ASHESI UNIVERSITY

Career Services On Finger Tips (CSOFT)

Final Project Report

Edward Ofosu Mensah

Nana Kwaku Amoako

Kelvin Kekeli Ahiakpor

Ronelle Cudjoe

Jemima Nhyira Antwi

CS415: Software Engineering

Dr. Umut Tosun

May 2, 2025

***Abstract*—Career Services On Finger Tips (CSOFT) is a software solution developed to modernize and streamline the operations of the Career Services Office at Ashesi University. It addresses inefficiencies in manual systems by automating key functions such as attendance tracking, internship request management, and communication between students and career advisors through a centralized platform.**

Built using a modern tech stack, React.js, Supabase, and PostgreSQL, CSOFT offers a user-friendly, scalable, and secure environment for managing career-related activities. Core features include QR-based session check-ins, automated internship workflows, real-time analytics dashboards, and notification systems.

By automating routine processes, CSOFT enables career services staff to focus on personalized student support while improving responsiveness and access to resources. The system enhances career readiness, strengthens employer engagement, and promotes data-driven decision-making, aligning with Ashesi University's commitment to innovation and student success.

I. Introduction

Career Services On Finger Tips (CSOFT) is a web-based platform developed to enhance and streamline the operations of the Career Services Office at Ashesi University. The system centralizes various functions, including attendance tracking for career development sessions, internship request submissions and approvals, and analytics for student engagement. Through CSOFT, students can interact more effectively with career services, while staff are better equipped to manage workflows and provide personalized support.

The platform incorporates automated notification features to inform students about upcoming sessions and status updates, alongside real-time dashboards that enable advisors to monitor trends in attendance and internship activity. These features reflect broader trends in higher education, where digital transformation initiatives are increasingly being adopted to foster student success by improving engagement and operational efficiency [1]. Furthermore, centralized career services systems have been shown to enhance both accessibility and effectiveness for students and administrative staff [2]. By integrating these components, CSOFT seeks to improve the visibility, responsiveness, and accessibility of Ashesi's career services, thereby supporting the institution's mission to graduate career-ready individuals capable of succeeding in a global workforce [3].

II. Software Engineering Methodology

The CSOFT team employed the Agile software development methodology, specifically adopting the Scrum framework. This approach was chosen due to its iterative nature and its ability to accommodate evolving requirements through continuous feedback from key stakeholders, including students, career advisors, and university administrators.

Key motivations for selecting Scrum include:

- Support for frequent testing and integration of new features, ensuring continued relevance and system functionality.
- Enhanced collaboration between developers and users, allowing for more user-centered enhancements.
- Flexibility to adapt to institutional changes, such as updated career service policies or operational procedures.
- Improved team communication and transparency, which ensures alignment on project goals and milestones.

III. Literature Review / Related Work

In recent years, there has been a growing recognition of the importance of digitizing student support services in higher education. Career services, in particular, have undergone significant transformation, with institutions leveraging digital tools to enhance student engagement and operational efficiency. A 2022 report by EDUCAUSE revealed that over 80% of higher education institutions in the U.S. are investing in digital platforms to modernize administrative and support services. These systems are designed to offer scalable, student-friendly solutions to meet rising expectations for accessibility, responsiveness, and data-driven guidance [4].

Globally, several platforms have emerged to address specific career development needs. For example, systems like Handshake and Symplicity provide centralized platforms where students can book appointments, access job listings, and submit internship or job application materials. While these systems offer robust features, they often lack the customization and contextual sensitivity required by smaller institutions like Ashesi University. Furthermore, many of these tools are not tailored to the specific workflows, institutional goals, or student experiences present in African higher education contexts.

In the African context, digital transformation in higher education is gaining momentum. A systematic review highlights that while there's a growing body of research on student support services in sub-Saharan Africa, there's a notable gap in the implementation of digital solutions tailored to the region's unique challenges and needs [5]. Additionally, the World Bank emphasizes the role of digital transformation in driving development across Africa, noting significant strides in internet access and the utilization of digital services, including online learning platforms [6].

At Ashesi, the Career Services Office currently relies on a combination of manual and semi-digital processes to manage student interactions. These include tracking attendance on paper, issuing internship introduction letters through email, and disseminating resources via shared drives or PDF documents. This disjointed system can lead to inefficiencies, delays, and a lack of data visibility. As noted by the National Association of Colleges and Employers (NACE), 65% of career services offices report administrative inefficiencies as a major barrier to providing individualized student support [7].

