST APIs. In TapConnect, it handles authentication requests, student profile operations, NFC card management, AI requests, administrative functions, and activity logging.

**MongoDB Atlas**

MongoDB Atlas is a cloud-hosted NoSQL database service used to store application data. TapConnect uses MongoDB to manage user accounts, student profiles, NFC card records, activity logs, and cached AI results. Its document-oriented data model supports the storage of profile information and related application records.

**Mongoose**

Mongoose is an object data modeling library for MongoDB and Node.js. It provides schemas, validation, and structured database operations. TapConnect uses Mongoose to define and manage application data models, including users, NFC cards, activity logs, and AI results.

**Google Gemini API**

The Google Gemini API provides the AI capabilities used in TapConnect. It supports professional summary generation, resume feedback, project recommendations, skill analysis, and natural-language explanations of skill comparisons. API requests are handled through the backend so that the API key can remain outside the frontend code.

**JSON Web Token (JWT)**

JSON Web Token is used as part of the application's authentication mechanism. After successful authentication, a token can be used to verify subsequent requests to protected backend routes. JWT-based authentication works together with role-based authorization to control access to application resources.

**Nodemailer**

Nodemailer is a Node.js library used to send emails through a configured email transport service. In TapConnect, it supports email-related workflows such as sending initial student credentials and password-reset information.

**Git**

Git is a version control system used to track source-code changes during development. It helps developers maintain code history, manage modifications, and collaborate on the TapConnect project.

### 5.3 WORKING PROCEDURE OF THE SYSTEM

The general working procedure of TapConnect is as follows:

1. The user opens the application through a web browser and logs in using the appropriate credentials when authentication is required.
2. The backend validates the user's identity and permissions.
3. Students create or update their professional profiles through the student dashboard.
4. The profile information is stored in MongoDB Atlas.
5. A student portfolio can be accessed through an assigned NFC card, a QR code, or a public profile URL.
6. When an AI feature is requested, the backend processes the request and communicates with the Google Gemini API.

7. For skill matching, the system compares the authenticated user's skills with those of the selected target student and presents the comparison.
8. Relevant profile interactions and administrative activities are recorded for analytics.
9. Department administrators and super administrators access the management functions permitted by their roles.

**5.4 SUMMARY**

The implementation of TapConnect combines modern web development technologies with NFC-based access, QR codes, centralized data storage, and AI-powered professional assistance. React and Vite provide the frontend, Node.js and Express.js manage backend operations, and MongoDB Atlas stores application data. The Google Gemini API supports AI-related features, while authentication and role-based authorization help protect application resources.

**5.5 CODE IMPLEMENTATION**

The TapConnect system is implemented using a client-server architecture. The frontend is developed using React.js and Vite, while the backend is developed using Node.js and Express.js. MongoDB Atlas is used to store application data, and Google Gemini API is integrated to provide AI-based features.

The implementation is divided into modules for authentication, student profile management, NFC and QR code access, AI-based skill matching, administrative management, and activity tracking.

**Backend Server Implementation**

The backend server handles API requests from the frontend and connects the application to the database. Express.js is used to define routes, while middleware supports request processing and error handling.

**Sample Code:**

```
// server.js
const express = require("express");
const cors = require("cors");
const mongoose = require("mongoose");
const app = express();
app.use(cors());
app.use(express.json());
mongoose
.connect(process.env.MONGODB_URI)
.then(() => console.log("MongoDB connected"))
.catch((error) => console.error("Database connection failed"));
app.get("/api/health", (req, res) => {
res.json({
status: "success",
message: "TapConnect API is running"
});
});
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => {
console.log(`Server running on port ${PORT}`);
});
```

The server establishes a connection with MongoDB and provides an API endpoint for checking server availability. In the complete application, authentication, profile, NFC, administrative, and AI routes are registered separately.

**Student Profile Implementation**

The student profile module stores professional information such as department, skills, projects, certificates, and external profile links. Mongoose schemas help define the structure of the stored information.

**Sample Code:**

```
// models/User.js
const mongoose = require("mongoose");
const userSchema = new mongoose.Schema(
{
name: {
type: String,
required: true
},
email: {
type: String,
required: true,
unique: true
},
role: {
type: String,
enum: ["student", "admin", "super_admin"],
default: "student"
},
department: String,
bio: String,
skills: [String],
projects: [String],
certificates: [String],
github: String,
linkedIn: String,
portfolio: String
},
{ timestamps: true }
);
module.exports = mongoose.model("User", userSchema);
```

This sample demonstrates how user and professional profile information can be represented in MongoDB. The actual project schema may include additional fields for authentication, resumes, department references, and other profile information.

**NFC Card and Public Portfolio Implementation**

TapConnect uses a unique public token to associate an NFC card with a student's public portfolio. When the card is tapped using a compatible smartphone, the browser opens the portfolio URL associated with that token.

**Sample Code:**

```
// Example Express route
app.get("/api/tap/:publicToken", async (req, res) => {
try {
const card = await NFCCard.findOne({
publicToken: req.params.publicToken,
status: "active"
});
if (!card) {
return res.status(404).json({
message: "Active NFC card not found"
});
}
const student = await User.findById(card.student);
if (!student) {
return res.status(404).json({
message: "Student profile not found"
});
}
res.json({
name: student.name,
department: student.department,
bio: student.bio,
skills: student.skills,
projects: student.projects,
github: student.github,
linkedIn: student.linkedIn,
portfolio: student.portfolio
});
} catch (error) {
res.status(500).json({
message: "Unable to retrieve portfolio"
});
}
});
```

This example illustrates token-based NFC portfolio retrieval. The NFCCard model, its relationship to the student, and the status field must match the actual project implementation. QR codes can provide a second entry point to the public portfolio.

**AI-Based Skill Matching Implementation**

The skill matching module compares the authenticated user's skills with those of the target student. Common skills and unique skills are calculated programmatically, while the AI service can generate a natural-language explanation of the comparison.

