

**A**  
**Project Report**  
on  
**Uni-Link: A Centralized Placement  
Data Visualization Portal**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE  
REQUIREMENT OF THE AWARD OF DEGREE OF  
**BACHELOR OF TECHNOLOGY DEGREE  
IN  
COMPUTER SCIENCE**



Submitted by

SHIVAM KUMAR (2100290120157)  
SUDHANSHU SINGH (2100290120165)  
RAJ SINGH (2100290120135)  
VAIBHAB BISHT (2100290120181)

Supervised by  
Ms. SHIVANI  
Assistant Professor  
Session 2024-25

**DEPARTMENT OF COMPUTER SCIENCE  
KIET GROUP OF INSTITUTION, GHAZIABAD**

(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow,  
U.P., India)

May 2025

## DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Shivam Kumar

Roll No.: 2100290120157

Signature

Name: Raj Singh

Roll No.: 2100290120135

Signature

Name: Sudhanshu Singh

Roll No.: 2100290120165

Signature

Name: Vaibhav Bisht

Roll No.: 2100290120181

Date:

## **CERTIFICATE**

This is to certify that Project Report entitled **“UNI-LINK: A CENTRALIZED PLACEMENT DATA VISUALIZATION PORTAL”** which is submitted by **Raj Singh, Shivam Kumar, Sudhanshu Singh, Vaibhav Bisht** in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**Date:**

Ms. Shivani Assistant Professor

**Supervisor:**

Department of Computer Science

## ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to **Assistant Professor Ms. Shivani, Department of Computer Science, KIET, Ghaziabad**, for his/her constant support and guidance throughout the course of our work. His/her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of **Dr. Ajay Kr. Srivastava, Head of the Department of Computer Science, KIET, Ghaziabad**, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Name: Shivam Kumar  
Roll No.: 2100290120157

Name: Raj Singh  
Roll No.: 2100290120135

Name: Sudhanshu Singh  
Roll No.: 2100290120165

Name: Vaibhav Bisht  
Roll No.: 2100290120181

<b>TABLE OF CONTENTS</b>	<b>Page No.</b>
DECLARATION	ii
CERTIFICATE	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v-vi
ABSTRACT.....	vii
LIST OF FIGURES.....	viii
LIST OF TABLES.....	ix
 CHAPTER 1 INTRODUCTION	 1-7
1.1 Introduction to Project	1-2
1.2 Project Category	2
1.3 Objectives	2-4
1.4 Structure of Report	4-7
 CHAPTER 2 LITERATURE REVIEW	 8-14
2.1 Literature Review	8-11
2.2 Research Gaps	12-13
2.3 Problem Formulation	13-14
 CHAPTER 3 PROPOSED SYSTEM	 15-18
3.1 Proposed System	15-16
3.2 Unique Features of the System	16-18
 CHAPTER 4. REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION	 19-43
4.1 Feasibility Study (Technical, Economical, Operational)	19-20
4.2 Software Requirement Specification Document	21-25

4.3	SDLC Model to Be Used	25-30
4.4	System Design	30-43
CHAPTER 5. IMPLEMENTATION		44-50
5.1	Introduction to Languages, Tools and Technologies Used for Implementation	44-50
CHAPTER 6 TESTING, AND MAINTENANCE		51-58
6.1	Testing Techniques and Test Cases Used	51-58
CHAPTER 7. RESULTS AND DISCUSSIONS		59-63
7.1	Presentation of result	59-63
CHAPTER 8. CONCLUSION AND FUTURE SCOPE		64-66
8.1	Conclusion	64
8.2	Future scope	65-66
REFERENCES		67-68
Plag Report		69-71
Proof of patent publication (Screenshot of Publication)		72

## **ABSTRACT**

In today's job market, which is rather dynamic, evidence-based decision-making is critical to students, educational institutions, corporate recruiters, and policymakers. Uni-Link is a central repository that systematically aggregates a report of placement data from Indian higher educational institutions, providing a holistic and accessible platform for all stakeholders of the placement ecosystem. Through the aggregation of detailed data on student placements, leading recruiters, salary trends, industry-wise break-ups, and past placement trends, Uni-Link empowers users with high-value insights to analyze trends, identify new prospects, and make wise career and recruitment decisions. Education institutions can also leverage this data to enrich their curriculum, and improve training programs. Given its high-level insight into talent pools at the institutional level and placement trend analysis, uni link will help it to rationalize its recruitment processes. Besides corporate and institutional access, Uni- Link stands in the greater interest of policymakers. Policymakers can view large-scale placement data to trace industry trends, evaluate education policies' performance, and generate fact-based policy decisions to support better outcomes in higher education. Policymakers and governmental agencies can also seek information from Uni-Link about their analysis of the placement data for guiding their own decisions regarding workforce development programs and strategic planning in closing the gap of academic and industrial demands. With its data- driven central model, Uni-Link channels not only a positive recruitment but also strives toward a more open and informed placement setting. With its horse, a platform involving students, institutions, recruiters, and policy-makers, Uni-Link is charting a course for higher education and employment in India.

## **LIST OF FIGURES**

<b>Figure No.</b>	<b>Description</b>	<b>Page No.</b>
<b>3.1</b>	<b>Structure Diagram</b>	<b>18</b>
<b>4.1</b>	<b>Unilink Development Life cycle</b>	<b>22</b>
<b>4.2</b>	<b>0 Level DFD</b>	<b>32</b>
<b>4.3</b>	<b>1 Level DFD</b>	<b>33</b>
<b>4.4</b>	<b>2 Level DFD</b>	<b>34</b>
<b>4.5</b>	<b>Use-case Diagram</b>	<b>36</b>
<b>4.6</b>	<b>Placement Upload Flow</b>	<b>38</b>
<b>4.7</b>	<b>Verification Process Flow</b>	<b>39</b>
<b>4.8</b>	<b>Login Flow</b>	<b>40</b>
<b>4.9</b>	<b>ERD</b>	<b>43</b>
<b>7.1</b>	<b>Sign in Page</b>	<b>61</b>
<b>7.2</b>	<b>Role based Access</b>	<b>61</b>
<b>7.3</b>	<b>New Institute Setup form</b>	<b>62</b>
<b>7.4</b>	<b>New Institute Request to Super Admin</b>	<b>62</b>
<b>7.5</b>	<b>Government Dashboard</b>	<b>63</b>
<b>7.6</b>	<b>Data Representation</b>	<b>63</b>



## LIST OF TABLES

Table No.	Description	Page No.
2.1	Literature Survey	9-11
6.1.2.1	User Authentication and Role-Based Access	55-56
6.1.2.2	Organization Creation (Admin)	56
6.1.2.3	Placement Data Management	57
6.1.2.4	Chatbot based report query system	57-58
6.1.2.5	UI and Navigation	58

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Introduction**

Unilink, via the unilink platform, revolutionizes placement data management for colleges, corporate recruiters, and government agencies by automating primary functions like uploading student placement data, verifying placement records, and creating real-time reports and visualizations. this automation greatly minimizes the amount of manual effort on college placement cells without sacrificing accurate and systematic placement records for internal analysis, external corporate use, and centralized government monitoring.

Unilink promotes trust and transparency in the culture among educational institutions, corporate recruiters, and regulatory authorities. by aggregating placement data of various colleges on a single platform, corporate partners can obtain direct access to authenticated placement history and recruitment trends, and government authorities can create detailed reports and visualizations to track placement performance at various regions and institutions. The single platform facility enhances data reliability, reduces discrepancies, and facilitates planning for companies and policy making by government officials.

Unilink allows corporate recruiters to easily filter, search, and examine placement details without needing to deal with each college separately. Real-time confirmed records like company names, packages provided, and

branch-wise placement percentages become accessible so companies are able to make recruitment decisions faster and better. government officials are able to see overall placement figures, gender-wise placement analysis, and geographic placement trends in an attempt to make evidence-based decisions on policies, identify areas for improvement, and make sure that the colleges are providing placements on a standards-based basis.

## **1.2 Project Category (Full Stack Web Development)**

Unilink, is a web development project built as a full-stack application with next.js as the frontend, prisma as the orm, and postgresql as the database. the project aims to make the centralised management, verification, and visualization of placement data in different educational institutions easy. it provides for the secure capture of placement records, including student details, companies they were placed with, packages provided, and placement status in different branches. with sophisticated data visualization capabilities and real-time reporting, unilink simplifies data management for college placement cells, genuine and trustworthy access to data assured for corporate recruiters, and enables government authorities to monitor placement trends and statistics at a consolidated level. this cutting-edge platform enhances transparency, accuracy, and accessibility of data, resulting in benefits to colleges, recruiters, and regulatory agencies in the education and corporate sector.

## **1.3 Objectives**

### **1.3.1 Centralized Placement Data Management**

The primary objective of UniLink is to establish a centralized database for the storage, management, and retrieval of placement data from numerous

educational institutions across the country. By consolidating placement records, including student profiles, offered packages, firms, and placement statuses, UniLink enables hassle-free data transfer among colleges, recruiters, and government agencies, and prevents fragmented and inconsistent record maintenance.

#### **1.3.2.1 Effective Verification and Accuracy of Data**

UniLink simplifies placement verification process through the capability of institutions to directly upload placement information while allowing third-party verification processes, including the provision of letter authentication or comparison of LinkedIn profiles. This makes data authentic and genuine, thus eradicating instances of fake or exaggerated placement records, thus ensuring trust among institutions, students, and recruiters.

#### **1.3.3.1 Real-Time Data Visualization and Reporting**

With inbuilt visualization features, UniLink allows stakeholders to access real-time reports and analytics of placement trends, student performance, and recruitment patterns. College administrators, recruiters, and government officials receive real-time information about placement results, gender balance, salary trends, and company-by-company recruitments, and take informed data-driven decisions at every level.

#### **1.3.3.2 Regulatory Supervision and Transparency**

UniLink encourages transparency and accountability from beginning to end of the placement process. The government agencies and the regulatory bodies share information of placements across institutions and may check compliance, track employment patterns, and assess the efficacy of

placement programs. Greater oversight makes a policy-based approach possible to enhance employability and learning.

## **1.4 Structure of Report**

This section is an overview of the report structure, stating the key content of each chapter. It is a brief description of the topic of each section, providing an idea of the information and areas of concern covered under the project report period.

### **Chapter 2 Literature Review**

The literature review examines existing research, technologies, and best practices in placement data management, recruitment portals, and verification processes in colleges and universities. It identifies the limitations of classical placement tracking methods, such as being typically disparate records, labor-intensive data input, and not real-time authentications. The chapter also highlights the growing place of centralized portals to increase the transparency, accuracy, and faith among students, colleges, recruiters, and governmental agencies. Research papers, industry reports, and case studies relevant to the conceptualization and system design of the UniLink system are included.

### **Chapter 3 Proposed System**

This chapter defines UniLink, its system architecture, components, and features. It describes how the suggested system meets the problems enumerated by the literature review, e.g., storage of placement information in an integrated manner, real-time checking mechanisms, and complete

analytics generation. The chapter also enunciates the substantial advantages to the stakeholders, e.g., colleges, students, recruiters, and government bodies, in terms of data transparency, operational efficiency, and real-time reporting. External placement cell integration, college ERP integration, and government portal integration are also mentioned.

#### **Chapter 4 Requirement Analysis and System Specification**

This section outlines the respective functional and non-functional requirements of Unilink with regards to role-based access, automated verification of placements, and real-time analytics. The proposed system will create a centralized reporting point for all verified placements while creating an open channel for the circulation of information between institutions and their staff, recruiters, and government authorities. With companies responsible for validating the placement records before publication, College Admins, Placement officers, Government officials, and Recruiters have the rights of access as set out by the system. The interactive dashboard displays trends in hiring and employment rates and helps cultivate data-informed decision-making. Unilink, being scalable, secure, and compliant with regulations, provides features for authentication, access control, and data privacy in order to create and maintain a secure and transparent placement management system.

