

A Student-Centric Freelancing Platform for Bridge Academia and Industry

Sakshi Kate¹, Om Kawale², Piyush Ghore³, Pranav Belgamwar⁴, Prof. S. H. Thengil⁵

¹Department Of Information Technology, Sinhgad College Of Engineering, Pune-41

²Department Of Information Technology, Sinhgad College Of Engineering, Pune-41

³Department Of Information Technology, Sinhgad College Of Engineering, Pune-41

⁴Department Of Information Technology, Sinhgad College Of Engineering, Pune-41 Email-
sakshi.kate.scoe.it@gmail.com

Abstract - Existing freelancing platforms are highly competitive and mainly favor experienced professionals, making it difficult for students to gain real-world Equation, or Figure starts a sentence, it is spelled out. When occurring in the middle of a sentence, these words are abbreviated Sec., Ref., Eq., and Fig.

At the first occurrence opportunities and practical experience. To address this issue, this project proposes a **Student-Centric Freelancing Platform for Bridging Academia and Industry** that provides a secure and supportive environment exclusively for students. The platform enables students to showcase their skills, create portfolios, apply for freelance projects, and interact with clients efficiently.

The system is developed using modern technologies such as Node.js, Express.js, Supabase, PostgreSQL, HTML, CSS, JavaScript, and Tailwind CSS. It includes features like secure authentication, service management, payment integration, file handling, and project recommendation mechanisms. The proposed platform helps bridge the gap between academic learning and industry requirements by improving student employability, practical exposure, and professional confidence.

1.INTRODUCTION (Size 11, Times New roman)

The rapid growth of digital technology and the gig economy has increased the demand for freelancing platforms that connect skilled individuals with real-world opportunities. However, most existing freelancing platforms are dominated by experienced professionals, making it difficult for students and beginners to gain projects, practical exposure, and industry experience. To address this issue, the proposed project, “**A Student-Centric Freelancing Platform for Bridging Academia and Industry**,” provides a secure and dedicated environment exclusively for students to showcase their skills, create portfolios, apply for freelance projects, and collaborate with clients effectively. The platform integrates features such as secure authentication, service management, payment integration, file handling, and

recommendation systems using modern technologies like Node.js, Express.js, Supabase, PostgreSQL, HTML, CSS, JavaScript, and Tailwind CSS. The system aims to bridge the gap between academic learning and industry requirements by enhancing practical knowledge, employability, and professional confidence among students.

2 ARCHITECTURE FLOW:

This diagram illustrates the high-level system architecture of a web-based freelancing platform. It follows a modern, decoupled client-server model, dividing the system into three distinct layers: the presentation layer (frontend), the application layer (backend), and the data/external services layer.