**Sample Code:**

```
function compareSkills(viewerSkills = [], targetSkills = []) {
const normalize = (skills) =>
skills
.filter((skill) => typeof skill === "string")
.map((skill) => skill.trim())
.filter(Boolean);
const viewer = normalize(viewerSkills);
const target = normalize(targetSkills);
const viewerSet = new Set(
viewer.map((skill) => skill.toLowerCase())
);
const targetSet = new Set(
target.map((skill) => skill.toLowerCase())
);
return {
commonSkills: target.filter((skill) =>
viewerSet.has(skill.toLowerCase())
),
viewerUniqueSkills: viewer.filter(
(skill) => !targetSet.has(skill.toLowerCase())
),
targetUniqueSkills: target.filter(
(skill) => !viewerSet.has(skill.toLowerCase())
)
};
}
module.exports = { compareSkills };
```

The function identifies common skills and skills unique to each profile without relying on AI to perform the basic comparison. The resulting information can be sent to the Gemini API to generate an explanation of possible areas of shared interest and complementary expertise.

In the complete system, the viewer's identity must be obtained from the authenticated session, and the target profile must be validated by the backend.

**Frontend Implementation**

React.js is used to build the user interface and display student profiles, dashboards, and AI results. The frontend communicates with the backend through HTTP requests.

**Sample Code:**

```
import { useEffect, useState } from "react";
function PublicPortfolio({ publicToken }) {
const [profile, setProfile] = useState(null);
```

```
const [error, setError] = useState("");
useEffect(() => {
const controller = new AbortController();
async function loadProfile() {
try {
const response = await fetch(
`/api/tap/${encodeURIComponent(publicToken)}`,
{ signal: controller.signal }
);
if (!response.ok) {
throw new Error("Unable to load portfolio");
}
const data = await response.json();
setProfile(data);
} catch (err) {
if (err.name !== "AbortError") {
setError(err.message);
}
}
}
loadProfile();
return () => controller.abort();
}, [publicToken]);
if (error) return <p>{error}</p>;
if (!profile) return <p>Loading portfolio...</p>;
return (
<main>
<h1>{profile.name}</h1>
<p>{profile.department}</p>
<p>{profile.bio}</p>
<h2>Skills</h2>
<ul>
{profile.skills?.map((skill) => (
<li key={skill}>{skill}</li>
))}
</ul>
</main>
);
}
export default PublicPortfolio;
```

This sample demonstrates retrieving and displaying a public portfolio. In the actual application, the component should be connected to the configured API base URL and the existing routing, loading, error, and design components.

**Features Implementation**

**Sample Code:**

**1.Backend server**

```
const express = require("express");
const cors = require("cors");
const mongoose = require("mongoose");
require("dotenv").config();
const app = express();
app.use(cors());
app.use(express.json());
mongoose
.connect(process.env.MONGODB_URI)
.then(() => console.log("MongoDB connected successfully"))
.catch((err) => console.error("MongoDB connection error:", err));
app.get("/api/health", (req, res) => {
res.json({
success: true,
message: "TapConnect API is running"
});
});
// Register your existing application routes here.
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => {
console.log(`TapConnect server running on port ${PORT}`);
});
```

**2.Student profile model**

```
const mongoose = require("mongoose");
const userSchema = new mongoose.Schema(
{
name: { type: String, required: true },
email: {
type: String,
required: true,
unique: true
},
role: {
type: String,
enum: ["student", "admin", "super_admin"],
default: "student"
},
department: String,
bio: String,
```

```
skills: [String],
projects: [String],
certificates: [String],
resume: String,
github: String,
linkedIn: String,
portfolio: String
},
{ timestamps: true }
);
module.exports = mongoose.model("User", userSchema);
```

**3. Skill matching**

```
function compareSkills(viewerSkills = [], targetSkills = []) {
const normalize = (skills) =>
skills
.filter((skill) => typeof skill === "string")
.map((skill) => skill.trim())
.filter(Boolean);
const viewer = normalize(viewerSkills);
const target = normalize(targetSkills);
const viewerSet = new Set(
viewer.map((skill) => skill.toLowerCase())
);
const targetSet = new Set(
target.map((skill) => skill.toLowerCase())
);
return {
commonSkills: target.filter((skill) =>
viewerSet.has(skill.toLowerCase())
),
viewerUniqueSkills: viewer.filter(
(skill) => !targetSet.has(skill.toLowerCase())
),
targetUniqueSkills: target.filter(
(skill) => !viewerSet.has(skill.toLowerCase())
)
};
}
module.exports = compareSkills;
```

**4. NFC portfolio lookup**

```
const express = require("express");
const router = express.Router();
const NFCCard = require("../models/NFCCard");
```

```
const User = require("../models/User");
router.get("/:publicToken", async (req, res) => {
try {
const card = await NFCCard.findOne({
publicToken: req.params.publicToken,
status: "active"
});
if (!card) {
return res.status(404).json({
message: "Active NFC card not found"
});
}
const student = await User.findById(card.student).select(
"name department bio skills projects certificates github linkedIn portfolio"
);
if (!student) {
return res.status(404).json({
message: "Student profile not found"
});
}
return res.json({
success: true,
profile: student
});
} catch (error) {
return res.status(500).json({
message: "Unable to retrieve portfolio"
});
}
});
module.exports = router;
```

**5. React public portfolio**

```
import { useEffect, useState } from "react";
export default function PublicPortfolio({ publicToken }) {
const [profile, setProfile] = useState(null);
const [error, setError] = useState("");
useEffect(() => {
const controller = new AbortController();
async function fetchProfile() {
try {
const response = await fetch(
`/api/tap/${encodeURIComponent(publicToken)}`,
{ signal: controller.signal }
```

```
);
if (!response.ok) {
throw new Error("Unable to load student portfolio");
}
const result = await response.json();
setProfile(result.profile);
} catch (err) {
if (err.name !== "AbortError") {
setError(err.message);
}
}
}
fetchProfile();
return () => controller.abort();
}, [publicToken]);
if (error) return <p>{error}</p>;
if (!profile) return <p>Loading portfolio...</p>;
return (
<main className="mx-auto max-w-3xl p-6">
<h1 className="text-3xl font-bold">{profile.name}</h1>
<p>{profile.department}</p>
<p className="mt-4">{profile.bio}</p>
<h2 className="mt-6 text-xl font-semibold">Skills</h2>
<ul className="list-disc pl-6">
{profile.skills?.map((skill, index) => (
<li key={`${skill}-${index}`}>{skill}</li>
))}
</ul>
<h2 className="mt-6 text-xl font-semibold">Projects</h2>
<ul className="list-disc pl-6">
{profile.projects?.map((project, index) => (
<li key={`${project}-${index}`}>{project}</li>
))}
</ul>
</main>
);
```

**Administrative Module Implementation**

The administrative module provides student management, NFC card management, and activity monitoring. Department administrators operate within their assigned departments, while the super administrator has platform-wide administrative responsibilities.