To address these challenges, platforms like CareerSpaces and ReadyforWork have emerged, offering AI-powered career guidance and digital skill development tailored to the African context. CareerSpaces utilizes AI algorithms to

analyze a student's interests, skills, and goals, providing personalized career recommendations and resources to help students identify potential career paths that align with their strengths and aspirations [8]. Similarly, ReadyforWork is an immersive digital career accelerator that uses Artificial Intelligence and Machine Learning to equip job seekers with in-demand skills, connecting them to job opportunities in high-growth firms [9].

CSOFT seeks to build upon these existing efforts by offering a unified, context-sensitive platform designed specifically for Ashesi's needs. It incorporates best practices from global career services software, such as real-time reporting, automated notifications, and secure document handling, while also introducing localized features like a dedicated resource hub tailored to Ashesi students. The platform also aligns with the broader educational shift towards data-informed decision-making, enabling career advisors to assess engagement trends and optimize programming accordingly.

This literature and technology landscape underscores the need for a purpose-built system like CSOFT, which bridges the gap between global innovation and local implementation, ensuring that career services at Ashesi can effectively support student success from application to graduation and beyond.

IV. Software Engineering Cycle

A. Functional Requirements

FR01: User Management
The system should allow administrators to create, update, and delete user accounts for students and career advisors.

- FR 1.1.1: Administrators can create, update, and delete user accounts.
- FR 1.1.2: Role-based access control for system settings, internship requests, and approvals.

FR02: Attendance Tracking & Event Management
The platform should enable students to check in to career development sessions using a QR code or unique identifier, and allow administrators to create and manage events.

- FR 1.2.1: Students can check in to sessions using a QR code or unique identifier.
- FR 1.2.2: Career advisors can view and export attendance reports for students and sessions.
- FR 1.2.3: Administrators can create events and generate QR codes for attendance tracking.

FR03: Internship Letter Request Management
Students should be able to request internship letters through the platform. To be eligible, students must attend 3 events, provide feedback on them, and book a one-on-one session.

- FR 1.3.1: Students can submit internship request letters.
- FR 1.3.2: Career advisors can review, approve, or reject internship requests.
- FR 1.3.3: Automated notifications sent to students on request status updates.
- FR 1.3.4: System enforces eligibility requirements (3 events, feedback, one-on-one session).

FR04: Resume Review System
The system should allow students to upload resumes and receive feedback from career advisors.

- FR 1.4.1: Students can upload resumes in PDF/DOCX formats.
- FR 1.4.2: Career advisors can annotate and provide feedback on resumes.
- FR 1.4.3: System tracks revision history and improvement metrics.

FR05: One-on-One Session Booking
The system should facilitate the scheduling of one-on-one sessions between students and career advisors.

- FR 1.5.1: Students can view advisor availability and book sessions.
- FR 1.5.2: Advisors can set their availability and manage bookings.
- FR 1.5.3: System sends reminders and confirmation emails.

FR06: Reporting and Analytics
The platform should provide dashboards for career advisors to view career session attendance, internship request trends, and approval timelines.

- FR 1.6.1: Dashboards display attendance rates, request trends, and timelines.
- FR 1.6.2: Reports exportable in PDF or Excel formats.

FR07: Notifications and Reminders
Automated email or in-app notifications for students about career sessions, pending requests, and approval statuses.

- FR 1.7.1: Students receive automated notifications on sessions, requests, and statuses.
- FR 1.7.2: Career advisors receive reminders for pending requests.

FR08: Resource Hub
Page with all the necessary resources like cover letter templates, resume templates, and others.

- FR 1.8.1: Students can download specific resources they need.
- FR 1.8.2: Students can request the addition of new resources, and career advisors can review these requests.

B. Non-Functional Requirements

NFR01: Performance
The system should ensure smooth operation even under high user load.

- NFR 2.1.1: Support up to 1000 concurrent users without performance degradation.
- NFR 2.1.2: Response time for user actions (e.g., submitting a request, checking attendance) should not exceed 5 seconds under normal load.

NFR02: Scalability
The platform should accommodate growth in users and data efficiently.

- NFR 2.2.1: The platform should scale horizontally to handle increasing numbers of users and data as the institution grows.