#### **Chapter 5 Implementation**

The Implementation chapter discusses the data development life cycle of UniLink, consisting of planning, design, coding, integration, and deployment. This system uses Next.js at the frontend and Node.js with Express.js at the backend, while PostgreSQL has been adopted as the

primary database. The operation was based on the agile processes to make iterations in improvements. Some of important modules are automatic placement validation, role-based access control, and interactive analytics dashboards. This chapter will discuss some challenges such as reducing data consistency across institutions and optimizing query performance in PostgreSQL, and the solutions behind these challenges to make the ensuing systems more efficient.

## **Chapter 6 Testing and Maintenance**

This chapter outlines the testing methods used to ensure UniLink stability, precision, and security. Unit testing of individual modules, integration testing of data exchange between components, and user acceptance testing with college administrators and placement officers are covered. Test outcomes are documented, including system performance, data accuracy percentage, and users' feedback. The maintenance plan outlines ongoing system monitoring, frequent updating, and ways of collecting user remarks to allow for ongoing improvement and feature addition in future releases.

## **Chapter 7 Results and Discussions**

Results and discussions chapter presents the results and impact achieved with the design and deployment of UniLink. The chapter discusses how the system improved the placement data management process, data transparency, and reduced manual errors. Single view of placement data across institutions, real-time reporting capabilities, and automated verification process are evaluated based on effectiveness and user satisfaction. The chapter also presents lessons learned, including the need for data standardization, coordination among stakeholders, and regulatory

compliance. Future growth and new features are also emphasized.

## **Chapter 8 Conclusion**

The conclusion encapsulates the key success of the UniLink project and highlights its value in enabling placement transparency, data integrity, and institutional reputation. It considers the value created to educational institutions, students, recruiters, and regulatory authorities. The chapter also gives hints for future development, including AI-based fraud detection, predictive analytics for placement prediction, and more integration to government employment databases, enabling future innovation and scaling.



## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 Literature Review**

Dinesh G has suggested a centralized system to collect and verify placement records, which he considers an important aspect of recruitment placement dashboard [7]. Such a system would help institutions, organizations, and governments make data-driven decisions on the trends in student employability, as well as scout changes in recruitment strategy. This is in line with UniLink's vision of creating an interactive, validated placement database through which transparent analytics can be generated.

Deepmala Srivastava describes tools used for data visualization and classifies them into spreadsheets, specialized software, and programming libraries, pointing out the importance of those tools for trend analysis [5]. As far as Michael Stonebraker, Lawrence A. Rowe, and Michael Hirohama are concerned, PostgreSQL is an advanced database that provides rule-based triggers and time-travel storage, thus possessing advantages over traditional databases[3]. Ahmad Tasnim Siddiqui has gone through visualization techniques: Power BI and Tableau, discussing both scalability and accuracy problems encountered in their use[6]. This balancing of coherence and performance in synchronous implementation in PostgreSQL, MySQL, and SQL Server has been discussed by Ciprian-Octavian Truică, Alexandru Boicea, and Florin Rădulescu[4]. Tarun, Dr. Vishal Shrivastava, and Dr. Akhil Pandey have expressed views on Next.js with respect to SSR, SSG, dynamic routing, and the integration of APIs for performance and SEO advantages[2]. Antony Unwin distinguishes between exploratory

graphics and presentation graphics and emphasizes clarity of presentation in visualization processes [11]. The studies emphasize automation, real-time data handling, and role-based access that confirm the design of UniLink. These insights shaped the architecture of UniLink such that a complete balance is achieved between data visualization, database integrity, and system performance potentials for stakeholder efficiency in placement analytics.

<b>Sr. No.</b>	<b>Authors</b>	<b>Title of the Paper</b>	<b>Contribution to UniLink</b>
1	G. Dinesh, MNV. Surekha, et al.	Placements Analytics and Dashboard	This paper describes a centralized platform to collect, process, and analyze placement data, aligning directly with UniLink's purpose of aggregating institution-wide placement records. It Aalso contributes techniques for generating placement dashboards and analyzing placement trends, which are essential features in UniLink.
2	Deepmala Srivastava	An Introduction to Data Visualization Tools and Techniques in Various Domains	The paper provides an overview of visualization tools and techniques which directly supports UniLink's goal to offer real-time placement data visualizations for colleges, corporate recruiters, and government bodies.

3	Michael Stonebraker, Lawrence A. Rowe, Michael Hirohama	The Implementation of POSTGRES	This foundational paper on PostgreSQL helps UniLink optimize its placement data storage, retrieval, and indexing strategies, ensuring efficient handling of large-scale placement data from multiple colleges.
4	Ahmad Tasnim Siddiqui	Data Visualization: A Study of Tools and Challenges	This paper highlights visualization challenges and tools, contributing to UniLink's approach to building effective dashboards and visual reports that present placement trends in a user-friendly format.
5	Ciprian-Octavian Truică, Alexandru Boicea, Rădulescu	Asynchronous Replication in Microsoft SQL Server, PostgreSQL and MySQL	This paper contributes to UniLink's backend architecture by offering strategies for PostgreSQL replication, ensuring reliable data consistency across distributed systems
6	Tarun, Dr. Vishal Shrivastava, Dr. Akhil Pandey	Introduction on Next.js and its Benefits and Analysis	This paper explains Next.js benefits, directly supporting UniLink's technology choice for the frontend, especially in delivering fast-loading dashboards and public reports.
7	Antony Unwin1	Why is Data Visualization Important? What Is	This paper emphasizes the importance of clear, purposeful visualizations in decision-making

		Important in Data Visualization?	processes. It reinforces UniLink's goal to create clean, easy-to-understand visual dashboards for corporates and government agencies, allowing them to monitor placement trends and institution performance at a glance.
8	Chander Velu Stuart Madnick Marshall Van Alstyne	Centralizing Data Management with Considerations of Uncertainty and Information based Flexibility	This paper highlights the challenges of centralizing data from diverse sources, focusing on handling incomplete data and ensuring flexibility for evolving requirements. It supports UniLink's goal of standardizing placement data from different colleges and building a flexible system that can adapt to changes in formats and reporting needs over time.
9	Mohhammad Fariz Syah Lazuardy, Dyah Anggraini	Modern Front End Web Architectures with React.Js and Next.Js	The paper focuses on central platforms improving placement transparency, which supports UniLink's goal of ensuring transparent placement tracking across institutions, companies, and regulators.

Table 2.1: Literature Survey

## **2.2 Research Gaps**

UniLink is closing some of the present research gaps in centralized placement data management and validation through the higher educational institutions in India:

### **2.2.1 Absense of centralized system**

Placement Management Systems usually work in isolation within disparate colleges with no centralized system to consolidate data across institutions. This fragmentation causes improper data comparisons, challenges conducting studies across institutions, and impedes corporates and government agencies with complete placement information. UniLink thus resolves this gap by providing a single platform where placement records of all registered institutions are being collected, standardized, and shared with authorized bidders.

### **2.2.2 Limited Data Verification Mechanisms**

Placement reporting systems are generally dependent upon self-reported information from institutions with no externally verified source, raising questions about their credibility. Responses to claims made by institutions regarding placement are overwhelmingly in a direct placement proof format in the form of graduates being placed and placing claims by companies and government agencies through a multi-stage verification procedure within UniLink. Hence, the integrity of data increases, and chances of false reporting decrease.

In general placement reporting systems are based on self-reported information without any external verification, which leaves a lot to be desired in terms of credibility. Placement data is subject to an

overwhelmingly direct placement proof format in the form of graduates being placed and placing claims by companies and government agencies through a multi-stage verification procedure within UniLink. Hence, the integrity of data increases, and chances of false reporting decrease.

### **2.2.3 Absence of Visual Analytics for Stakeholders**

Legacy Placement reports are themselves more like static tables and papers, limiting the scope of dynamic exploration and analysis of trend movement. This is what UniLink intends to overcome through interactive dashboards and real-time data visualization to allow institutions to monitor their placement performance, corporates to evaluate talent pools by region, and government agencies to track national placement trends for policy-making.

### **2.2.4 Data-Driven Policy Formulation**

UniLink's unified database and analytical resources give governments placement data that are accurate and real-time across the nation, allowing the development of evidence-based policies to enhance employability and to mesh curricula with industry requirements.

## **2.3 Problem Formulation**

The system being developed, unilink, is affordable for educational institutions, government departments, and corporate clients as it saves placement tracking and paperwork. With placement data being integrated from various institutions and placed all on a singular platform, the administrative burden for the schools has been considerably lowered; hence, Unilink provides a way for the institutions to upload placements

through csv upload, reducing manual data entry operations and the errors that come with it.

In a time-efficient manner, by automating processing of placements, verification procedures, and report generation, the verification and validation processes are expedited. Institutions now have the option of bulk-uploading placement records of students for the system to automatically send out requests for verifications to the corresponding companies. This allows for instant validation and real-time updates. Built-in dashboards and visual reports for institutions, corporations, and government institutions provide instant insights without the need for manual report preparation.

Unilink is a cost-effective solution in that it does not require any specialized hardware or expensive proprietary software. Unilink is based on Next.js technology on the client-side and Prisma with Postgres as the database on the server-side, using standard web infrastructure, which is already present in the colleges.

The system has an easy-to-use and intuitive user interface, allowing placement officers, corporate clients, and government administrators to easily navigate placement information, verification status, and analytics dashboards. Limited training for the users is required, as the structure of the platform and the workflows are based on the standard web application structure, thereby making the system functional for users with varying levels of technical ability.

The unilink platform's responsiveness makes it possible to use on laptops, desktops, tablets, and mobile phones so that users can track and control placement activities from anywhere.

## **CHAPTER 3**

### **PROPOSED SYSTEM**

#### **3.1 Proposed System**

UNILINK is suggested as a web-based centralized system through which placement data management, verification, and analysis can be facilitated for Indian higher educational institutions. Access is easy via commonly used web browsers by our users.

The institutions will batch-upload the placement data regarding student information, company names, and placement status after which notifications will alert the companies to verify.

Real-time dashboards will also allow colleges to track placement performance, firms to monitor verified hires, and government bodies to view consolidated reports and trends at regional and national levels. Access control, based on roles, gives an option of restricting views to only stakeholders thereby ensuring privacy and protecting the data.

Having a centralized data repository enables the avoidance of data duplication through a similar method of handling data among member institutions. A further automated validation minimizes the risk of errors and ensures compliance with this schema upon upload of files.

UNILINK is built on Next.js, Prisma, and PostgreSQL, hence security, speed, and scalability are intact. Secure APIs facilitate data sharing amongst institutions, industries, government, and MIS for verification of placement records. Government dashboards also possess region-based, institution type-based, and industry sector-based customized filters to support



evidence-policy-making.

In short, UNILINK facilitates placement data management simplification, placement verification intensification, real-time insights, and ensuring transparency and accountability of system.

## **3.2 Unique Features of System**

### **3.2.1 An on-line system.**

The system is purely web-based and accessible by way of any web-enabled device via an internet connection, thereby permitting placement officers, corporate leaders, and government personnel to log on to the system in a secured mode from any possible location.

UNILINK is an application that runs on the web and has been developed using Next.js for the front-end, with prisma for the back-end data management based on Postgres. It doesn't need users to have special software installations, thus facilitating general access via ordinary web browsers across educational institutions, branches of corporate offices, and government agencies.

### **3.2.2 A universal common process.**

UNILINK has a uniform process for uploading, verifying, and accessing placement information, such that all the colleges that participate do so in the same format and workflow. Various levels of users — institution admins, corporate verifiers, and government users — have different permissions, but they all deal with the same, centralized data set. This allows for consistency in placement reporting within institutions, enabling

cross-institution comparison and analysis at the national level.