The backend validates the user's role and department scope before allowing sensitive operations. Bulk import functionality processes student records from CSV or XLSX files, validates required fields, and reports unsuccessful records for correction.

Administrative activity can be recorded in the activity log to support auditing and dashboard analytics.

### 5.6 RESULTS AND DISCUSSION

TapConnect is designed to integrate digital portfolio sharing, AI-assisted career development, and student skill matching into a unified application. The following outcomes describe the intended system behaviour and the areas that should be demonstrated during project testing. Actual pass results should be recorded only after the corresponding functions have been tested.

**Functional Results**

| Module | Expected outcome |
|---|---|
| User Authentication | Users can log in and access permitted features according to their roles. |
| Student Profile | Students can manage their professional information and skills. |
| NFC Card Access | Tapping an active NFC card opens the associated public portfolio. |
| QR Code Access | Scanning a valid QR code opens the corresponding portfolio. |
| AI Center | Supported AI features return career-related feedback and recommendations. |
| Skill Matching | The system identifies common and unique skills between two profiles. |
| Department Admin | Administrators can manage students and cards within their departments. |
| Super Admin | The super administrator can manage department administrators and view platform-level activity. |
| Activity Analytics | Supported interactions are recorded and presented in activity dashboards. |

**User Interface Results**

The user interface is organized into role-specific dashboards. Students can access their profiles, AI tools, and networking features. Department administrators can manage student accounts and NFC cards, while the super administrator can monitor the overall platform.

The public portfolio provides a convenient way to share professional information through NFC and QR codes. Responsive layouts help users access the application from desktop computers and compatible mobile devices.

**AI and Skill Matching Results**

The AI Center is intended to help students improve their professional presentation by generating summaries, reviewing resumes, recommending projects, and analysing skills.

The skill matching feature supports peer discovery by comparing a logged-in user's skills with the skills listed in another student's profile. Presenting common skills and complementary capabilities can help students identify shared interests and potential opportunities for academic cooperation.

The quality of AI-generated output depends on the information provided, the configured Gemini model, API availability, and the prompts used by the application.

**Testing and Validation**

The following test cases can be used to validate the application before final submission.

| Test case | Expected result | Actual status |
|---|---|---|
| Valid user login | User is authenticated | To be tested |
| Invalid login credentials | Access is rejected | To be tested |
| Student profile update | Updated information is saved | To be tested |
| Active NFC card tap | Correct portfolio is displayed | To be tested |
| Valid QR scan | Correct portfolio is displayed | To be tested |
| Skill comparison | Common and unique skills are returned correctly | To be tested |
| Unauthorized admin request | Access is denied | To be tested |
| AI request | AI-generated response is returned when the service is available | To be tested |
| Activity recording | Relevant event is stored and displayed | To be tested |

These tests should be executed against the integrated application. The actual status column can be updated to **Pass** or **Fail** after each test is completed.