NFR03: Usability
The platform should offer an intuitive and accessible user experience.

- NFR 2.3.1: Intuitive user interface requiring minimal training for students, advisors, and administrators.
- NFR 2.3.2: Mobile-responsive design to ensure accessibility across various devices.

NFR04: Reliability
The system should provide consistent availability for all users.

- NFR 2.4.1: System uptime should be at least 80% to ensure continuous availability.

NFR05: Maintainability
The system should allow easy updates and efficient maintenance.

- NFR 2.5.1: Modular architecture for easy updates and feature additions.
- NFR 2.5.2: Comprehensive documentation for developers and administrators to facilitate troubleshooting and maintenance.

V. Analysis and Design

The CSOFT (Career Services On My Finger Tips) platform has been designed with scalability, modularity, and user-centered functionality in mind. The system architecture is component-based, leveraging modern design patterns and agile practices to ensure maintainability and efficient collaboration.

A. Software Engineering Methodology and Tools

The CSOFT development team adopted the Agile methodology, applying the Scrum framework to manage progress in an iterative and collaborative manner. This choice aligned with our need to remain flexible and responsive to evolving user needs throughout the project.

Each sprint cycle lasted two weeks and was structured around:

- **Sprint Planning Meetings:** At the start of each sprint, the team held planning meetings to define tasks, assign responsibilities, and set achievable goals.
- **15-minute Daily Standups:** These short meetings ensured all team members remained aligned, allowing us to track individual progress and address blockers early.
- **Sprint Retrospectives:** Retrospectives were conducted at the end of each sprint to reflect on what went well, what could be improved, and how to adapt going forward.

We used **Jira** as our central coordination tool, leveraging it to:

- Document user stories, acceptance criteria, and design assets,
- Assign tasks and track their priority and status,
- Monitor sprint goals through burndown charts and reports,
- Store testing reports, feedback notes, and retrospective outcomes.

To illustrate our workflow and agile process, we have included evidence from **Sprint 1, our Jira project documentation, and a project timeline diagram.** These materials reflect real planning data, key development decisions, and outputs such as system diagrams, UI screenshots, and demo links.