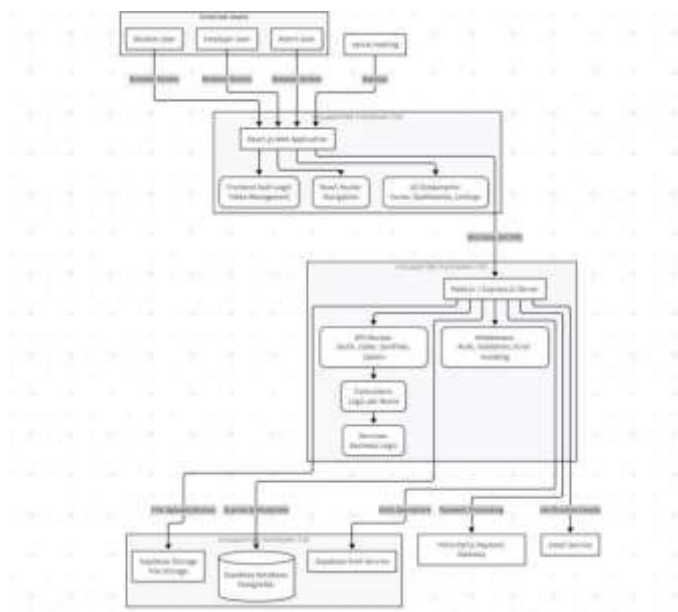


Fig.1 Architecture flow

The proposed system follows a modern multi-tier client-server architecture designed to provide scalability, security, and efficient communication between users, application services, and cloud resources. The architecture is divided into three major layers: the Presentation Layer (Frontend), the Application Layer (Backend), and the Data & External Services Layer. In the Presentation Layer, three types of users—Student User, Employer User, and Admin User—access the platform through web browsers. The frontend application is developed using React.js and deployed on Vercel Hosting, which ensures fast and reliable delivery of web content. The frontend contains UI components for forms, dashboards, and listings, React Router for navigation, and frontend authentication logic for session and token management. The frontend communicates securely with the backend through HTTPS-based REST API calls.

The Application Layer acts as the core processing unit of the system and is implemented using Node.js and Express.js. This layer contains API routes for handling functionalities such as authentication, job management, profile handling, and administrative operations. Middleware components are used for authentication, validation, and error handling to ensure system security and reliability. Controllers process incoming requests and coordinate with the business logic services that implement the main functionalities of the freelancing platform. The backend interacts with the Data & External Services Layer, where Supabase PostgreSQL is used for database management, Supabase Storage handles file uploads and access, and Supabase Authentication manages secure user login and verification. Additionally, a third-party payment gateway is integrated for secure transaction processing, while the email service is used for account verification and notification emails. This architecture ensures modularity, maintainability, secure communication, and efficient performance of the proposed student-centric freelancing platform.

2.1 DATA FLOW:

A Data Flow Diagram (DFD) is a graphical tool used to represent how data moves through a system. It shows the flow of information between different processes, data stores, and external entities, helping to visualize how inputs are transformed into outputs.

DFDs are useful in system analysis and design because they make complex processes easier to understand. They focus on what data is processed and where it flows rather than how the system operates, providing a clear overview of data movement within the system

2.1.1 Level-0 DFD:

The Level-0 Data Flow Diagram (DFD) provides a high-level overview of the proposed student-centric freelancing platform and illustrates how data flows between users, processes, databases, and external services. The system mainly consists of three core processes: **Account Management**, **Marketplace Listings**, and **Transaction Processing**. The student user interacts with the system to register, log in, search services, create listings, and place orders, while the system admin manages users, products, and order records.

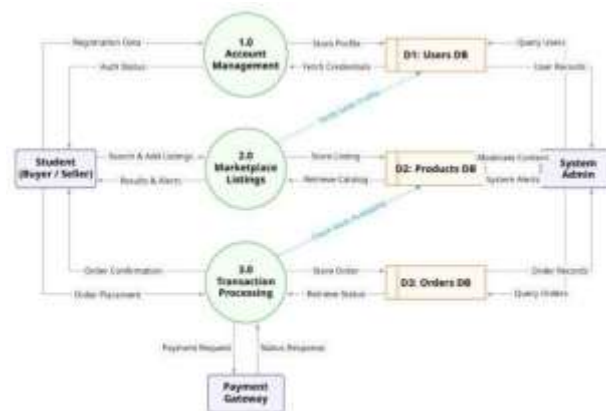


Fig 2: Level -0 DFD

The **Account Management (1.0)** process handles user registration and authentication by storing and retrieving user information from the **Users Database (D1)**. The **Marketplace Listings (2.0)** process allows students to create and search service listings, which are stored in the **Products Database (D2)** and monitored by the admin for moderation and verification. The **Transaction Processing (3.0)** process manages order placement, order confirmation, and payment activities by interacting with the **Orders Database (D3)** and the external **Payment Gateway**. Overall, the diagram represents the flow of information within the platform and shows how different modules interact to provide secure and efficient freelancing services.

2.1.2 Level-0 DFD:

The Level-1 Data Flow Diagram (DFD) shows the detailed working of the student-centric freelancing platform. It includes processes such as user authentication, profile management, service listing,

browsing services, order processing, and payment handling. The student can register, manage profiles, create listings, search services, place orders, and make payments through the platform.