**Output**

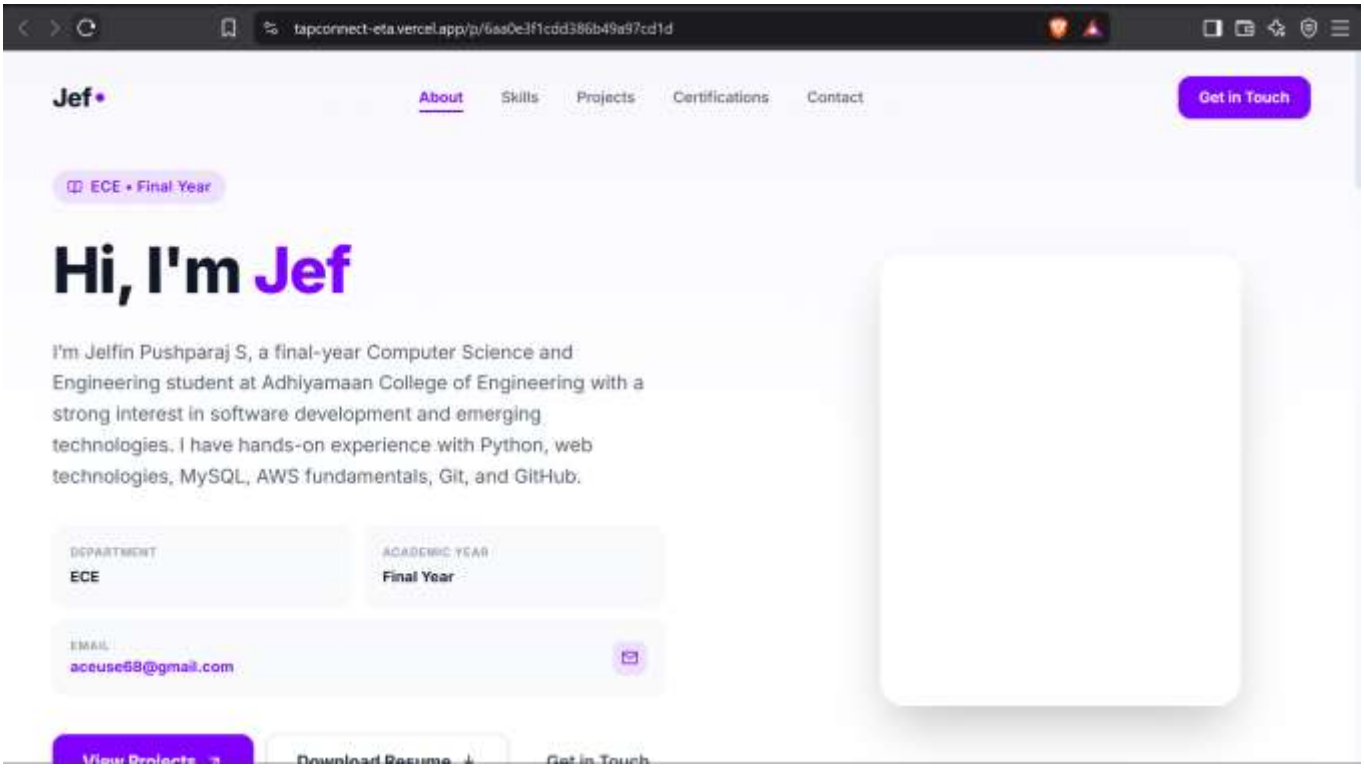



**Figure 5.1 Student Portfolio**

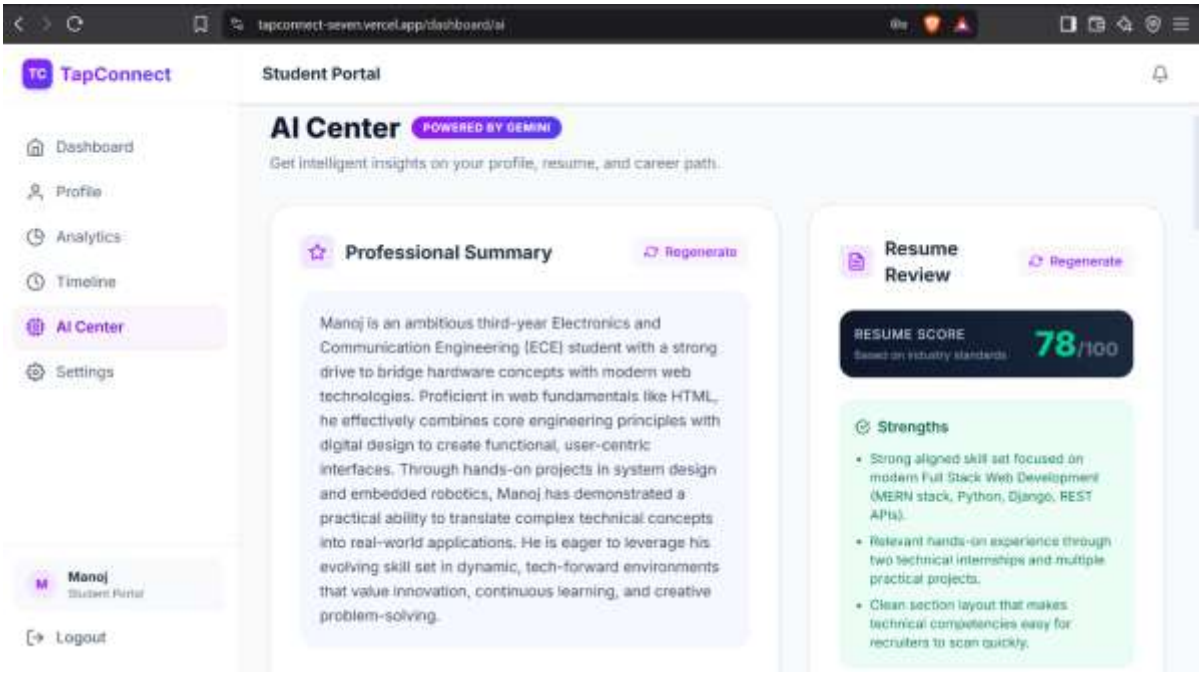



**Figure 5.2 AI Center**

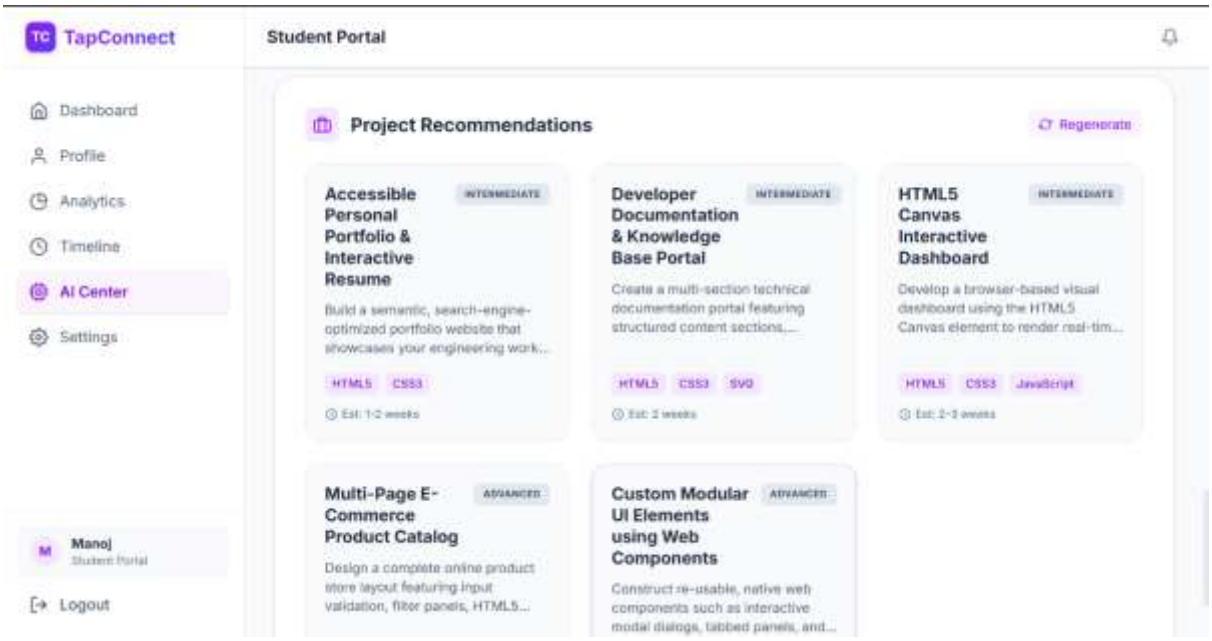



**Figure 5.3 AI Project Recommendation**

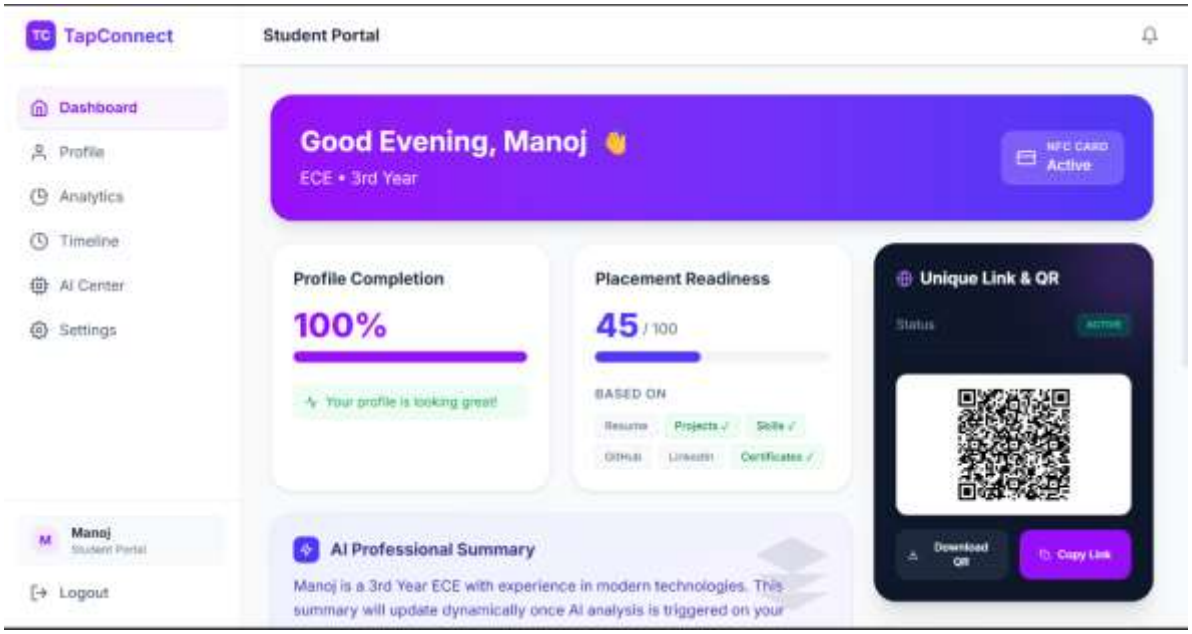