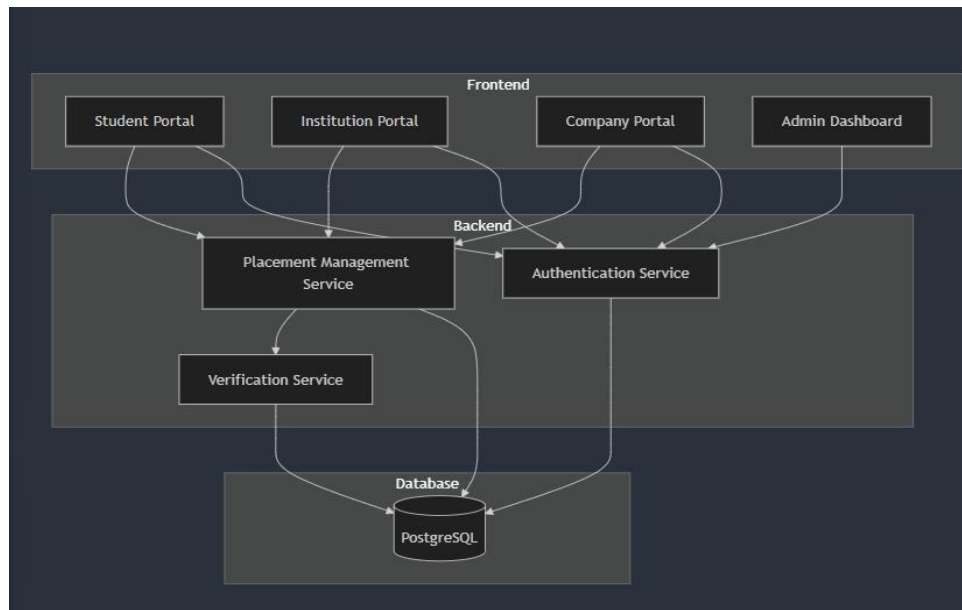
### **3.2.3 Track information.**

The platform has a rich history of all placement data, verification activities, and trends in placement over time. This enables institutions to compare performance, facilitates companies to monitor the history of hiring, and allows government agencies to produce time-series reports to aid policy analysis and intervention planning.

This structure diagram given below demonstrates the structure of the placement system in a layer-wise composition segregated into frontend, backend, and database components. The frontend contains four isolated portals—Student, Institution, Company, and Admin—each optimized for individual roles. These portals interact with backend services that are listed as below:

- Authentication Service for secure login and role authorization
- Placement Management Service to handle student placement information
- Verification Service for authenticating and approving records uploaded by companies.

These services are coordinated to provide data integrity and system security. Fundamentally, a PostgreSQL database holds all of the critical information which includes user credentials, placement details, and verification logs. The clear definition of tasks and modular structure allow for scalability, simplicity of maintenance, and secure access control at all user levels.



**Figure 3.1: Structure Diagram**

## **CHAPTER 4**

# **REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATIONS**

### **4.1 Feasibility Study**

#### **4.1.1 Economical Viability**

The Unilink system will prove cost-effective to all educational institutions, corporate recruiters, and government agencies. It will eliminate time and money expended on manual record keeping, paperwork, and offline communication between recruiters and colleges. The system automates placement data collection, verification, and reporting processes to minimize administrative overhead while saving considerable time and resources for all parties involved. The system streamlines placement workflows, minimizing human errors and providing accurate, real-time placement insights, avoiding investments in costly infrastructure.

#### **4.1.2 Technical Feasibility**

All these technologies are open source and greatly support UniLink's feasibility. Next.js supports the front end, while Prisma serves as the ORM backed by PostgreSQL for the database. The system is all-web-browser-based, with no extra hardware or propriety software required by colleges, corporates, or government officials, which makes its deployment and

scalability easy across multiple institutions.

#### **4.1.3 Operational Feasibility**

Unilink is designed as user-friendly as possible, built with modern web technology (next.js) and is responsive, intuitive, and easy to use on any device, including desktops, laptops, and tablets. Placement officers, company representatives, and government users can navigate through dashboards, verification requests, and data visualizations without too much training.

#### **4.1.4 Behavioral Feasibility**

The UniLink system is designed for ease of use with a simple, user-friendly interface that requires almost no technical expertise on the part of the user. Placement officers, HR teams, and government officials can conduct all placement functions of uploading data, tracking verification, and generating reports directly through the platform without special-purpose training. With status indicators, notifications, and real-time updates, everyone is kept well-informed, allowing for easy acceptance across all stakeholders.

### **4.2 Software Requirement Specification**

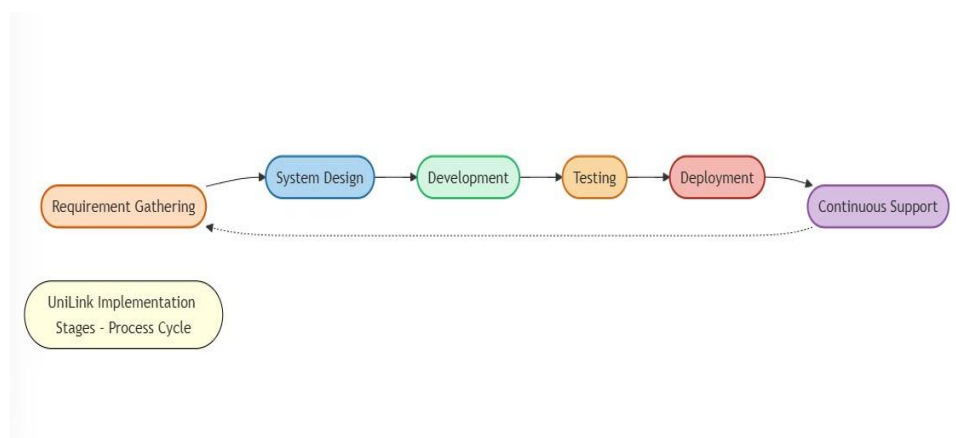
Very briefly, UniLink stands for a platform that centralizes information regarding placements from numerous colleges in India. It is the medium that connects students, colleges, and recruiters to observe and analyze placement trends in real time and compare these among colleges. The theory

is to be aimed at an orderly and transparent recruitment process.

The UniLink process requires thorough planning, configuration, and deployment such that all stakeholders concerned are considered, colleges uploading placement information, companies viewing candidate profiles, and students accessing placement information. Some features of the system include data acquisition from institutions, validation of placement claims, graphical representation, and secure access for users. All the requirements are clearly captured from all stakeholders to ensure that the platform supports real processes for usability and reliability.

Before deployment, all testing is done well to ensure that data integrity is present and functioning properly, as well as access control. The staggered rollout approach serves as a fair and even-handed inclusion for colleges and companies, as well as documentation and support to be offered to end-users.

Centralized management of placement data through UniLink would ensure that an open placement ecosystem is created for all institutions involved.



**Figure 4.1: UNILINK Development Lifecycle**

The phases are:

- I.      Requirement Gathering  
Determine system needs from students, companies, and colleges. Specify data formats, roles, and validation processes.
- II.     Design  
Design user interfaces, data streams, dashboards, and verification processes. Create secure access and role-based process plans.
- III.    Development  
Implement the data upload modules, validation engine, visualization dashboards, and role-based portals for students, companies, and colleges.
- IV.    Testing  
Validate data upload, validation accuracy, user access control, concurrent loads performance, and overall platform performance.
- V.      Deployment  
Phase-wise rollout for colleges and businesses, followed by students. Proper documentation and initial support.
- VI.    Continuous Support  
Offer regular user training, technical support, and periodic updates based on user input and regulatory updates

#### **4.2.1 Data Requirement**

**Student Information:** Detailed student profiles, such as enrollment

number, expected graduation year, gender, department, student id.

**Placement Data:** For every student, company name, job title, package offered, offer date, joining date, and the uploaded copy of the offer letter (if available).

**College Information:** Information regarding involved institutions, like college name, address, city, state(province), website, institution id, authorization letter.

**Company:** Company details like college name, address, city, state(province), website, company id.

#### **4.2.2 Functional Requirement**

##### **Centralized Data Upload:**

Colleges upload placement data on the platform. The system parses, validates, and consolidates data into a centralized repository.

##### **Automated Data Validation:**

System cross-checks uploaded data with students' company regarding the placement details.

##### **Placement Visualization:**

Creates visually appealing graphs so that easy to interpret.

##### **Audit Trail:**

Every action—data upload, validation step, system modification, and user activity—is logged for transparency and future audits.

#### **4.2.3 Performance Requirement**

##### **Response Time:**

Normal operations (data search, report generation, and dashboard loading)



must complete within 3 seconds. Under high-load conditions, response time should not exceed 6 seconds.

**Data Processing Capacity:**

The system needs to process placement information for thousands of students in hundreds of institutions with effective batch processing and parallel validation features.

**Scalability:**

The platform should be horizontally scalable to accommodate growing participation from institutions nationwide and manage millions of records in the long run.

**Availability:**

Provide 99.5% system availability, backed by automated backups, failover processes, and geographically dispersed servers to avoid data loss and downtime.

#### **4.2.4 Maintainability Requirement**

**Modular Architecture:**

Independent modules for data capture, validation, visualization, user administration, and reporting allow individual upgrades and quicker bug fixes.

**Detailed Documentation:**

Database schemas, API documentation, data validation rules, and system workflows are all part of technical documentation to make maintenance and inducting new developers easy.

**Continuous Feedback Integration:**

Regular user feedback surveys must be used to shape future developments in enhancing usability, introducing new filters/reports, and accommodating changing placement requirements.

#### **4.2.5 Security Requirement**

##### **Data Encryption:**

Transit and storage data must be encrypted with industry-standard encryption to safeguard sensitive student and recruiter data.

##### **Role-Based Access Control (RBAC):**

Strict access rules allow colleges only to see and administer their own data, and recruiters only view anonymized profiles until verification has been performed.

##### **Two-Factor Authentication (2FA):**

Delicate operations like data submission and verification approval must be made to demand 2FA to improve system security.

##### **Audit Trails & Activity Logs:**

There should be a full record of data uploads, changes, validation attempts, and user logins, which can assist in detecting any unauthorized or suspicious activity.

##### **Regulatory Compliance:**

The system will need to meet India's Digital Personal Data Protection Act (DPDP Act) and, if applicable, international requirements such as GDPR, particularly if student data is accessed by recruiters from abroad.

### **4.3 SDLC Model Used**

For the development of UniLink, the Prototyping Model of the Software Development Life Cycle (SDLC) was employed. The model was used since UniLink required much stakeholder input and constant fine-tuning, as it would process sensitive and institution-specific placement information from colleges throughout India. The Prototyping Model enabled developers

to develop an initial working prototype of the platform, receive feedback, and make incremental improvements to the system until it met user expectations and system requirements.

The Prototyping Model, as it was applied to UniLink, proceeded through these main phases:

## **I. Requirements Gathering**

During this phase, initial requirements were collected from colleges, placement officers, corporate recruiters, and government agencies. These were:

- Data fields to be collected (student, company, placement status, etc.).
- Verification processes required, like institute documents and company confirmations.
- Important performance indicators and dashboards are needed for colleges and government users.
- Access levels and needed role-based views.
- Security, audit, and compliance needs.

As placement procedures are different in every institution, preliminary feedback pointed toward the requirement for upload template customizability and adaptable validation rules, which were documented for the first prototype.

## **II. System Design**

At the design level, the technical specification for UniLink was produced, including:

- System architecture: Cloud-based, modular, and API-oriented for enabling seamless integration of data and extensibility in the future.
- Database design: Designed to accommodate hierarchical placement data, validated status tracking, and audit trails.
- User Interface (UI): role-based intuitive user interfaces for admins, recruiters, colleges, students, so intuitive, transparent and easy to use.
- Verification processes: automatic data checks against LinkedIn, parsing offer letters using optical recognition technology, and in-case recruiter acknowledgements.
- Dashboard UI: Live placement trend, recruiter engagement, verification status dashboards across colleges.

Security infrastructure: Multi-stage protection of data by encryption, access, and audit.

### **III. Development**

Development concentrated on constructing UniLink in modular parts to enable autonomous development and release of:

- Data Upload Module: Enabling bulk uploading of placement data by colleges and monitoring the status of the upload.
- Verification Engine: Facilitating automated verification of LinkedIn, parsing of documents, and sending confirmation requests for companies.
- Dashboard & Reports Module: Providing live reports on placement performance and verification status.
- User & Role Management Module: Facilitating secure login for colleges, recruiters, and students.