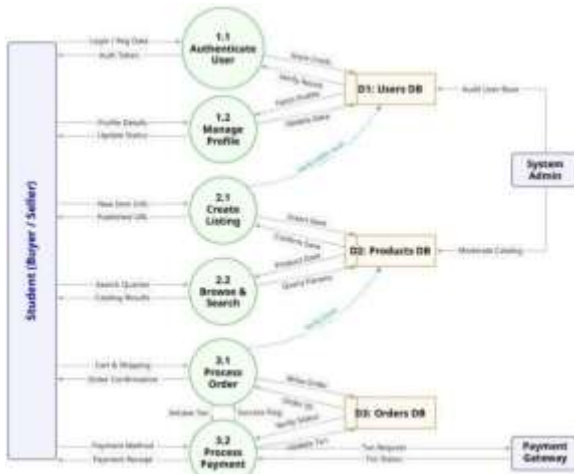


Fig 3: Level -1 DFD

The system interacts with three databases: **Users Database (D1)** for storing user information, **Products Database (D2)** for managing service listings, and **Orders Database (D3)** for storing order details. The **Payment Gateway** handles secure online transactions, while the **System Admin** monitors users and moderates platform activities to ensure security and smooth functioning of the system.

2.1.3 Level-2 DFD:

The Level-2 Data Flow Diagram (DFD) explains the detailed workflow of the order processing module in the student-centric freelancing platform. It includes three major processes: Manage Shopping Cart (3.1.1), Verify Stock & Price (3.1.2), and Generate Order Record (3.1.3). The student user can add or remove items from the cart, and the updated cart information is stored in the Cart Cache (D4).

During checkout, the system verifies product availability and pricing using the **Products Database (D2)**. After successful validation, the system generates an order record and stores it in the **Orders Database (D3)**. Finally, the payment process is initiated, and the user receives the order invoice and payment confirmation. This diagram represents the detailed internal flow of order and payment processing within the platform.

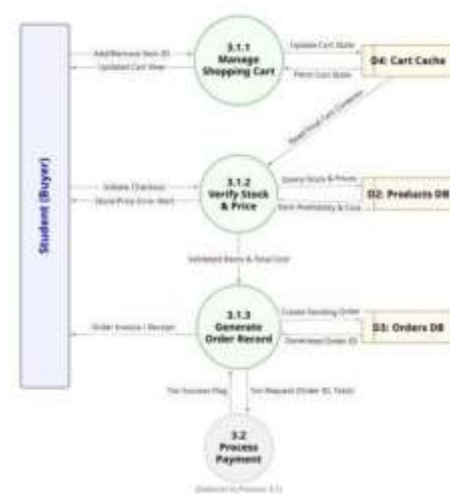


Fig 4: Level-2 DFD

3. IMPLEMENTATION:

The implementation of the proposed system was carried out using a modular and structured development approach to ensure scalability, maintainability, and efficient performance. The frontend of the application was developed using HTML, CSS, JavaScript, React.js, and Tailwind CSS to create a responsive and user-friendly interface. The backend was implemented using Node.js and Express.js, which handle server-side operations, API management, authentication, and business logic processing. The system uses Supabase with PostgreSQL as the database for storing user profiles, service listings, orders, reviews, and transaction details securely. Additionally, Supabase Authentication and Storage services were integrated to provide secure login functionality and safe file upload management for resumes and portfolios.

The platform includes modules such as user authentication, profile management, service creation, order processing, payment integration, and recommendation systems. RESTful APIs were developed for communication between the frontend and backend. A secure payment gateway was integrated to manage online transactions between clients and students. The recommendation system suggests suitable projects based on student skills and preferences, improving user engagement and project matching. The application was tested using unit and system testing methods to ensure reliability, security, and smooth performance. Finally, the system was deployed using cloud-based hosting services, making the platform accessible through web browsers for real-world usage.