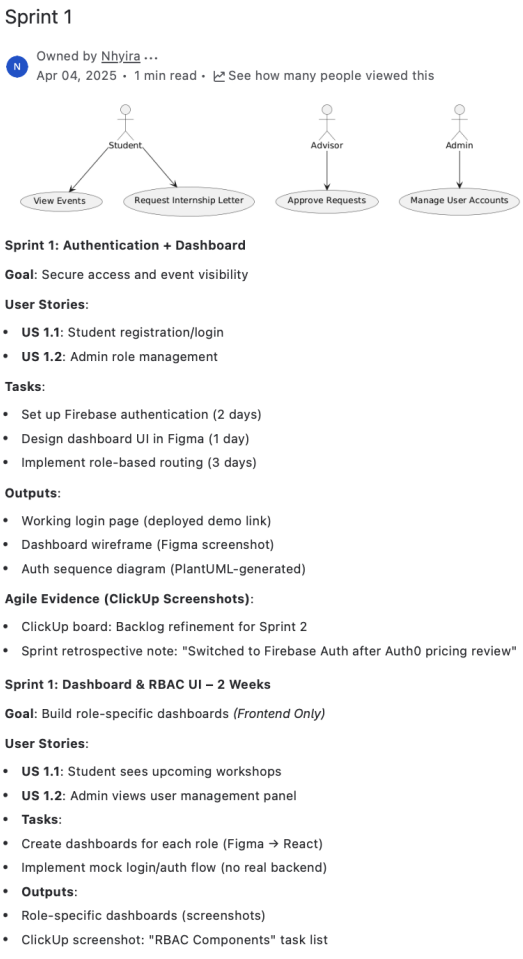

Fig. 1. Sprint 1 Planning and Tasks

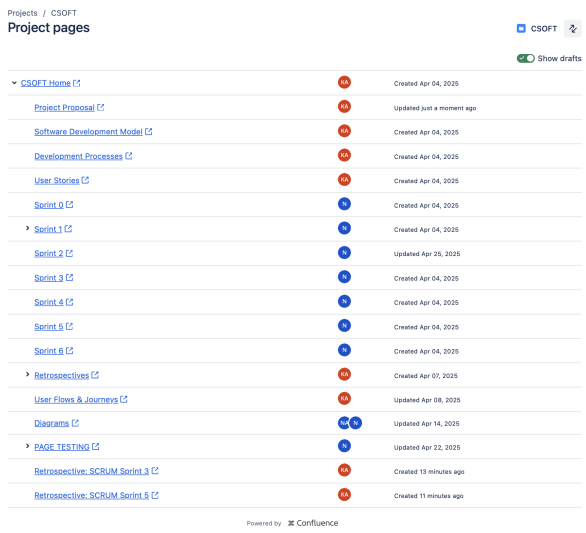

Fig. 2. Jira Project Page Documentation

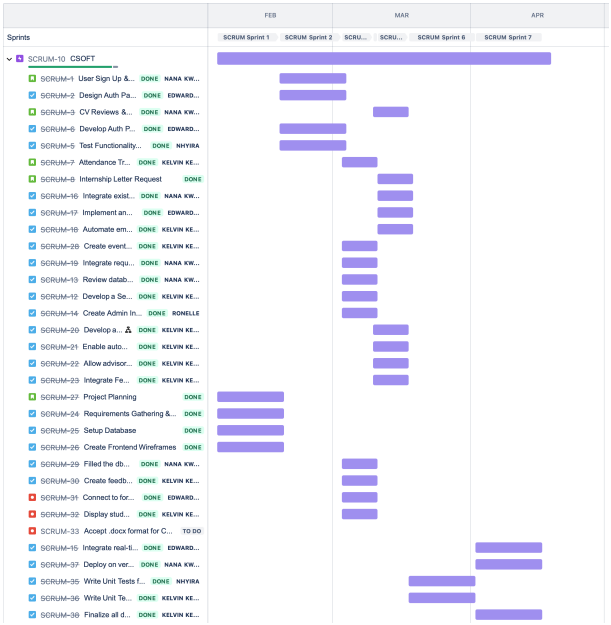

Fig. 3. Project Timeline Diagram

This methodology supported a structured, transparent, and adaptive development cycle aligned with both user needs and project objectives.

B. Design Patterns

To enhance system scalability and promote clean architecture, the team implemented the following design patterns:

1) Observer Pattern: The Observer pattern was used to handle dynamic, state-based UI updates across the platform.

Use Cases in CSOFT:

- **Theme Management:**
  - *Subject:* `ThemeProvider` manages the current theme state (light or dark).
  - *Observers:* UI components (e.g., buttons, backgrounds) subscribe via `useTheme()` to automatically adapt to theme changes.
- **Notifications System:**
  - *Subject:* `NotificationCenter` receives new notifications.
  - *Observers:* `HeaderBellIcon`, `NotificationListPage`, and others update reactively when new notifications are pushed.

This pattern ensures real-time synchronization across UI elements, improving responsiveness and user experience.

2) Singleton Pattern: The Singleton pattern was employed to manage shared service instances across multiple components.

Use Case in CSOFT:

- **Attendance Management:**
  - *Singleton:* `ActiveAttendance` service ensures only one instance controls a live QR code scanning session during an event.
  - *Users:* `QRScannerPage` and `AdminDashboard` both interact with the same instance to prevent duplication or conflicts.

This ensures consistent and centralized control over event attendance data.

C. Feature Organization

The CSOFT platform is modularly organized into several core components:

- **Dashboard:** Central hub with quick access to essential career services.
- **Resume:** Upload and manage CVs; submit for reviews.
- **Events:** Discover and register for career workshops, job fairs, and seminars.
- **Internship Requests:** Submit, track, and receive updates on internship request letters.
- **1-on-1 Sessions:** Book personalized consultations with career advisors.
- **Resources:** Repository for critical templates and guides (e.g., resumes, cover letters).
- **Notifications:** Central feed for alerts and reminders about upcoming events, request statuses, and system updates.

This structure enables ease of navigation, promotes user engagement, and ensures scalability for future feature additions.

D. System Diagrams

To complement the textual description of the system, the following diagrams provide a visual overview of the architecture and component interactions within CSOFT:

1) Class Diagram: This diagram illustrates the structural organization of CSOFT, showing the main classes and their relationships. It includes key entities such as `User`, `Resume`, `Event`, `Notification`, `Session`, and `Resource`.