- Notification System: Alerting users automatically for errors, outstanding verifications, and system upgrades.
- Audit & Compliance Module: Logging all actions performed by the user to facilitate transparency and accountability.

Development proceeded under agile paradigms with feedback cycles that repeated regularly in order to iteratively hone features against stakeholder feedback.

## **IV Testing**

A thorough test drive ensured that UniLink conformed to all the functional, performance, and security requirements:

- Functional Testing: Correct behavior being checked in every critical module — uploads, verification, reporting, and notifications.
- Integration Testing: Validating smooth data exchange between modules and external APIs like LinkedIn, email.
- Performance Testing: Mocking up large data imports from many colleges to check for real-time dashboard refreshes and processing performance.
- Security Testing: Performing vulnerability scans, role-based access checks, and data encryption tests.
- User Acceptance Testing (UAT): Testing the platform with typical placement officers, recruiters, and students, verifying interface and functionality meet their workflows.

## **V. Deployment**

The roll-out of UniLink was in phases to guarantee stability and readiness

of users. A prototype of UniLink was first rolled out to a selected number of colleges and stakeholders to obtain initial feedback and determine possible problems. Through the prototype, users were able to pilot core functionalities, including data upload, placement verification, and report generation, to allow the development team to learn about usability gaps and functional enhancements

Based on the feedback collected during this pilot phase, the platform was refined to enhance performance, usability, and data accuracy features. The user interface (UI) was also reworked to offer a more intuitive and user-friendly experience for students, colleges, and recruiters.

Once the platform had satisfied all functional, performance, and security standards, UniLink was officially rolled out for broader adoption in educational institutions and corporate partners. Extensive onboarding assistance was offered to make it easy for users to shift onto the platform, with training sessions and support guides made accessible to enable users to familiarize themselves with the system.

## **VII. Continuous Support and User Empowerment**

After the implementation of UniLink, continuous support and user enablement facilitated smooth functionality and user satisfaction.

- **User Training:** Thorough sessions and user manuals were offered to recruiters, colleges, and students to familiarize them with the features of the platform, including data upload, placement verification, and access to the dashboard.
- **Helpdesk Support:** A special support team was provided to resolve users' technical problems, data uploads, and verification problems.
- **System Monitoring & Updates:** Continuous monitoring made sure platform operation, and frequent updates brought about new

features, enhanced security measures, and additional data validation tools.

- **Feedback & Enhancements:** Feedback from the users was gathered actively to help improve features and user experience constantly.
- **Compliance & Security:** Regular audits confirmed data integrity, system security, and compliance with changing legal and regulatory requirements.

## **4.4 System Design**

The System Design is an important part of UniLink development, in which each process — from collecting and verifying the data to reporting and controlling the access of the users — is well defined and logically organized. The system design process specifies the way data flows through different modules, how multiple users interact with the system, and how the data integrity and security are protected in all the transactions. Efficient system design facilitates seamless interaction of colleges, recruiters, students, and government agencies with the platform, reducing operational friction and enhancing transparency.

A number of industry-standard design diagrams were developed to depict UniLink's internal workings, data structures, and user interactions, which served as the basis for both backend development and frontend user interface.

#### **4.4.1 Data Flow Diagrams (DFD)**

Data Flow Diagrams (DFDs) graphically illustrate how placement data, verification requests, and system notifications move through UniLink. They give an end-to-end perspective on data movement, and it becomes easier to identify possible bottlenecks, redundant operations, or security threats. Through the mapping of external entities, processes, data stores, and flows, the DFD facilitates the conversion of user requirements into system logic that developers and stakeholders can comprehend and verify with ease.

##### **DFD Level 0 (Context Diagram)**

The Level 0 DFD, or Context Diagram, presents UniLink in its most abstracted, high-level form. The diagram depicts UniLink as one central process, directly dealing with outside organizations like Colleges, Companies, Students, and Government Authorities. It depicts only the major data exchanges — college placement data uploads, company verification responses, and government authority access to placement reports — without revealing internal procedures. This sets the boundaries of the system and its major inputs and outputs.



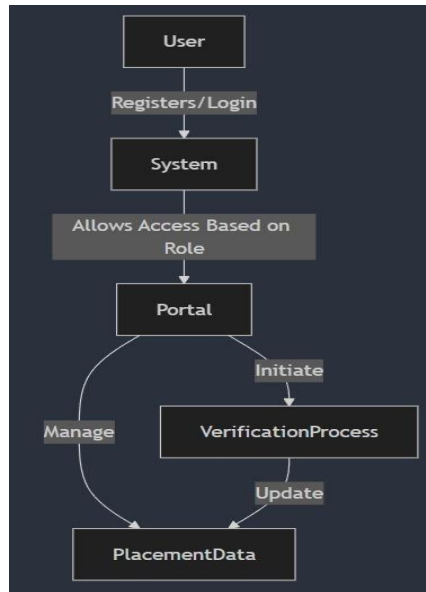


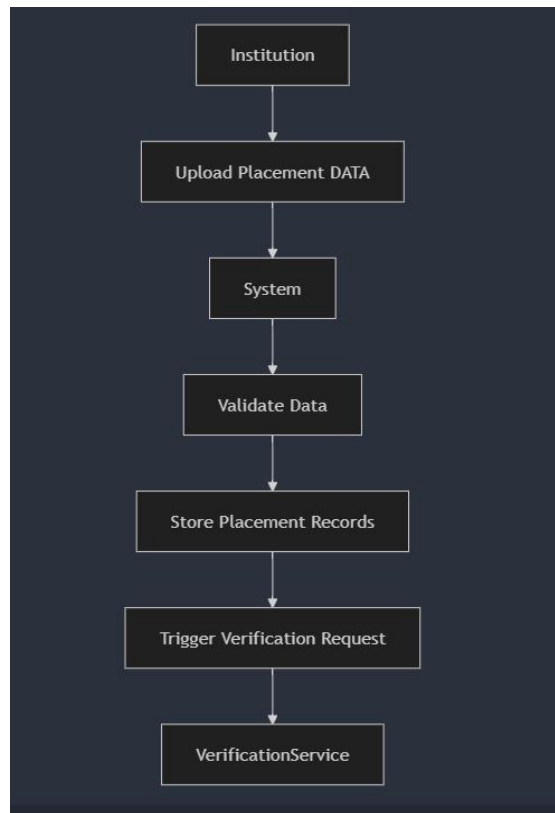
Figure 4.2: DFD Level 0

## DFD Level 1

Level 1 DFD develops the context diagram further by detailing major internal processes. This chart illustrates:

- Data Upload Process, through which colleges upload bulk placement records.
- Data Validation and Verification Process, in which uploaded data is automatically verified against LinkedIn profiles, offer letters, and company feedback.
- Reporting and Analytics Process, where authenticated data is compiled into real-time dashboards and institutional comparison reports.
- Notification Process, where automatic notifications are dispatched to colleges, students, and companies when events such as data uploads, verification requests, and status changes happen.

This diagram enables developers to see key functional blocks, ensuring modular and scalable development and unambiguous data handling flows.



**Figure 4.3: DFD Level 1**

## **DFD Level 2**

The Level 2 DFD extends the internal activities demonstrated in Level 1 further into separate processing steps.

For example, within the Verification Process, the Level 2 DFD would depict:

- Parsing uploaded data into structured records.
- Triggering LinkedIn profile checks for individual students.
- Generating automated verification requests sent to company HR

representatives.

- Flagging inconsistencies (such as missing fields, unmatched company names, or differences between uploaded and verified data).
- Logging the verification history and student's placement record update.

This level of detail is priceless for developers to create and implement each functional module with consistency to stakeholder requirements and regulatory compliance.



**Figure 4.4: DFD Level 2 ( Verification Request)**

#### **4.4.2 Use Case Diagram**

The Use Case Diagram specifies which users (actors) interact with which aspects of the system. It graphically displays:

- Institution Admins uploading placement data, verification statuses tracking, etc.
- Company Representatives examining student applications, confirming or denying placements, etc.
- Government Representatives obtaining aggregated placement statistics, looking at trend analyses, and creating policy-relevant reports.
- System Administrators administering user accounts, handling data conflicts, and tracking system health and audit logs.

The Use Case Diagram guarantees that every feature of the system is assigned to a particular user role, defining permissions and preventing unauthorized access to data. It is also an early point of reference for the definition of role-based access control (RBAC).

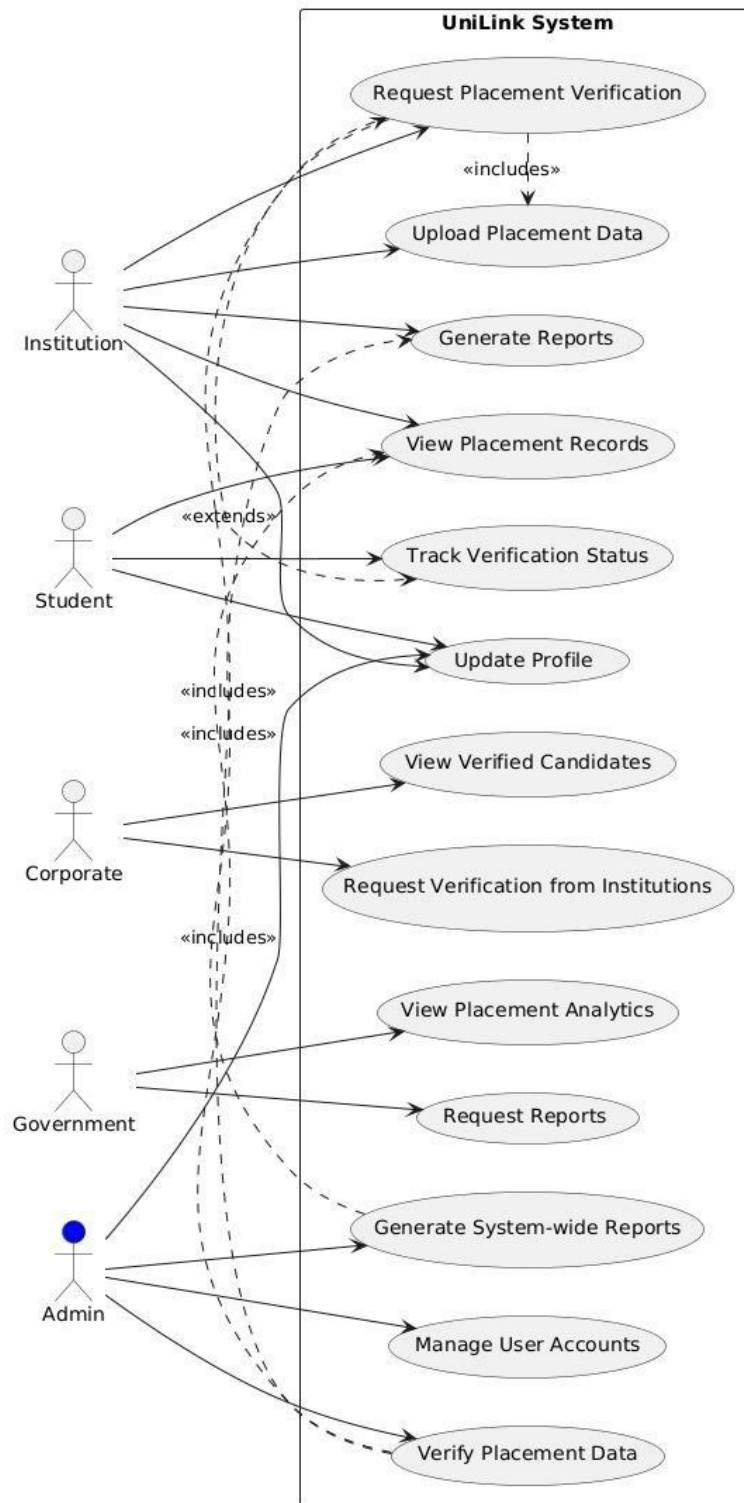


Figure 4.5: Use-case diagram

### 4.4.3 Sequence Diagram

The Sequence Diagram captures the time sequence of interactions among system components and external actors. It describes how colleges, companies, and students interact with UniLink's internal processes step by step.