4. RESULTS:

The output interface of the proposed student-centric freelancing platform is designed to provide a simple, responsive, and user-friendly experience for students, clients, and administrators. The system includes multiple interfaces such as the registration and login page, student dashboard, service listing page, project management section, payment interface, and admin panel. Students can create profiles, upload portfolios, browse available projects, and manage orders through an interactive dashboard, while clients can search services, place orders, track project progress, and provide feedback efficiently.

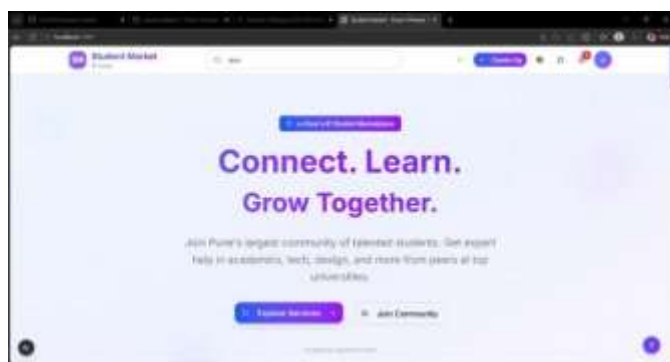


Fig 5.1 User Interface

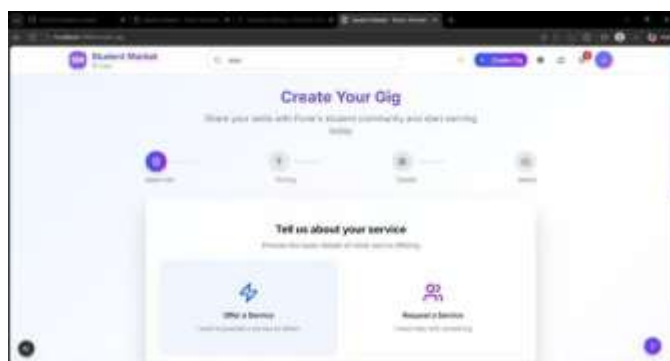


Fig 5.2 User Interface

5. CONCLUSIONS:

The project “A Student-Centric Freelancing Platform for Bridging Academia and Industry” successfully addresses the challenges faced by students in gaining practical industry exposure and freelance opportunities. The system provides a secure and dedicated platform where students can showcase their skills, create portfolios, apply for projects, and interact with clients efficiently. By integrating features such as authentication, service management, secure file handling, payment processing,

and recommendation systems, the platform ensures smooth and reliable communication between users.

The proposed system is developed using modern technologies such as Node.js, Express.js, Supabase, PostgreSQL, React.js, and Tailwind CSS, which improve the scalability, performance, and security of the application. The implementation and evaluation results demonstrate that the platform effectively bridges the gap between academic learning and industry requirements by enhancing practical knowledge, employability, and professional confidence among students. Overall, the project provides a scalable and user-friendly solution that supports student growth, skill development, and career readiness in the digital freelancing ecosystem.

6. REFERENCES:

- [1] Berg, V., Birkeland, J., Nguyen-Duc, A., Pappas, I. O., & Jaccheri, L., “Software Startup Engineering: A Systematic Mapping Study,” *Journal of Systems and Software*, vol. 144, pp. 255–274, 2018.
- [2] Satell, G., “The 4 Types of Innovation and the Problems They Solve,” *Harvard Business Review*, vol. 6, pp. 1–6, 2017.
- [3] Burke, A., & Cowling, M., “On the Critical Role of Freelancers in Agile Economies,” *Small Business Economics*, vol. 55, pp. 393–398, 2020.
- [4] Gupta, V., Fernandez-Crehuet, J. M., & Hanne, T., “Fostering Product Innovations in Software Startups through Freelancer Supported Requirement Engineering,” *Results in Engineering*, vol. 8, 100175, 2020.
- [5] Gupta, V., Fernandez-Crehuet, J. M., Hanne, T., & Telesko, R., “Requirements Engineering in Software Startups: A Systematic Mapping Study,” *Applied Sciences*, vol. 10, no. 17, 6125, 2020.