**Figure 5.4 Student Dashboard**

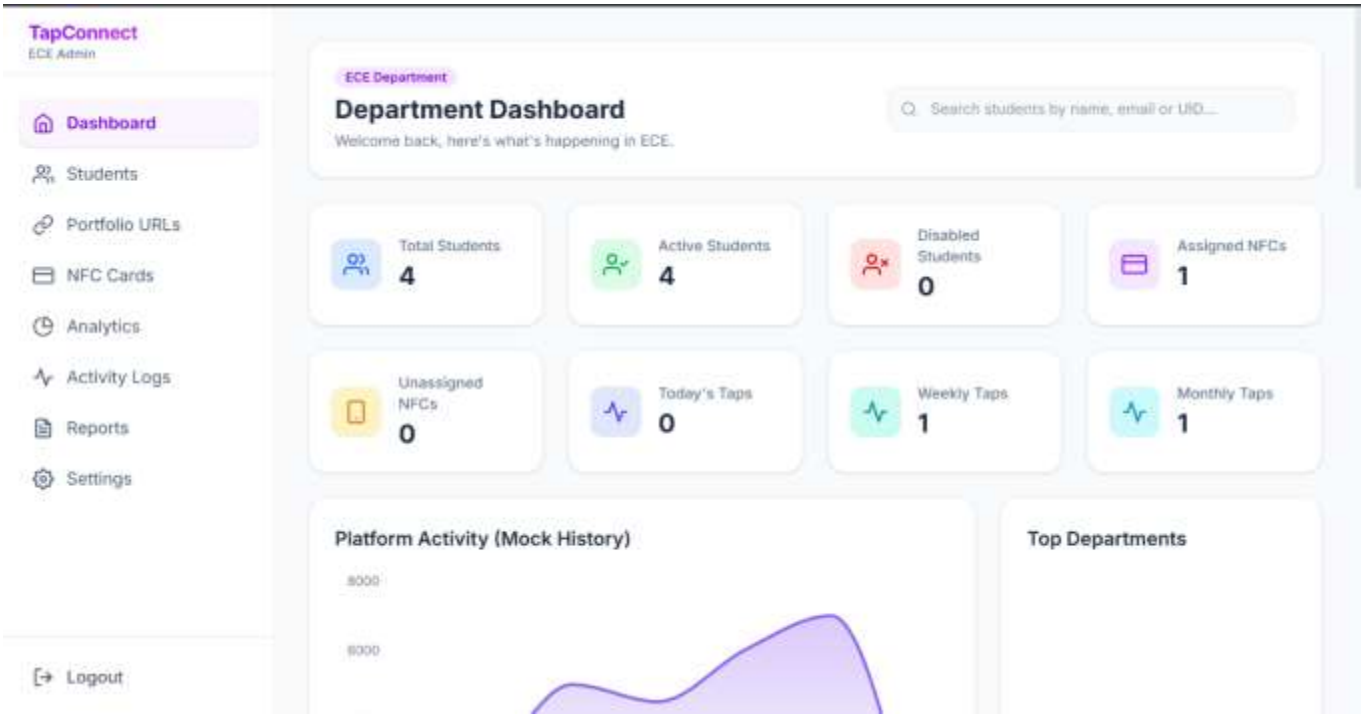



**Figure 5.5 Admin Dashboard**

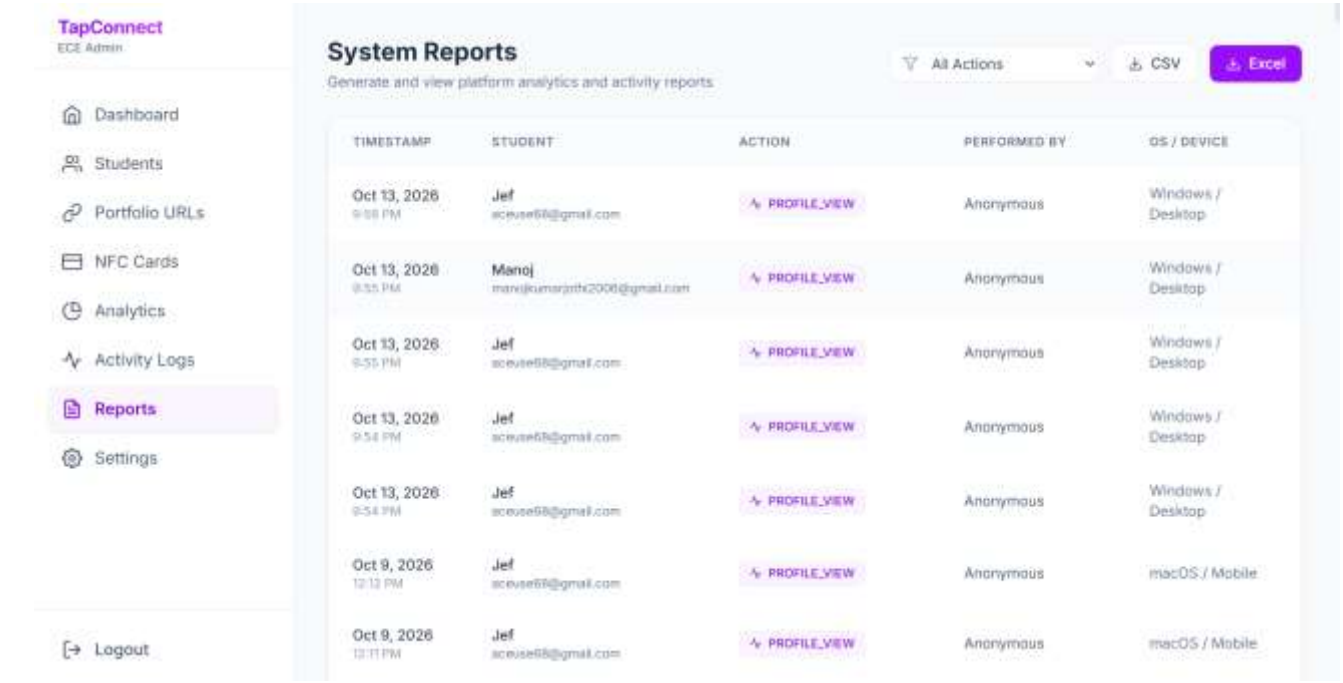



**Figure 5.6 System Reports**

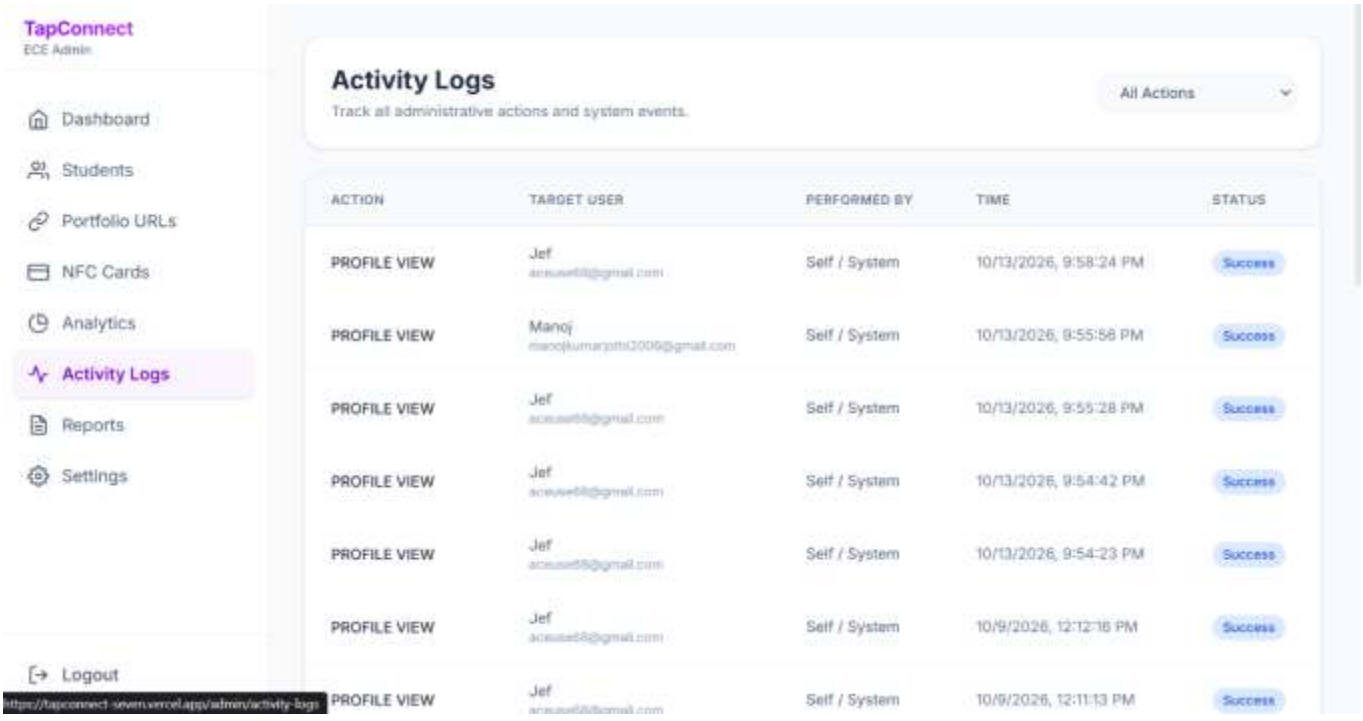



**Figure 5.7 Activity Logs**

## 5.7 SUMMARY

This chapter describes the implementation approach and expected results of TapConnect. The frontend provides the interface for student profiles, public portfolios, dashboards, and AI features. The backend handles authentication, database operations, profile retrieval, administrative permissions, and communication with the AI service.