The uploading of Placement Data:

#### **Institution Admin Uploads Data:**

The placement information is uploaded by the institution through the UniLink frontend.

#### **Frontend Sends Request to Backend:**

The frontend part of UniLink sends an HTTP POST request to the backend API endpoint - POST /api/placements/uploa

#### **Backend Validates and Stores Data:**

The UniLink backend takes in the data, performs a validation check on them for proper format and required fields, and then proceeds to store them within the database.

#### **Database Confirms Storage:**

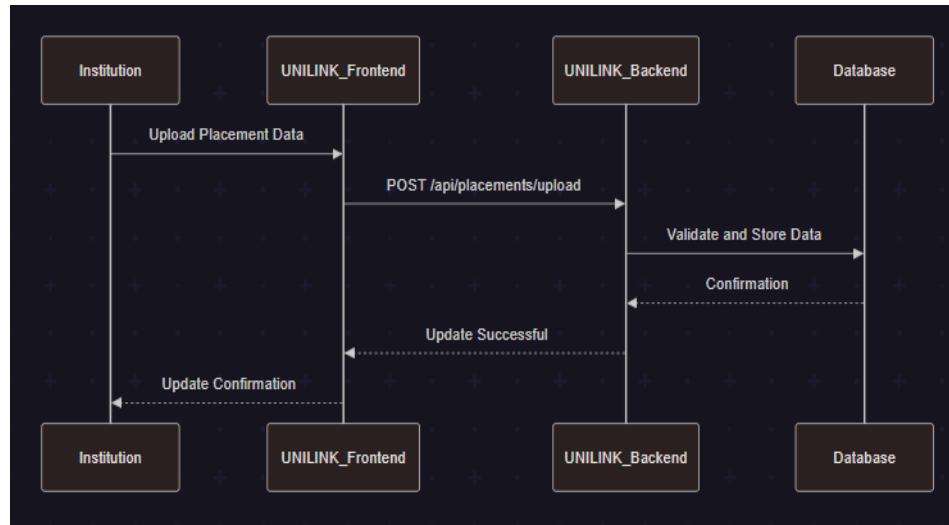
From the time the storage of data is confirmed by the database.

#### **Backend Sends Success Response:**

Once proper validations complete, and data is stored by the backend, it responds back to the frontend with a success message indicating that the upload was successful.

#### **Frontend Sends Confirmation to Institution:**

Finally, the confirmation of the completed upload along with success is advised to the institution through the front end.



**Figure 4.6: Placement upload flow**

The placement verification process might be as follows:

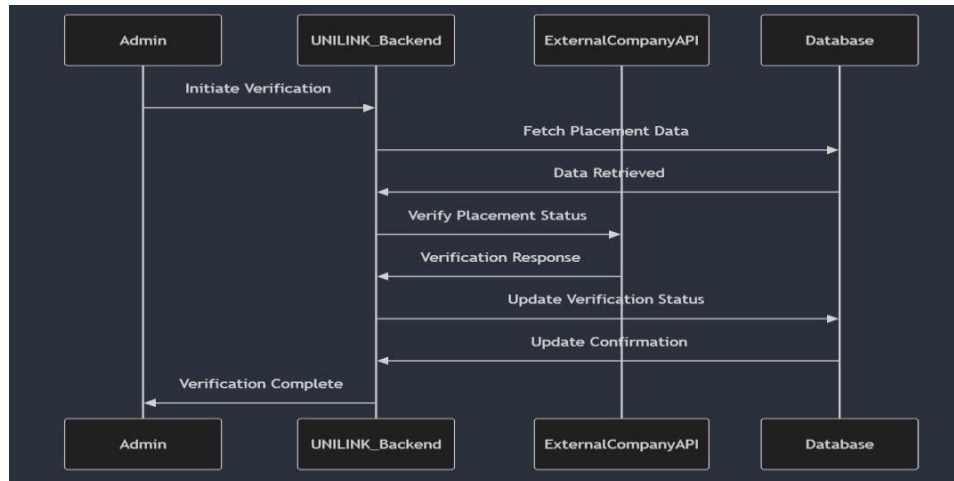
Institution Admin uploads placement information.

The file is parsed by the system and formatting errors are marked.

The Verification Engine sends requests to representatives of companies.

Company representatives validate and accept or reject the claim of the student's placement.

Status is reflected in real-time dashboards available to colleges, students, and government authorities.



**Figure 4.7: Verification process flow**

The Login Process will be as follows:

**User Enters Credentials:** This signifies that a user feeds into the frontend interface the login information, such as the username and password.

**Frontend Issues Login Request:** It is then passed down by the frontend to the backend towards the login, using the entered credentials.

**Backend Credential Verification:** The backend shall forward the credentials received from the user to the database and request verification.

**Database Validation:** The database tries to see if the credentials fit into an existing user record and answer with either "Valid" or "Invalid" status.

**Result of Backend Returns:** The backend of our project will return the state whether the initials were correct or incorrect.

**Front-end Present the User with the Outcome:** In the end the Front-end will showcase on the platform whether the Login was success or failed.



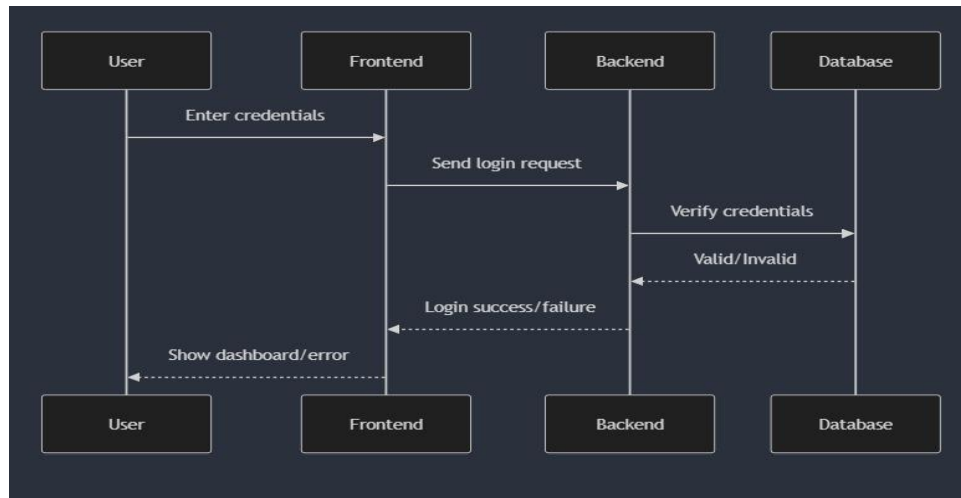


Figure 4.8: Login flow

#### 4.4.4 Database Desing / Entity Relationship Diagram (ERD)

The Uni-Link platform data architecture framework is demonstrated through the Entity Relationship Diagram (ERD) which shows the basic structure of users with related components for institutions, students, companies and government organizations.

##### 1. User

The central entity representing every system user

Attributes: ID, email, emailVerified, name, image, load, isVerified, verificationStatus, created At, updatedAt

Relationships: one-to-one with InstitutionProfile, CompanyProfile, GovernmentProfile, StudentProfile, one-to-many with Session, Document, Account

##### 2. Institution and institution profile

Institution stores detail of educational institutions

Attributes: ID, name, address, city, state, isVerified, verificationStatus, createdAt, updatedAt

Institution profile is linked to institution via the user

Attributes: ID

Relationships: one-to-one with user, one-to-many with student profile

### **3. Student profile**

Represents a student under an institution

Attributes ID enrollment number name gender branch department created at update at

Relationships mini to one which institution one to many with placement PR record verification token

### **4. Company and company profile**

Company contains information about the recruiting companies

Attributes: ID, name, website, address, city, state, isVerified, verificationStatus, createdAt, updatedAt

Company profile links company info to the user

Attributes: ID

Relationships: one-to-one with user, one-to-many with placement pecord

### **5. Government and government profile**

Government stores details of verifying authorities

Attributes: ID, name, level, department, website, isVerified, verificationStatus, created At, updatedAt

Government profile holds additional detail about the government user

Attributes: ID, time, department, designation

Relationships: one-to-one with user

## **6. Placement PTRRecord**

Represents placement detail of students

Attributes: ID, position, salary, joiningDate, placedStatus, verificationStatus, createdAt, updatedAt

Relationships: many-to-many with studentProfile and company

## **7. Verification token**

User to verify student's placement record

Attributes: ID, token, identifier, expires

Relationships: many-to-one with studentProfile

## **8. Document**

Stores uploaded documents associated with users

Attributes: ID, documentType, type, fileURL, createdAt, updatedAt

Relationships: many-to-one with user

## **9. Sessions**

Maintain user session information

Attributes: ID, sessionToken, expires

Relationships: many-to-one with user

## **10. Account**

Handles 3rd party login detail for a user

Attributes: ID, type, providerAccountID, refresh\_token, access\_token, expiresAt, token\_type, scope, ID\_token, session\_state

Relationships: many-to-one with user

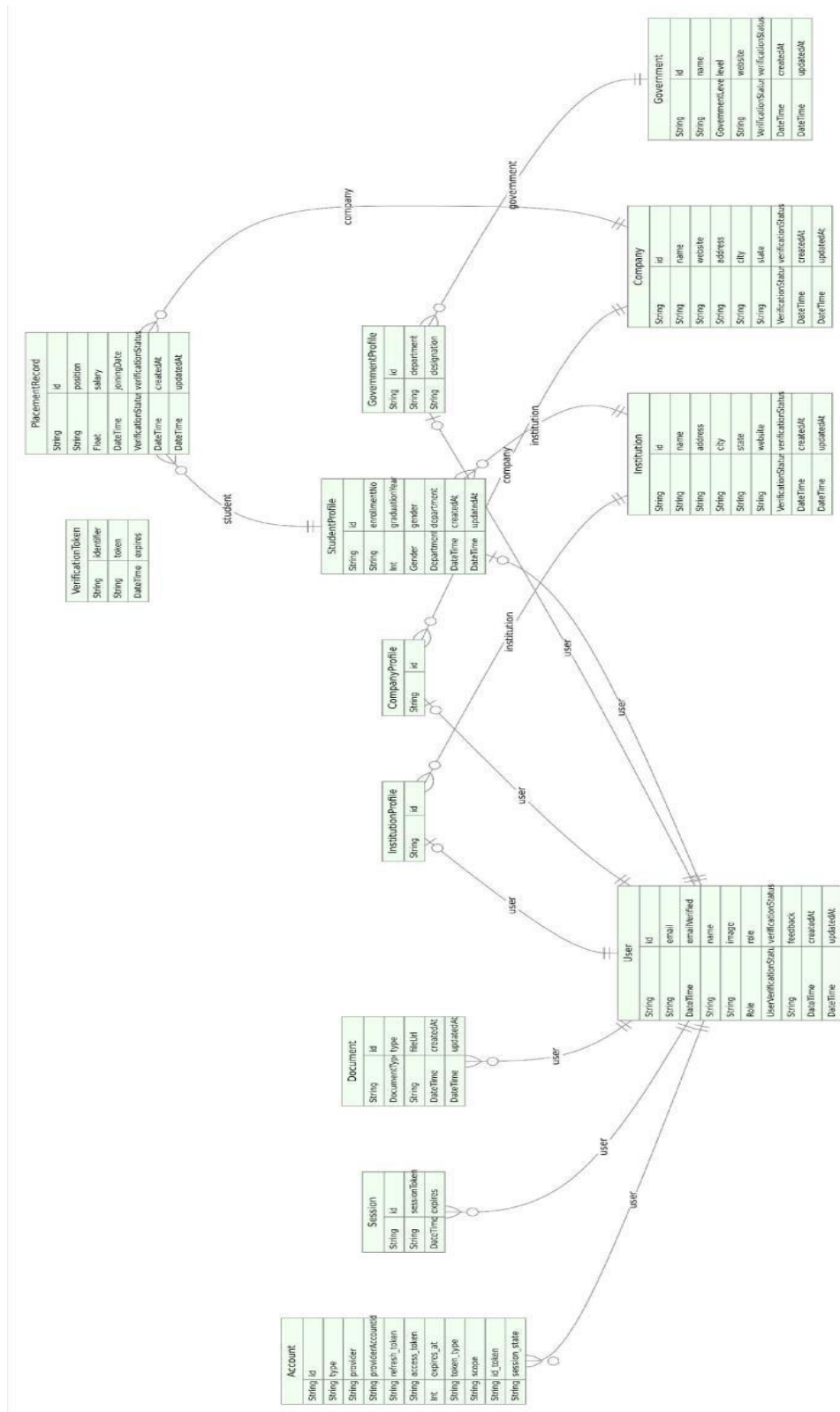


Figure 4.9: ER-Diagram

## **CHAPTER 5**

### **IMPLEMENTATION**

#### **5.1 Introduction Tools and Technologies Used**

##### **5.1.1 Languages:**

###### **JavaScript (with JSX in Next.js)**

JavaScript (JS) is a high-level, general-purpose, and popular scripting language used mostly for developing dynamic and interactive user interfaces in web development. In the UniLink platform, JavaScript was paired with JSX (JavaScript XML) within the Next.js framework to provide smooth rendering of UI components. Implementing JavaScript in Next.js also made client-side and server-side rendering (SSR) possible, with pages such as dashboards, verification logs, and placement records being quickly loaded and dynamically refreshed.