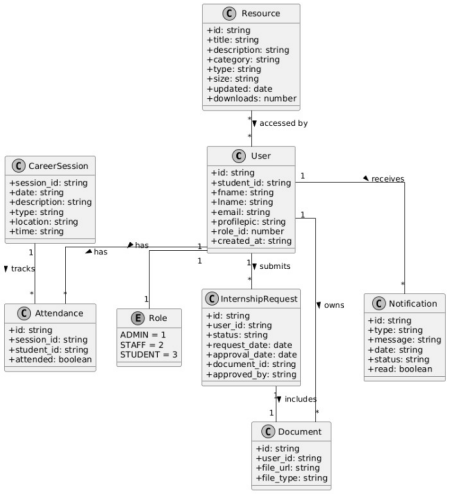

Fig. 4. Class Diagram for CSOFT

2) Flowchart: The overall flowchart maps the primary navigation and user flow through the platform—from landing on the dashboard to engaging with services like booking sessions, uploading resumes, and accessing resources.

Fig. 5. System Flowchart

3) Sequence Diagrams: These sequence diagrams illustrate dynamic interactions for specific features:

- **Attendance Tracking and Event Creation:** Demonstrates how the admin creates an event and how QR code attendance is recorded via a shared `ActiveAttendance` service (Singleton).
- **Resume Uploads and Notes:** Shows how users upload resumes, receive feedback, and interact with saved notes or review comments.

These diagrams collectively highlight the internal logic, user-system interactions, and the design choices made to enhance maintainability and real-time responsiveness.

VI. Implementation

A. General System Overview

The Career Services Online Facilitation Tool (CSOFT) is a comprehensive web application designed to streamline career services management for educational institutions. The system facilitates interaction between students, administrators, and career services staff through a robust, role-based platform.

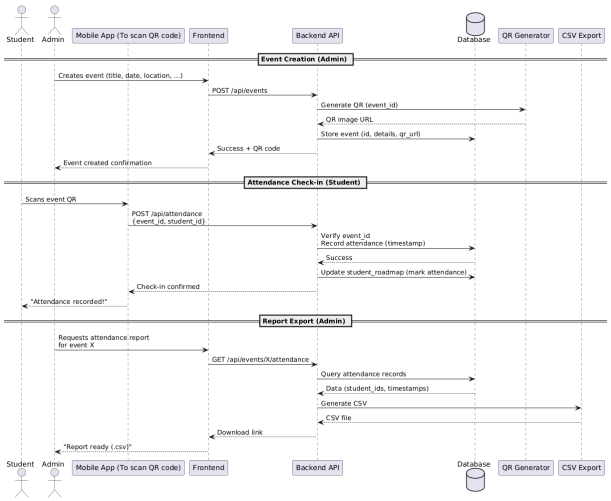

Fig. 6. Sequence Diagram: Attendance Tracking and Event Creation

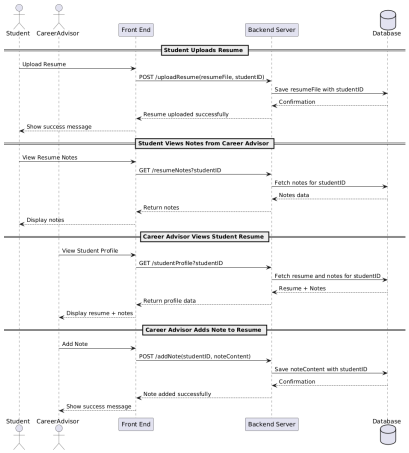

Fig. 7. Sequence Diagram: Resume Review Flow

1) Core System Components:

- **Authentication System:**
  - Multi-role user authentication (Student, Admin, Super Admin)
  - Secure login, signup, and password management
  - Invitation-based account creation for controlled access
  - Session management with token refresh mechanisms
- **Dashboard Interfaces:**
  - Role-specific dashboards with tailored functionality
  - Student dashboard for accessing resources, events, and career tools
  - Admin dashboard for managing events, students, and resources
  - Super Admin dashboard for system-wide administration

- **Event Management:**
  - Creation and management of career events and sessions
  - QR code-based attendance tracking
  - Event feedback collection and analysis
  - Historical attendance records
- **Resource Management:**
  - Career resources repository
  - Resume upload and management
  - Career roadmap visualization
  - Internship request processing
- **Profile Management:**
  - Personal information management
  - Progress tracking for career development
  - Resume management and feedback
- **Notification System:**
  - User notifications for important updates
  - Event reminders and announcements

B. Technical Architecture

The system is built using `Next.js 14`, leveraging its App Router for server-side rendering and client-side navigation. The application follows a component-based architecture with reusable UI components from a custom component library based on `shadcn/ui`.