The system combines NFC and QR code access with digital portfolio management and AI-based skill analysis. Once the proposed test cases are executed and the results are recorded, the completed evaluation

will demonstrate the reliability and functional suitability of TapConnect for student professional networking.

## 6. CONCLUSION

TapConnect is an AI-powered digital professional networking platform designed to simplify professional profile sharing and skill-based networking among college students. The system integrates Near Field Communication (NFC) technology, QR codes, digital portfolios, and Artificial Intelligence to provide a convenient and centralized platform for students to present their skills, academic projects, certificates, and professional achievements.

The application provides separate interfaces for students, department administrators, and the super administrator. Students can maintain their digital profiles and access AI-powered features such as professional summary generation, resume review, project recommendations, and skill analysis. NFC cards and QR codes enable quick access to public portfolios, reducing the need for traditional printed visiting cards and resumes. The administrative modules support student account management, NFC card management, and activity monitoring.

The integration of Google Gemini API enables intelligent analysis and personalized career-related suggestions, while the skill matching module helps identify common and complementary skills between students. The use of React.js, Node.js, Express.js, and MongoDB Atlas provides the technological foundation for developing and managing the application.

Overall, TapConnect aims to encourage professional networking, improve students' digital presence, and help them discover peers with relevant skills and interests. The project demonstrates how NFC technology and Artificial Intelligence can be combined to support student networking and career development in an academic environment. Future enhancements may include improved networking recommendations, event-based connections, internship opportunity matching, and expanded analytics to provide students with more opportunities for professional growth.

### 6.1 FUTURE SCOPE

Furthermore, the future development of TapConnect can focus on improving scalability, accessibility, and personalization to meet the evolving needs of students and educational institutions. The platform can be extended to support multiple colleges, allowing students from different institutions to discover professional connections and share their expertise.

Additional features can also be introduced based on feedback from students, faculty members, and administrators. These improvements can make TapConnect more adaptable to changing technologies and professional networking requirements.

### 6.2 ADVANCED AI-BASED RECOMMENDATIONS

In addition, machine learning techniques can be incorporated to improve the accuracy and relevance of recommendations. The system could analyse students' learning interests, completed projects, certifications, and preferred career paths to suggest suitable technical courses, workshops, and project ideas. Recommendations could be updated as students acquire new skills or modify their career goals. This would provide a more personalized experience and encourage continuous learning and professional growth.

### 6.3 INTERNSHIP AND JOB OPPORTUNITY MATCHING

The proposed opportunity-matching feature could also include notifications about application deadlines, eligibility criteria, required qualifications, and recruitment processes. Students could receive suggestions

based on their current skill levels and identify the competencies they need to improve before applying. Integration with verified job portals and company recruitment platforms could provide access to a wider range of opportunities. This enhancement could help students prepare more effectively for internships, placements, and entry-level employment.

### 6.4 ENHANCED PROFESSIONAL NETWORKING

Future versions of TapConnect could provide a more interactive networking environment by allowing users to create professional connections and communicate through the platform. Students could follow peers with similar interests, exchange project ideas, and form teams based on their technical strengths. Networking groups could be organized around specific domains such as artificial intelligence, web development, cybersecurity, and data science. Such features could encourage knowledge sharing and help students build stronger professional relationships.

### 6.5 EVENT-BASED NETWORKING

Event-based networking could be further enhanced by introducing digital attendee directories, event-specific profiles, and post-event connection suggestions. Organizers could create event pages where participants can discover attendees according to their skills and interests. NFC cards and QR codes could simplify profile sharing during technical events, reducing the need for manual exchanges of contact information. After an event, participants could revisit profiles and identify potential collaborators for future projects.

### 6.6 ADVANCED SKILL ANALYSIS

The skill analysis feature could be expanded to evaluate the relationship between students' existing competencies and the skills required for specific academic projects or career roles. The system could present skill-gap reports and recommend suitable courses, practice activities, or project experiences to address identified gaps. Students could also maintain updated skill records as they complete new projects and certifications. This would help them track their professional development and prepare more effectively for future opportunities.

### 6.7 ALUMNI NETWORKING AND MENTORSHIP

The alumni module could be developed to support structured mentorship programmes between current students and graduates. Alumni could share career experiences, provide technical guidance, and offer suggestions for improving resumes and interview preparation. Students could search for mentors according to industry, professional role, technical expertise, and career interests. Such interactions could strengthen relationships between educational institutions and their alumni while providing students with practical career insights.

### 6.8 MOBILE APPLICATION DEVELOPMENT

A dedicated mobile application could provide a more convenient way to access TapConnect features while attending classes, workshops, seminars, and placement activities. The application could include profile editing, QR code scanning, notifications, and quick access to professional portfolios. NFC functionality could be supported on compatible devices, subject to hardware and operating-system restrictions. Offline access to selected profile information could also be explored, with appropriate privacy and synchronization controls.

### 6.9 IMPROVED ANALYTICS AND REPORTING

Advanced analytics could provide visual reports on profile engagement, frequently listed skills, popular project domains, and student participation in networking activities. Department administrators could use aggregated insights to identify areas where additional training or technical workshops may be beneficial.