JavaScript made asynchronous processing of data requests (with Promises and Fetch API) possible, providing real-time updates on placement verification without reloading the page. JavaScript was also instrumental in processing event-driven activities, such as sending notifications when firms validate student placements or sending alerts to colleges about data mismatches during upload.

The flexibility of JavaScript was further utilized through libraries such as Recharts.js for graph visualizations of placement data, providing improved graphical beauty to the platform. In addition, JavaScript in Next.js API

routes was utilized on the server side to process API requests for data validation, data insertion, and fetching from the PostgreSQL database.

## **TypeScript**

TypeScript, a JavaScript superset, was employed in UniLink to improve the readability, scalability, and maintainability of the code. It enabled developers to specify static types, with fewer runtime errors and more deterministic data flow. TypeScript was especially helpful in managing data models, for example, in declaring the Student, Company, Institution, and Placement Record schemas so that data being passed between the backend and frontend had a uniform structure.

It also made code more scalable, as more than one developer could work on various modules without disrupting existing code functionality. Using TypeScript with Prisma ORM helped ensure that database queries were strongly typed, decreasing the likelihood of runtime errors and query misconfigurations. TypeScript also helped improve documentation of code for less maintenance time in the future.

## **CSS (Cascading Style Sheets)**

CSS (Cascading Style Sheets) was mainly applied to manage visual design and responsiveness of the UniLink platform. The platform was developed with Tailwind CSS, a utility-first CSS framework for quicker UI development using pre-set classes. CSS ensured uniform styling for all dashboards of the users, ranging from data tables to navigation menus, verification logs, and graphic reports. The implementation of responsive design concepts enabled UniLink to function optimally on desktops, tablets, and smartphones, making it accessible to everyone.

New CSS features such as Flexbox and Grid Layout were utilized to achieve a dynamic user interface without compromising performance.

Custom transitions, animations, and color schemes were also implemented using CSS to make the platform visually engaging for users from colleges, companies, and government agencies.

### **5.1.2 Technology Used**

#### **Next.js (React Framework)**

Next.js is an open-source React framework built by Vercel that is most famously used to create server-side rendered (SSR) and static sites. Next.js was selected as the core frontend framework for UniLink because of its capacity to create very dynamic and interactive user interfaces.

The main strength of Next.js is its hybrid server-side and static rendering, enabling the platform to load more quickly by pre-rendering pages and providing improved SEO. It adheres to the component-based structure, in which the whole frontend interface of UniLink (placement dashboards, data upload modules, verification status boards, etc.) was divided into individual reusable components.

Next.js API routes were used to implement backend endpoints, providing easy communication between the frontend and backend, eliminating the necessity of an external backend framework. It provided automatic page routing, which made it possible for the platform to provide multi-page navigation for login, placement dashboards, verification portals, and company access panels. API support built into Next.js enabled easy integration with third-party APIs such as LinkedIn API, Email services, and company verification APIs, making the automation capabilities of the platform stronger.

The prime reason why Next.js was used was because of its support for pre-rendering (SSR) and the provision of quick page load times, giving users a seamless experience even though they were processing large datasets from many colleges.

### **Node.js (Backend Runtime Environment)**

Node.js is a cross-platform, open-source, server-side runtime environment based on Google Chrome's V8 engine that enables developers to run JavaScript outside the browser. Node.js handled mainly the following in UniLink:

Frontend API calls (upload of placement data, verification requests, generating reports). Database operations via Prisma ORM (insertion, retrieval, updating of placement data). Email alerts sent to companies, colleges, and students.

The event-based and non-blocking nature of Node.js enabled the handling of concurrent requests without waiting, thus maximizing performance.

It facilitated the building of a RESTful API service to perform functions such as:

- Initiating placement verification requests.
- User session management.
- Management of company confirmations and status updates.

The choice of Node.js was motivated by its light and scalable nature, which is ideal for handling high-volume concurrent requests in UniLink.

### **Prisma ORM (Object Relational Mapper)**

Prisma ORM (Object Relational Mapper) was utilized to enable



communication between the application backend and the PostgreSQL database without having to write raw SQL queries. Prisma made it easier to manage database operations like inserting placement data, checking the status of the placement, retrieving company confirmation logs, and returning verification status reports. Prisma's strongest tool is its ability to migrate the schema of databases without downtime so that developers can change database schemas.

Prisma made it certain that data integrity, consistency, and structure remained intact throughout the platform so that data corruption and duplication would not be possible.

Moreover, Prisma facilitated support for TypeScript models, so data retrieved from the database followed prescribed types strictly and reduced runtime errors.

The usage of Prisma helped UniLink scale its data models rapidly without having to tamper with sophisticated SQL queries by hand.

### **PostgreSQL (Relational Database)**

PostgreSQL was selected as UniLink's lead relational database on account of high performance, high scalability, and sophisticated query capacity.

It was tasked with storing:

- Records of student placement.
- Company confirmations (approved/rejected placements).
- Logs of verification history.
- Roles and permissions of users.

PostgreSQL enabled the platform to handle complicated JOIN operations efficiently, making it possible for colleges, companies, and government agencies to view real-time placement information.

ACID-compliant transactions ensured data consistency and reliability in modifications such as placement status updates, company confirmations, and verification logs. Foreign key constraints in PostgreSQL also eliminated duplicate placement records, ensuring data consistency.

### **5.1.3 Tools**

While developing UNILINK, a number of tools were implemented to make development smooth, control versions, and deploy the platform. These tools played an essential role in streamlining the code, collaboration, and deployment process, thereby enhancing the development pace and efficiency. The major tools employed in UNILINK are:

#### **VS Code (Visual Studio Code)**

VS Code is a popular source code editor designed by Microsoft and selected due to its simplicity and robust capabilities. It offered capabilities such as syntax highlighting, code completion, and live debugging, enabling developers to write clean and error-free code. The terminal built into it enabled the team to execute Next.js server, Prisma commands, and database migrations within the editor itself. With JavaScript, TypeScript, and Prisma support, VS Code improved the overall development productivity. Moreover, GitHub version control was integrated directly into VS Code, making it simple to perform code push/pull operations.

#### **GitHub (Version Control)**

Version control was done through GitHub, and the team was able to work and manage changes to code accordingly. Developers could push, pull, and merge code changes without impacting the stability of the project. Code reviews and pull requests made it possible for changes to be inspected

and tested prior to merging with the main branch. GitHub also stored a copy of the complete source code so that there would not be accidental loss of data. Integration with Vercel enabled automatic deployment whenever new code was pushed.

### **Vercel (Deployment Platform)**

Vercel deployed and hosted UNILINK to ensure that the platform remained live and available to users at all times. It ensured automatic deployment whenever changes were implemented in the code, thus avoiding manual deployment. Vercel also featured a global content delivery network (CDN), ensuring faster loads for users from various locations.

It offered in-built HTTPS security, which encrypted and secured all data transmissions. It also supported custom domain configuration, enabling UNILINK to possess a professional and secure web address.

## **CHAPTER 6**

### **TESTING AND MAINTAINANCE**

#### **6.1 Testing Techniques and Test Cases Used**

##### **7.1.1 Testing Approaches**

Testing for UNILINK was a significant component of the development process, ensuring accurate placement information delivery, smooth verification procedures, and secure role-based access control. Because the platform handled extremely sensitive placement information input by colleges and validated by employers, it was imperative to thoroughly test various modules to ensure data integrity, system stability, and correct communication flow.

The test methods applied to UNILINK made sure all the user roles (institution, administrators, company agents, and government officials) correctly interacted with the system and also ensured the placement records were uploaded, confirmed, and rendered visually without glitches. The main test methods employed in UNILINK are:

##### **1. Functional Testing:**

Functional testing was performed to make sure that all the basic functionalities of

UNILINK behaved as per their intended functionality.

Test cases were also used to check if:

- Institution Admins were able to upload placement information in bulk through CSV files.

- Company Representatives were able to receive verification requests and approve or reject placements.
- Government Officials were able to view dashboards to graphically display placement statistics.

Data validation was also tested for the platform, which rejected invalid file formats, missing fields, or incorrect placement information with proper error messages.

Functional testing also included the verification flow, so that when placement verification was accepted by a company, the status updated in real-time and was accurately reflected throughout the platform.

## **2. Integration Testing:**

Integration testing was conducted to validate seamless interaction between various parts of UNILINK, including:

- Frontend (Next.js) with Backend (Node.js).
- Prisma ORM with PostgreSQL Database.
- File Upload Modules with CSV data parsing.

Test cases were created to make sure that:

- When a placement record was uploaded, it automatically sent a verification request to the company.
- When a company replied (approved/rejected), the status was automatically updated on the college's dashboard.
- Moreover, integration testing confirmed the seamless exchange of data between:
- Company verification → Placement Status Update → Government Report Generation.

This testing was essential in guaranteeing zero data loss between database

interactions and third- party APIs.

### **3. Performance Testing:**

Performance testing was done to see how well UNILINK performed under various load scenarios, particularly during:

- Simultaneous verification approvals/rejections by companies.
- Multiple real-time dashboard access by government personnel.

Response time was also tested to verify that:

- Data uploads were completed within 5 seconds.
- Company verification approvals updated status changes within 2 seconds.
- Dashboards for government upload real-time data within 3 seconds.

### **4. Security Testing:**

Security testing was the most important component of UNILINK as it touched sensitive placement data, student records, and corporate verification logs.

Key security test cases were:

- Role-Based Access Control (RBAC): Guaranteeing that
- Institution Admins were not allowed to view company or government dashboards.
- Companies were not able to edit placement records; they could only accept or reject.
- Government officials could only view placement analytics in total, not individual student records.

**SQL Injection Prevention:**

Preventing malicious queries from being able to manipulate or contaminate the database.

**File Upload Vulnerability:**

Testing to ensure that no unauthorized files (like malware or incorrect formats) could be uploaded.

**Data Encryption:**

Confirming that all the communication (frontend, backend, and database) was encrypted through HTTPS protocols.

The platform was successfully subjected to penetration testing, guaranteeing no unauthorized access, data leak, or intrusion by a third party.

**5. User Acceptance Testing:**

Institution Admins validated the platform by uploading placement information. Company Representatives validated the platform by sending responses to verification requests. Government Officials experimented with the dashboards to see placement trends and create reports.

The UAT primarily addressed:

- User role ease of use.
- Mirroring status updates in real time on dashboards
- Report generation accuracy for government agencies.

Feedback from users aided in enhancing the platform's UI/UX, minimizing data processing lag, and enhancing dashboard responsiveness.

## 6. Regression Testing:

Regression testing was conducted every time new features, bug fixes, or updates were added to UNILINK to make sure that previously functioning functionalities were not broken.