Key technical features include:

- Server-side rendering for improved SEO and initial load performance
- Client-side navigation for a smooth user experience
- Server components for data fetching and business logic
- Client components for interactive UI elements
- API routes for backend functionality
- Middleware for authentication and route protection
- Responsive design for mobile and desktop compatibility

C. 3-Tier Architecture

CSOFT implements a classic 3-tier architecture, clearly separating the presentation, business logic, and data access layers while leveraging Next.js's hybrid rendering capabilities.

1) Presentation Layer (UI Tier): The presentation layer is responsible for rendering the user interface and handling user interactions.

Key Components:

- **Page Components:** Located in the `app` directory, these components define the routes and main page structures (e.g., `app/dashboard/student/page.jsx`, `app/dashboard/admin/page.jsx`)
- **UI Components:** Reusable UI elements in the `components/ui` directory providing consistent styling and behavior
- **Layout Components:** Components that define the overall page structure, including navigation and sidebars (e.g., `components/layout/dashboard-layout.jsx`)
- **Feature-specific Components:** Components for specific features like attendance tracking, profile management, etc.

Technologies:

- React.js for component-based UI development
- Tailwind CSS for styling
- shadcn/ui for UI component library
- Next.js App Router for routing and navigation

The presentation layer communicates with the business logic layer through:

- Server Components that fetch data during rendering
- Client-side API calls to backend endpoints
- Form submissions that trigger server actions

2) Business Logic Layer (Application Tier): The business logic layer handles the core application logic, processing requests from the presentation layer and interacting with the data layer.

Key Components:

- **API Routes:** Located in `app/api` directory, these endpoints handle client requests, implement business rules, and interact with the database
- **Server Actions:** Functions that execute on the server to process form submissions and other user actions
- **Middleware:** Authentication and authorization logic in `middleware.js` that protects routes based on user roles
- **Utility Functions:** Helper functions for common operations like date formatting, validation, etc.

Technologies:

- Next.js API Routes for RESTful endpoints
- Server Actions for form processing
- Middleware for request processing and authentication

The business logic layer enforces:

- Data validation and sanitization
- Business rules and workflows
- Authentication and authorization
- Error handling and logging

3) Data Access Layer (Data Tier): The data access layer is responsible for interacting with the database and other external services.

Key Components:

- **Supabase Client:** Database access utilities in `utils/supabase` directory
- **Data Models:** Implicit data models defined through database queries
- **Data Access Functions:** Functions that abstract database operations

Technologies:

- Supabase for database and authentication services
- Server-side Supabase client for secure database access
- Client-side Supabase client for real-time updates

The data layer handles:

- CRUD operations on database tables
- Data retrieval and filtering
- Transaction management
- Data integrity enforcement

4) Cross-Cutting Concerns: Several components span across all three tiers:

- **Authentication and Authorization:**
  - Supabase Auth for user authentication
  - Middleware for route protection
  - Role-based access control
- **Error Handling:**
  - Client-side error boundaries
  - Server-side error handling
  - Error logging and reporting
- **Performance Optimization:**
  - Server-side rendering for initial load
  - Client-side navigation for subsequent navigation
  - Image optimization
  - Code splitting
- **Security:**
  - CSRF protection
  - Secure cookie handling
  - Environment variable management
  - Input validation

VII. Deployment Process

A. Build Process

The deployment process leverages Vercel's integration with Next.js, ensuring an efficient and optimized build. Key steps include:

- Code is compiled and optimized during the build phase.
- Pages are pre-rendered where applicable to improve performance and SEO.
- API routes are compiled into serverless functions for efficient handling of backend requests.
- Static assets are optimized and prepared for CDN delivery to ensure fast content loading times.

B. Environment Configuration

The environment configuration process ensures that the application runs smoothly across all stages (production, preview, development):

- Environment variables are securely stored in Vercel for proper configuration management.
- Production, preview, and development environments have separate configurations to ensure seamless transitions between stages.
- Sensitive information such as Supabase credentials is stored securely as environment variables to prevent unauthorized access.