Reports could also help institutions evaluate participation in career-development activities over time. Appropriate privacy safeguards and access permissions should be maintained to ensure that analytics are used responsibly.

### 6.10 ENHANCED SECURITY AND PRIVACY

Security improvements could include multi-factor authentication, session management, configurable profile visibility, and more detailed administrative audit logs. Students could be given greater control over which details are publicly displayed when their NFC cards or QR codes are scanned. Additional safeguards could protect uploaded resumes, certificates, and other documents from unauthorized access. Regular security testing, input validation, and monitoring could further improve the reliability and trustworthiness of the platform.

### 6.11 INTEGRATION WITH EXTERNAL PLATFORMS

Integration with external platforms could reduce the need for students to maintain professional information manually across multiple services. Subject to the availability of authorized APIs, TapConnect could retrieve or link relevant information from professional networking websites, learning management systems, certification providers, and recruitment portals. Students could use these integrations to keep their portfolios more complete and up to date.

## ACKNOWLEDGEMENT

The authors express their sincere gratitude to **Mrs. R. Vijayalakshmi**, Assistant Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering, for her valuable guidance, continuous support, and encouragement throughout the development of the TapConnect project.

## REFERENCES

1. P. Rahayu, D. I. Sensuse, B. Purwandari, I. Budi, F. Khalid, and N. Zulkarnaim, "A Systematic Review of Recommender System for e-Portfolio Domain," *Proceedings of the 5th International Conference on Information and Education Technology*, 2017. Available: https://doi.org/10.1145/3029387.3029420
2. A. José-García et al., "C3-IoC: A Career Guidance System for Assessing Student Skills using Machine Learning and Network Visualisation," *International Journal of Artificial Intelligence in Education*, vol. 33, pp. 1092–1119, 2023. Available: https://doi.org/10.1007/s40593-022-00317-y
3. H. Crompton and D. Burke, "Artificial Intelligence in Higher Education: The State of the Field," *International Journal of Educational Technology in Higher Education*, vol. 20, article 22, 2023. Available: https://doi.org/10.1186/s41239-023-00392-8
4. S. Algarni and F. Sheldon, "Systematic Review of Recommendation Systems for Course Selection," *Machine Learning and Knowledge Extraction*, vol. 5, no. 2, pp. 560–596, 2023. Available: https://doi.org/10.3390/make5020033
5. S. S. K. Khanal, P. W. C. Prasad, A. Alsadoon, et al., "A Systematic Review: Machine Learning Based Recommendation Systems for E-Learning," *Education and Information Technologies*, vol. 25, pp. 2635–2664, 2020. Available: https://doi.org/10.1007/s10639-019-10063-9
6. "A Comprehensive Study on Personalized Learning Recommendation in E-Learning System," *IEEE Access*, vol. 12, pp. 100446–100482, 2024. Available: https://doi.org/10.1109/ACCESS.2024.3428419
7. "A Systematic Review of e-Portfolio Use During the Pandemic: Inspiration for Post-COVID-19 Practices," *Open Praxis*, 2024. Available: https://doi.org/10.55982/openpraxis.16.3.656

8. M. Tavakoli, A. Faraji, J. Vrolijk, M. Molavi, S. T. Mol, and G. Kismihók, "An AI-Based Open Recommender System for Personalized Labor Market Driven Education," *Advanced Engineering Informatics*, vol. 52, article 101508, 2022. Available: https://doi.org/10.1016/j.aei.2021.101508
9. "Recommender System for University Degree Selection: A Socioeconomic and Standardised Test Data Approach," *Applied Sciences*, 2024. Available: https://doi.org/10.3390/app14188311
10. H. Reed, "Bridging the Gap Between Education and Employment: An AI-Integrated LinkedIn Networking Pedagogy for Business Students," *Journal of Higher Education Theory and Practice*, vol. 25, no. 5, 2025. Available: https://doi.org/10.33423/jhetp.v25i5.7952
11. "Bridging the Education–Employment Gap: Linking Labour Market Needs with University Programmes Through AI-Assisted Insights and Micro-Credentials," *Education Sciences*, 2026. Available: https://doi.org/10.3390/educsci16071156
12. "Preparing University Graduates for the Labour Market Through Employability Skills Development and University–Industry Collaboration: A Systematic Review," *Education Sciences*, 2026. Available: https://doi.org/10.3390/educsci16030426
13. "Implementing a Recommendation System to Predict Course Selection for Higher Education," *Digital Education Review*, 2026. Available: https://doi.org/10.1344/der.2026.48.224-238
14. "SynCard – Smart Digital Business Card," *International Journal of Engineering Development and Research*, vol. 13, no. 4, pp. 898–904, 2025. Available: https://rjwave.org/ijedr/papers/IJEDR2504356.pdf
15. "An Intelligent Digital Platform for Alumni Engagement and Networking," *International Journal for Multidisciplinary Research*, vol. 7, no. 6, 2025. Available: https://doi.org/10.36948/ijfmr.2025.v07i06.60965
16. "A Systematic Literature Review: Improving University Guidance Through AI-Based Recommendation Systems," *Open Information Science*, 2025. Available: https://doi.org/10.1515/opis-2025-0021
17. "AI-Based Chatbot for Student Skill Assessment, Career Guidance, and Job Recommendation," conference paper, 2026. Available: https://zenodo.org/records/18755072
18. "Design and Development of an AI-Based Career Navigator, Internship Finder and Document Reminder System," *International Journal of Versatile Research and Analysis*, 2026. Available: https://ijpub.org/ijvra/viewpaperforall.php?paper=IJVRA2608546
19. R. S. Raut et al., "UPLINK: A Review of Skill Mapping, Career Recommendation and Academia–Industry Collaboration for Campus Career Ecosystems," *International Journal of Multidisciplinary Allied Research Review and Practices*, 2026. Available: https://ijmarrp.in/article/uplink-a-review-of-skill-mapping-career-recommendation-and-academia-industry-collaboration-for-campus-career-ecosystems-hjsG
20. M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force, 2015. Available: https://www.rfc-editor.org/rfc/rfc7519