Test cases were:

- Cross-verification of old and new placement data.
- Verification if placements verified before still showed correctly after deployment updates.

This guaranteed that any future deployment or feature update did not affect the current workflow.

### 7.1.2 Test Cases

Test Case ID	Test Description	Input	Expected Output	Status
TC01	Login with valid credentials	Email	Redirect to appropriate dashboard	Pass
TC02	Login with invalid credentials	Wrong email	Error: Invalid credentials	Pass
TC03	Access admin route as government user	Logged in as government	Redirect to authorized page	Pass



<b>TC04</b>	Logout flow	Click logout	Redirect to login screen	Pass
-------------	-------------	--------------	--------------------------	------

**Table 6.1.2.1 User Authentication and Role-Based Access**

<b>Test Case ID</b>	<b>Test Description</b>	<b>Input</b>	<b>Expected Output</b>	<b>Status</b>
<b>TC05</b>	Create new company organization	Valid company details	Pending page for verification of data	Pass
<b>TC06</b>	Submit organization form with missing fields	Blank organization name	Validation error shown	Pass
<b>TC07</b>	Duplicate organization creation	Repeated company name/email	Error: Organization already exists	Pass
<b>TC08</b>	Create government organization	Valid govt details	Pending page for verification of dat	Pass

**Table 6.1.2.2 Organization Creation (Admin)**

<b>Test Case ID</b>	<b>Test Description</b>	<b>Input</b>	<b>Expected Output</b>	<b>Status</b>
<b>TC09</b>	Upload placement record (Institute)	Valid Form input	Data visible in government dashboard	Pass
<b>TC10</b>	Verify placement record (Company)	Select & approve record	Record marked as verified	Pass
<b>TC11</b>	View unplaced student count (Government)	Government login	Accurate stats displayed	Pass

**Table 6.1.2.3 Placement Data Management**

<b>Test Case ID</b>	<b>Test Description</b>	<b>Input</b>	<b>Expected Output</b>	<b>Status</b>
<b>TC12</b>	Ask placement stats by year	“How many students placed in 2024?”	Response with 2024 placement stats	Pass
<b>TC13</b>	Query unplaced count by	“Unplaced students in Mechanical	Response with number from	Pass

	college	?”	verified data	
<b>TC14</b>	Ask malformed question	“Show cats in KIET”	Error message or default fallback response	Pass
<b>TC15</b>	Ask query in Hindi	“KIET mein kitne bachhe place hue?”	Respond with answer in English	Pass

**Table 6.1.2.4 Chatbot-based Report Query System**

<b>Test Case ID</b>	<b>Test Description</b>	<b>Input</b>	<b>Expected Output</b>	<b>Status</b>
<b>TC16</b>	Navigate to dashboard via logo click	Click site logo	Redirect to role-based homepage	Pass
<b>TC17</b>	Broken route handling	Enter unknown URL	Show 404 not found page	Pass
<b>TC18</b>	Check dashboard rendering	Login and open dashboard with placement data	Graph load correctly with accurate visible data	Pass

**Table 6.1.2.5 UI and Navigation**

## **CHAPTER 7**

### **RESULTS AND DISCUSSIONS**

#### **7.1 Implemented Modules Description**

##### **7.1.1 Home Module**

- Primary entry point for Unilink, providing overview of features & navigation around them.
- University events and placements sections were included into it.
- Built with Next.js 14 and Tailwind CSS, making it highly responsive and user-friendly.

***Scope:***

- Easy, intuitive, and interactive interface.
- Users would be kept engaged with real-time data display.

##### **7.1.2 Admin Module**

- A central dashboard to manage the entire platform by the administrators themselves.
- It deals with users roles, approvals for data, and tracking all the activities happening within the system.
- Built using Next.js and Prisma, creating a role-based authentication for the users thus enhancing security .

***Scope:***

- The platform would show only authenticated and verified data.
- Features the possibility of effective management of organizations and users.

### **7.1.3 Organization Management Module**

- Allows the admin to create and manage organizations-institutions, companies, and even government entities.
- Provides structured onboarding of institutions in the placement database.

#### ***Scope:***

- Ensure that organizations are exactly classified.
- Structured workflows will be in place for submission approvals of information.

### **7.1.4 Verification Requests Module**

- It is responsible for the verification of the placement records submitted by the various institutes.
- The verification handles are then divided into the companies, governments, and institutes tabs for smooth processing and implementation.

#### ***Scope:***

- Only the verified data will be then displayed for the placements.
- Encouraged trust and transparency among placement statistics.

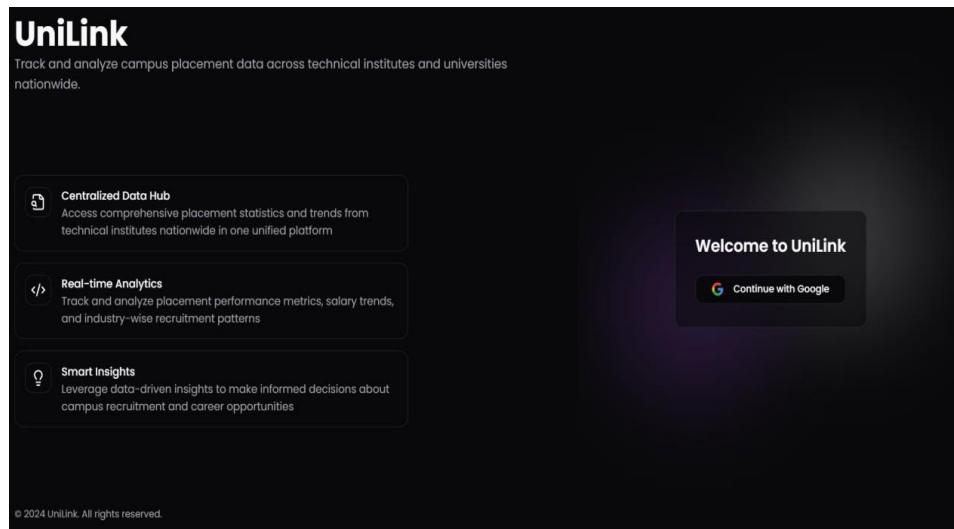


Figure 7.1: Sign in page

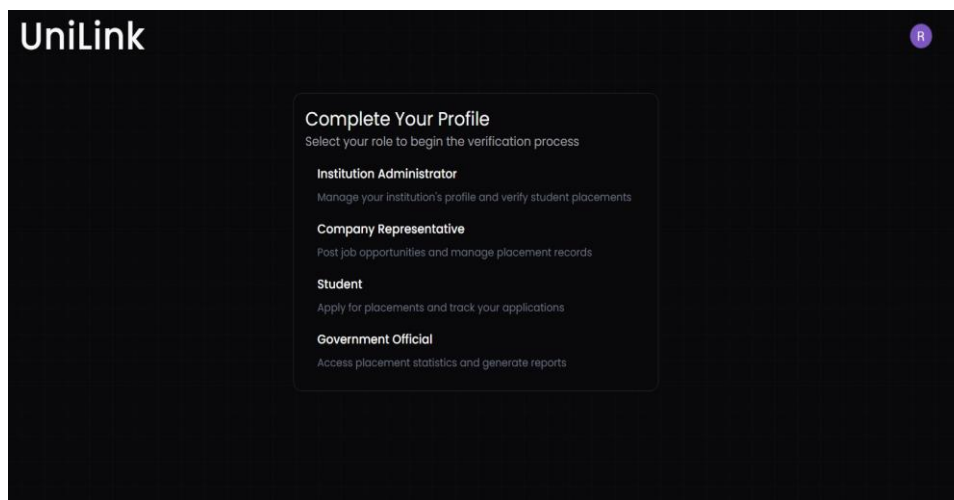
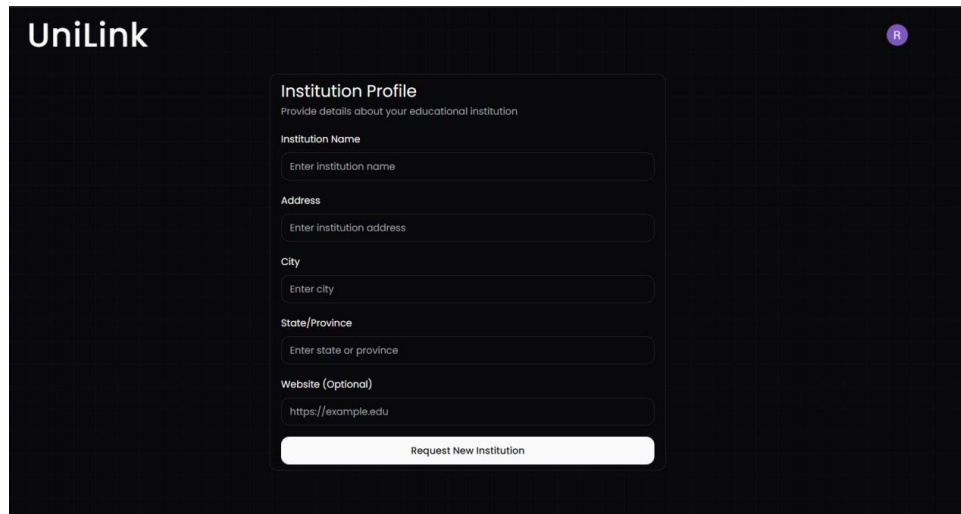


Figure 7.2: Role Based Access



The image shows a web form titled "Institution Profile" on a dark-themed interface. The form is part of the UniLink application, as indicated by the logo in the top left. It prompts the user to "Provide details about your educational institution". The form contains several input fields: "Institution Name", "Address", "City", "State/Province", and "Website (Optional)". Each field has a placeholder text indicating what to enter. At the bottom of the form is a button labeled "Request New Institution".

UniLink

**Institution Profile**  
Provide details about your educational institution

**Institution Name**  
Enter institution name

**Address**  
Enter institution address

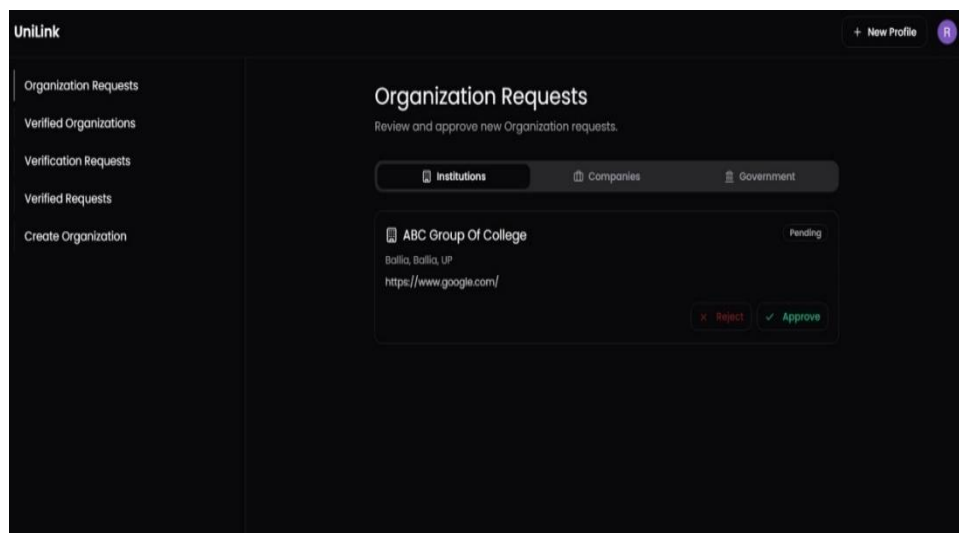
**City**  
Enter city

**State/Province**  
Enter state or province

**Website (Optional)**  
https://example.edu

Request New Institution

**Figure 7.3: New Institute Setup Form**



The image shows a web dashboard titled "Organization Requests" on a dark-themed interface. The dashboard is part of the UniLink application, as indicated by the logo in the top left. It prompts the user to "Review and approve new Organization requests." The dashboard has a sidebar on the left with links to "Organization Requests", "Verified Organizations", "Verification Requests", "Verified Requests", and "Create Organization". The main content area has a tabbed interface with three tabs: "Institutions", "Companies", and "Government". The "Institutions" tab is selected, showing a list of requests. The first request is from "ABC Group Of College" with the address "Ballia, Ballia, UP" and the website "https://www.google.com/". The request is in a "Pending" state. At the bottom of the request card are two buttons: "Reject" and "Approve".