C. Performance Optimizations

To ensure the application runs optimally, the following performance optimizations are applied:

- `Vercel Speed Insights` is integrated for performance monitoring, helping to identify and address potential bottlenecks.
- Image optimization is configured to adapt content to various device sizes, improving loading times on mobile and desktop devices.
- CSS and JavaScript files are minified and optimized to reduce load times and improve user experience.
- Package imports are optimized to reduce bundle size, resulting in faster initial loading and quicker navigation.

D. Security Measures

Security best practices are enforced across the deployment process to protect the application and its users:

- HTTPS is enforced for all connections to ensure secure communication between clients and servers.
- Headers are configured to follow security best practices, preventing common vulnerabilities.
- The `Powered-By` header is disabled to prevent information disclosure about the server.
- A Strict Content Security Policy (CSP) is implemented to mitigate risks such as cross-site scripting (XSS) attacks.

E. Monitoring and Analytics

To monitor and track system performance and usage, the following tools are employed:

- `Vercel Speed Insights` is used for ongoing performance monitoring to ensure the platform remains fast and responsive.
- Error tracking and reporting are integrated to detect and resolve any issues promptly.
- Usage analytics help assess user interaction with the platform, providing insights into areas for improvement.

The deployed website link is `[https://csoft-vert.vercel.app/]`.

VIII. Conclusion

The Career Services Online Facilitation Tool (CSOFT) represents a sophisticated implementation of modern web development practices, integrating a robust three-tier architecture with Next.js's hybrid rendering capabilities. This comprehensive career services management platform effectively supports the needs of students, administrators, and career services staff.

By clearly separating presentation, business logic, and data access concerns, while leveraging Supabase for authentication and database services, CSOFT achieves both maintainability and scalability. The deployment strategy utilizing Vercel's serverless infrastructure ensures optimal performance, security, and reliability through automated scaling, global CDN distribution, and continuous deployment workflows.

This architectural approach not only facilitates current operational needs with role-specific interfaces and comprehensive feature sets but also establishes a solid foundation for future enhancements. As user demands evolve, the system can scale efficiently to accommodate increased traffic and feature expansion, ensuring a seamless experience for all users involved in career services management.

In conclusion, CSOFT offers an innovative, secure, and high-performance solution that addresses the complex needs of career services in educational institutions, positioning itself as a reliable tool for long-term success and growth.

REFERENCES

[1] A. McIntosh, "How digital transformation efforts foster student success in higher ed," *EdTech Magazine*, 2024, retrieved from https://edtechmagazine.com/higher/article/2024/12/how-digital-transformation-efforts-foster-student-success-higher-ed.

[2] The Daily Texan Editorial Board, "Students deserve a centralized career services system," 2019, retrieved from https://thedailytexan.com/2019/01/25/students-deserve-a-centralized-career-services-system/.

[3] Inside Higher Ed, "Best-in-class career services," 2023, retrieved from https://www.insidehighered.com/sites/default/files/2023-05/Benchmarking%20Best-in-Class%20Career%20Services%20%281%29.pdf.

[4] EDUCAUSE, "2022 educause horizon report: Technology trends in higher education," 2022, retrieved April 28, 2025, from https://er.educause.edu/channels/emerging-technologies-trends.

[5] D. O. Mireku, B. Bervell, P. D. Dzamesi, and E. B. Nimo, "A systematic review of students' support services provision in higher education (ssspihe) in sub-saharan africa," *Cogent Education*, vol. 11, no. 1, p. 2351747, 2024.

[6] World Bank, "Digital transformation drives development in africa," 2024, retrieved April 28, 2025, from https://projects.worldbank.org/en/results/2024/01/18/digital-transformation-drives-development-in-afe-afw-africa.

[7] National Association of Colleges and Employers (NACE), "The gap in perceptions of new grads' competency proficiency and resources to shrink it," 2021, retrieved April 28, 2025, from https://www.naceweb.org/career-readiness/competencies/the-gap-in-perceptions-of-new-grads-competency-proficiency-and-resources-to-shrink-i

[8] CareerSpaces, "Careerspaces: Ai-powered career guidance platform," 2025, retrieved April 28, 2025, from https://www.studenthub.africa/.

[9] ReadyforWork Africa, "Readyforwork: Learn in-demand digital skills," 2025, retrieved April 28, 2025, from https://readyforwork.africa/.