UniLink

**Organization Requests**  
Review and approve new Organization requests.

Organization Requests | Verified Organizations | Verification Requests | Verified Requests | Create Organization

**Institutions** | Companies | Government

**ABC Group Of College** Pending  
Ballia, Ballia, UP  
https://www.google.com/

Reject Approve

**Figure 7.4: New Institute Request to Super Admin**

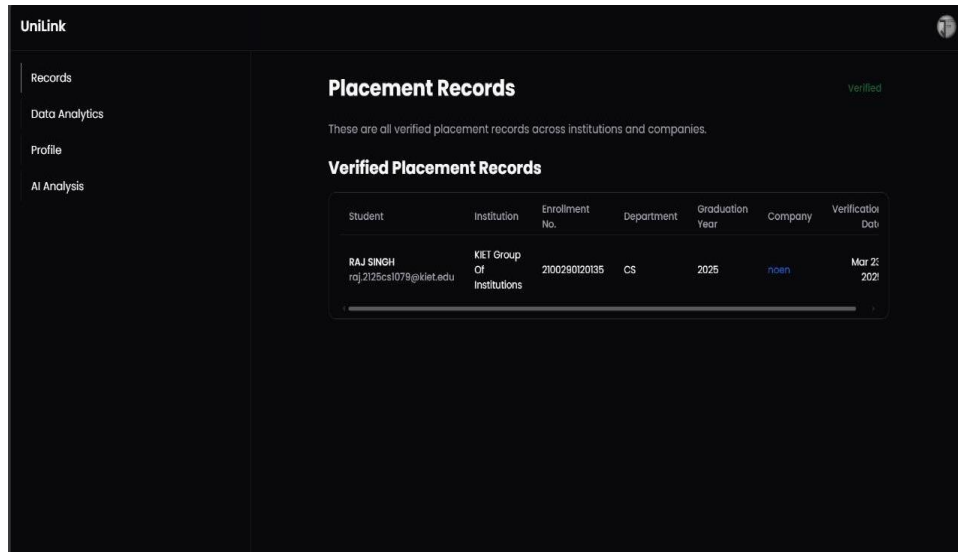


Figure 7.5: Government Dashboard



Figure 7.6: Data Representation



## **CHAPTER 8**

### **CONCLUSION AND FUTURE SCOPE**

#### **8.1 Conclusion**

Uni-Link platform has successfully addressed major issues of educational institutions in coordinating placement activities. By implementing innovative technologies such as Next.js, PostgreSQL, Prisma, and ShadCN UI, the platform ensures a seamless experience for recruiters, instructors, and students. Implementation of dynamic data visualization, efficient data handling, and safe authentication has ensured the entire placement process to be quicker, reliable, and hassle-free.

The Dashboard Module provides instant feedback, enabling decision-makers to track placement trends, student performance, and recruiter interaction with ease. The Notification Module ensures instant communication, avoiding delays and improving coordination. The Report Generation Module also makes data analysis easier by automating report generation, thereby saving much time and effort for institutional staff.

With the implementation of automation, Uni-Link has reduced manual interventions, improved data accuracy, and raised institutional transparency. Its ability to scale offers confidence that the platform will be easily manageable as rising data volumes and user requirements increase. The simplicity of the platform's design, coupled with its power, renders it an essential tool for education institutions seeking to improve their placement management system.

In the coming years, Uni-Link has great potential for growth and advancement. With constant evolution, the website can respond to future academic requirements, maintaining institutions in the lead position on

upcoming placement matters.

## **8.2 Future Scope**

While the Uni-Link platform has achieved its primary objectives, several enhancements can further improve its functionality and impact:

### **8.2.1 AI-Based Recommendations:**

- Implement AI-based career suggestions based on the skills, interests, and grades of students.
- Create computerized resume assessment software to give students individualized feedback on their CVs.
- Implement AI-driven interview preparation modules that provide simulated interviews and customized improvement advice.

### **8.2.2 Enhanced Data Analytics:**

- Create predictive models to predict placement trends and recommend industry-relevant skills.
- Incorporate a trend analysis dashboard for institutions to track changing recruiter preferences and job market needs.
- Allow real-time data insights to assist professors in guiding students effectively.

### **8.2.3 Integration with External Portals:**

- Facilitate smooth integration with leading job sites such as LinkedIn,

Naukri.com, and Indeed to increase placement prospects.

- Create APIs for data sharing between Uni-Link and HRMS portals to ease institutional processes.

#### **8.2.4 Improved Security Measures:**

- Enhance security with multi-factor authentication (MFA) for enhanced data protection.
- Enforce role-based access control in order to let sensitive information be accessed by permitted staff only.
- Implement audit logging to monitor all activities on the platform, providing accountability and transparency.

#### **8.2.5 Automated Feedback System:**

- Developing an automated feedback system to gather insights from students and recruiters, ensuring continuous platform improvement.

By incorporating these improvements, **Uni-Link** has the potential to become a robust, industry-standard platform that streamlines placement management and enhances institutional efficiency.

## REFERENCES

- [1] Mochammad Fariz Syah Lazuardy and Dyah Anggraini, “Modern Front End Web Architectures with React.Js and Next.Js,” *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.
- [2] Tarun, Shrivastava, V., & Pandey, A. (2024, March). Introduction on Next.js and its benefits and analysis. Department of Computer Science & Engineering, Arya College of Engineering & I.T., Jaipur, India.
- [3] Michael Stonebraker, Michael Hirohama, & Lawrence A. Rowe (1990). The implementation of Postgres. *IEEE Transactions on Knowledge and Data Engineering*, 2(1), 125–142. <https://doi.org/10.1109/69.50912>
- [4] Truica, C.O., Boicea, A., & Radulescu, F. (2013). Asynchronous replication in Microsoft SQL Server, PostgreSQL and MySQL.
- [5] Deepmala Srivastava. (2023). *An introduction to data visualization tools and techniques in various domains*. University of Texas at Dallas.
- [6] Ahmad Tasnim Siddiqui, “Data Visualization: A Study of Tools and Challenges,” *BGI, Faizabad; Department of Computer Science & Applications, Sherwood College of Professional Management, Lucknow, India*, 2021.
- [7] G. Dinesh, M. N. V. Surekha, D. Harika, A. Sunand, U. Dayanand Kumar, and A. Naveen, “Placements Analytics and Dashboard,” *Sri Vasavi Engineering College (Autonomous), Pedatadepalli, Tadepalligudem, Andhra Pradesh – 534101, India*, 2024.
- [8] Chander Velu, S. E. Madnick & Van Alstyne, M. (2013). Centralizing data management with considerations of uncertainty and information-based flexibility. *Journal of Management Information Systems*, 30(3),

179-212. <https://doi.org/10.2753/MIS0742->

[1222300307](https://doi.org/10.2753/MIS0742-1222300307)

- [9] Zatin Gupta, Rishabh Garg, Priyanshi Saxena & Ansh Bansal. (2022). *Data Analysis based Digital Step Towards Waste Management* Tirunelveli, India.
- [10] [Weiming Zhu](#) (2021). *A study of big-data-driven data visualization and visual communication design patterns*.
- [11] Antony Unwin (2020). *Why is data visualization important? What is important in data visualization?*

# PLAG REPORT

PCS25-41

## ORIGINALITY REPORT

<b>8%</b>	<b>8%</b>	<b>1%</b>	<b>5%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

## PRIMARY SOURCES

<b>1</b>	<b>www.coursehero.com</b> Internet Source	<b>2%</b>
<b>2</b>	<b>Submitted to Delhi Metropolitan Education</b> Student Paper	<b>1%</b>
<b>3</b>	<b>Submitted to ABES Engineering College</b> Student Paper	<b>1%</b>
<b>4</b>	<b>d_anggraini.staff.gunadarma.ac.id</b> Internet Source	<b>&lt;1%</b>
<b>5</b>	<b>Submitted to Universidade Nova De Lisboa</b> Student Paper	<b>&lt;1%</b>
<b>6</b>	<b>www.irjet.net</b> Internet Source	<b>&lt;1%</b>
<b>7</b>	<b>prer.hec.gov.pk</b> Internet Source	<b>&lt;1%</b>
<b>8</b>	<b>Submitted to University of Wales Institute, Cardiff</b> Student Paper	<b>&lt;1%</b>
<b>9</b>	<b>jurnal.ranahresearch.com</b> Internet Source	<b>&lt;1%</b>
<b>10</b>	<b>www.slideshare.net</b> Internet Source	<b>&lt;1%</b>
<b>11</b>	<b>hdl.handle.net</b> Internet Source	<b>&lt;1%</b>

25	Submitted to University of Western Sydney Student Paper	<1 %
26	docobook.com Internet Source	<1 %
27	imsec.ac.in Internet Source	<1 %
28	www.geeksmint.com Internet Source	<1 %
29	www.srtmun.ac.in Internet Source	<1 %
30	fddocuments.mx Internet Source	<1 %
31	www.digitalguardian.com Internet Source	<1 %
32	docslib.org Internet Source	<1 %
33	publications.lib.chalmers.se Internet Source	<1 %
34	www.aitooljunction.com Internet Source	<1 %
35	ajourneywithtime.weebly.com Internet Source	<1 %
36	atharvacoe.ac.in Internet Source	<1 %
37	ipu.msu.edu Internet Source	<1 %
38	pdfcoffee.com Internet Source	<1 %

39

pdfcookie.com

Internet Source

<1%

40

su-plus.strathmore.edu

Internet Source

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off



# PATENT PROOF



(Toll Free no.: 1800 1200 106 Mobile: 8588811998, 9911150880)

## Internal Undertaking for Patent/Copyright/Trademark

I/We SHIVANI (CS)

Main Applicant: KIET GROUP OF INSTITUTIONS DELHI - NCR

Co-applicant / INVENTOR: SHIVANI (CS) RAJ SINGH (212SCS1079) SHIVAM KUMAR (212SCS1106)  
SUDHANSHU SINGH (212SCS1128) VAIBHAV BISHT (212SCS1119)

I/We have in the course of my study/ employment invented titled, UniLink: A Unified Platform for Aggregating Placement Records of Higher Education Institutions in India by using the facilities of Institute and I/We are the true and first inventor.

I/We hereby abide by the IPR Policy which was approved by the management and now public to all stakeholders. Also, the intent of research policy of KIET is towards promoting and encouraging Students/Faculties for recognition of their work by promoting their invention through filing patent/copyright/trademark.

I/We are opting the OPTION - 1

I/We would like to engage with the institute for filing the patent/design/copyright/trademark as per IPR policy. I/We do not have any objection by giving unconditional rights to college (KIET Group of Institutions) to file and register the patent/design/copyright/trademark in their name.

I/We hereby state that we shall be abide by the IPR policy clause no. 8.2, 8.3, 9, 9.1, 9.2, 10, 10.1, a, b, c and 10.2 approved by college management.

I/We do not have any objection by giving unconditional rights to college (KIET Group of Institutions) to file and register the patent/design/copyright/trademark in their name. My/Our claims shall be as per the defined ratio in clause no. 10, 10.1, a, b, c and 10.2 of IPR for sharing revenue if generated through commercialization either by transferring technology fee/ royalty/ onetime fee or establishing the venture in future. Both parties shall keep update to each other as per clause and shall abide by the policy.

I/We have given this undertaking at my/our own will and without having any kind of compulsion and pressure by and on behalf of the Institute.

Signature of the Co-  
Applicant/Inventor(s)

.....  
.....

Dr. Vibhav Sachan  
(Dean R&D)

Recommendation of HoD

HOD.....  
Signature.....

Dr. K. Nagarajan  
(Associate Dean Patents)

Approved (Remarks if any) By: