BILINGUAL PUBLISHING GROUP

Journal of Electronic & Information Systems

https://journals.bilpubgroup.com/index.php/jeis

ARTICLE

Development of an Optimized Frontend User Interface for Real-Time Student Continuous Assessment Performance Tracking: A User-Centered Approach

Charles Udekwe [*] ***, Ebenezer Agboola***

Information and Communication Engineering Department, Federal University of Technology, Ondo State 340110, Nigeria

ABSTRACT

This research presents the design and evaluation of Track-CA, a frontend-based educational performance tracking platform tailored for continuous assessment in tertiary institutions. The system focuses on optimizing student and lecturer interactions through a structured, intuitive user interface underpinned by the Breadth-First Search (BFS) algorithm. BFS was applied to model and streamline user navigation paths across key sections such as onboarding, course tracking, assignment management, and performance visualization. Additionally, the interface design was guided by two theoretical frameworks: Self-Determination Theory (SDT), which emphasizes motivation through timely feedback, and the Task-Technology Fit (TTF) model, ensuring effective alignment between user tasks and interface structure. The platform was deployed and tested on Vercel, a serverless hosting environment, to simulate real-world performance. Empirical user testing revealed a 22% reduction in average navigation steps, a 17% improvement in task completion times, and a notable decrease in user confusion, particularly during multi-layered tasks like course material uploads and analytics access. Wireframes for both student and lecturer interfaces were created and tested iteratively, showcasing improved responsiveness, consistency, and usability. Despite its strengths, this work is currently limited to frontend implementation. Future work will incorporate backend functionalities, database integration, and advanced algorithms for real-time analytics and feedback. Nevertheless, this study offers a strong foundation for scalable, student-centered assessment platforms that enhance learning outcomes through efficient and intuitive digital interfaces.

Keywords: Continuous Assessment; Real-time Academic Tracking; Vue.js Frameworks; Breadth First Search Algorithm; Self-Determination Theory; Task-Technology Fit Model

*CORRESPONDING AUTHOR:
Charles Udekwe, Information and Communication Technology Department, Federal University of Technology, Ondo State 340110, Nigeria; Email: cnudekwe@futa.edu.ng

ARTICLE INFO
Received: 13 February 2025 | Revised: 3 March 2025 | Accepted: 25 March 2025 | Published Online: 2 April 2025
DOI: https://doi.org/10.30564/jeis.v7i1.9375

CITATION
Udekwe, C., Agboola, E., 2025. Development of an Optimized Frontend User Interface for Real-Time Student Continuous Assessment Performance Tracking: A User-Centered Approach. Journal of Electronic & Information Systems. 7(1): 72–97. DOI: https://doi.org/10.30564/jeis.v7i1.9375

1. Introduction

Assessment constitutes a foundational pillar of the teaching-learning process, serving multifaceted roles in educational ecosystems. Contemporary pedagogical practice employs diverse assessment modalities—ranging from high-stakes examinations (mid-term, end-of-term, and annual exams) to continuous evaluations (assignments, projects, seminar presentations, and debates). These mechanisms are designed to fulfill critical functions: motivating learners, providing actionable feedback to both students and instructors, measuring academic progress, informing curriculum refinement, diagnosing learning gaps, and fostering collaborative learning environments [1–3].

In today's data-driven educational landscape, the continuous evaluation of student learning outcomes has become an essential component of effective teaching and academic accountability. *Continuous assessment* (CA) refers to an ongoing, systematic process where educators regularly evaluate students' progress through various academic activities, including tests, assignments, classroom participation, and projects [4]. Unlike summative assessments, continuous assessment provides timely feedback, enabling both teachers and students to make informed decisions aimed at improving learning outcomes. Since its adoption by Nigeria's National Policy on Education in the early 1980s, continuous assessment has been recognized as a vital strategy for tracking student performance and guiding curriculum delivery [5].

Despite its pedagogical value, the implementation of continuous assessment in many higher institutions faces significant challenges, especially in environments with large student populations. Lecturers often struggle with compiling and analyzing numerous scores across multiple assessment types, which can result in delays, errors, and limited transparency. This burden reduces the efficiency and effectiveness of the evaluation process and often hinders timely feedback to students. The increasing availability and adoption of internet technologies present a promising opportunity to address these challenges [6]. For example, the Federal University of Technology Akure (FUTA), Nigeria, faces significant challenges due to the absence of a real-time academic performance tracking system. Currently, students experience delays in receiving test results, which hinders their ability to gauge their academic progress effectively. Lecturers often assign mini-projects and capstone projects without providing timely feedback or grades, and students lack clear information on how attendance affects their overall performance. The delay in grading and feedback for assignments exacerbates these issues, leading to disconnect between students' academic activities and their performance assessments. This situation impedes students' ability to monitor their performance in real-time, which is crucial for making timely adjustments and avoiding potential academic failures. As a result, the absence of an efficient tracking and feedback system poses a significant barrier to student success and optimal academic performance at FUTA. Furthermore, the urgency of digital transformation in education was further underscored by the global outbreak of COVID-19, which disrupted traditional in-person education and forced institutions to pivot towards online learning and evaluation. This shift has exposed both the potential and the gaps in existing assessment systems, particularly in developing countries where the adoption of educational technology remains inconsistent [7]. The need for user-friendly, responsive, and accessible interfaces for continuous assessment tracking has never been greater.

Web-based systems for academic evaluation have emerged as powerful tools for streamlining the administration, tracking, and reporting of student performance. In particular, real-time assessment tracking platforms allow for more dynamic interaction between students and lecturers, enhancing learning engagement and promoting academic accountability [8]. Yussuff et al. (2024) developed a web-based continuous assessment management system to address challenges in manual grading, particularly in large classes. Leveraging **HTML, CSS, JavaScript, and Laravel**, their system streamlines assessment tasks (e.g., score uploads, retrieval) while enhancing flexibility over paper-based methods. Key strengths include **ICT-driven efficiency, Browser/Server architecture** for seamless access, and reduced administrative workload. However, the study lacks optimization for frontend navigation and real-time feedback [9]. **Murugan (2021)** examines ICT's role in modernizing educational assessment, emphasizing continuous evaluation and real-time feedback. The study highlights ICT's potential for performance-based assessment but identifies gaps in usability optimization [10]. Chekwube and Mgbeafulike (2021) developed an integrated e-assessment system to address vulnerabilities in traditional paper-based exams, including leakage risks and grading errors [11]. Their web-based solution using HTML, PHP, and MySQL automates assessment management while ensuring faster, more secure result processing. While demonstrating improvements in workload reduction and feedback speed, the study focuses primarily on backend functionality rather than frontend usability [11].

The digitization of education has transformed traditional assessment methods, yet significant challenges persist in continuous assessment systems within higher education. Current learning management platforms often exhibit critical limitations, including delayed feedback mechanisms, suboptimal user interfaces, and inefficient navigation structures that increase cognitive load. These shortcomings are particularly problematic in contexts requiring real-time performance tracking, where delayed feedback can negatively impact student motivation and learning outcomes. While some modern systems incorporate analytics capabilities, few address the fundamental usability challenges posed by complex, multi-level assessment data navigation. Some learning management system (LMS) platforms like Moodle and Blackboard, although comprehensive, are not specifically designed for continuous assessment. Automated grading tools such as Gradescope focus more on end of term evaluations, and web-based exam platforms like ProctorU are limited to high-stakes assessments rather than ongoing evaluations [12]. Custom-built SMS platforms, while integrating multiple school functions, often lack the specialized focus on continuous assessment required for detailed performance tracking [13]. This research gap presents an opportunity to develop optimized interfaces that leverage modern web technologies and algorithmic approaches to enhance both functionality and user experience.

This study introduces an innovative frontend solution designed to overcome these limitations through three key contributions. First, we implement a responsive, component-based architecture using Vue.js 2 and Vuetify, ensuring cross-device compatibility and adherence to Material Design principles [14,15]. Second, we innovate in interface navigation by applying Breadth-First Search (BFS) algorithms to optimize menu traversal in hierarchical assessment structures, significantly reducing navigation complexity [16,17]. Third, we establish a robust theoretical foundation by integrating Self-Determination Theory and the Task-Technology Fit model, demonstrating how our design enhances autonomy, competence, and engagement while ensuring alignment with user workflows [18]. The theoretical framework underpinning this work draws from two complementary perspectives. Self-determination theory informs our approach to real-time feedback visualization, addressing the well-documented relationship between timely assessment and student motivation. Concurrently, the Task-Technology Fit model guides our interface design decisions, particularly in optimizing the alignment between navigation structures and user tasks. This dual theoretical lens not only validates our technical approach but also contributes to ongoing discussions about human-computer interaction in educational technology [19].

Our methodology combines iterative UI/UX design processes with optimized search process using the breadth-first-search algorithm and the vue.js frameworks and deployed in vercel web app. The resulting system demonstrates measurable improvements over traditional interfaces, particularly in reducing navigation complexity and enhancing real-time data visibility. These advancements have significant implications for educational practice, offering a model for developing assessment systems that are both technically robust and pedagogically effective. The findings contribute to the growing body of research on educational technology optimization while providing practical insights for institutions seeking to enhance their digital assessment infrastructure

2. Materials and Tools Used

This section details the technical framework, and tools, employed in developing the optimized frontend interface for real-time continuous assessment tracking. The implementation combines modern web technologies, design tools, and algorithmic optimization to address the limitations of existing learning management systems (LMS).

Here are the tools and frameworks used in implementing this project.

2.1. Frontend Development Frameworks

The frontend development tools used to implement this project are discussed as follows:

2.1.1. Vue.js

Vue.js served as the foundational JavaScript framework for developing the reactive, component-based user interface in this project [20]. The implementation leveraged several key features of Vue.js to enhance functionality and performance. Single-File Components (SFCs) were employed to create modular *.vue* files that encapsulated HTML templates, CSS styling, and JavaScript logic in cohesive units, exemplified by components such as *GradeCard.vue* and *AssessmentTable.vue*. The framework's reactivity system enabled automatic Document Object Model (DOM) updates whenever assessment data changed, facilitating real-time adjustments to grades and other critical information. Additionally, Vue.js's Virtual DOM implementation significantly optimized rendering performance for dynamic content updates. Compared to alternative frameworks, Vue.js 3 offered distinct advantages for this educational application, including a lighter architecture than Angular and greater flexibility than React, making it particularly suitable for rapid prototyping. Its seamless integration capabilities with both Vuetify for UI components and Vuex for state management further reinforced its selection as the optimal framework for this continuous assessment system.

2.1.2. Vuetify

Vuetify was implemented as the Material Design component framework to deliver a consistent and responsive user interface throughout the application [21]. The system strategically utilized several of Vuetify's pre-built components to enhance functionality and user experience. The <v-data-table> component was employed to present assessment results with integrated sorting and pagination capabilities, enabling efficient data visualization. For navigation, the <v-navigation-drawer> component formed the foundation of the BFS-optimized menu system, providing intuitive access to various assessment features. Student performance dashboards were constructed using the flexible <v-card> component, which offered modular containers for organizing and displaying academic metrics. Compared to traditional CSS approaches, Vuetify provided significant advantages by automatically handling responsive design requirements, eliminating the need for manual implementation of mobile breakpoints and cross-device styling. Furthermore, the framework's built-in compliance with Web Content Accessibility Guidelines (WCAG) ensured the interface met modern accessibility requirements without additional development effort, making it particularly suitable for educational environments serving diverse user needs.

2.1.3. Vercel

Vercel is a cloud platform designed for frontend frameworks and static sites, built to enable fast, global deployment with minimal configuration [22]. It's particularly optimized for modern JavaScript frameworks like **React, Next.js, Vue, and Svelte**, and supports **CI/CD**, serverless functions, and instant previews.

2.2. Design and Prototyping Tool: Figma

Figma served as the primary design and prototyping tool for developing the system's user interface, enabling both wireframing and high-fidelity prototyping [23]. The platform was instrumental in creating detailed mockups of key application components, including student and lecturer dashboards featuring real-time grade visualization interfaces. These designs were further refined into interactive prototypes that simulated the BFS-optimized menu navigation flow, allowing for early usability testing and validation of the proposed information architecture. A critical aspect of the Figma implementation was its seamless integration with the development workflow - specifically, the exported CSS variables were carefully structured to align with Vuetify's theming system, ensuring design consistency between prototypes and the final implemented interface. This integration significantly streamlined the handoff process from designers to developers, maintaining visual fidelity while enabling efficient translation of design concepts into functional Vue.js components. The tool's collaborative features also facilitated iterative improvements based on stakeholder feedback throughout the development lifecycle, particularly in refining the assessment data visualization components and navigation patterns.

2.3. Algorithm Tool: Breadth-First-Search (BFS) Algorithm

The Breadth-First Search (BFS) algorithm was implemented to optimize navigation efficiency within the system's multi-level settings panel [24]. This graph traversal approach models the menu architecture as a directed graph, where individual pages serve as nodes and their hierarchical relationships as edges. The algorithm's level-order traversal capability enables intuitive access to nested menu tiers while maintaining a single-interface design. Within the Vue.js framework, this was achieved through strategic use of v-for directives to dynamically render menu items without requiring full page reloads. By prioritizing breadth-wise expansion over depth-first exploration, the BFS implementation ensures users encounter the most frequently accessed settings first, significantly reducing navigation time compared to traditional nested menu systems. This approach proved particularly effective for the assessment system's complex configuration requirements, where lecturers need rapid access to diverse settings across multiple levels of the interface hierarchy.

2.4. Theoretical Design Philosophy Models

2.4.1. Self-Determination Theory (SDT)

Self-Determination Theory is a psychological framework that focuses on human motivation and well-being [25]. It posits that individuals are most motivated and engaged when three basic psychological needs are met: **autonomy** (feeling in control of one's actions), **competence** (feeling effective in one's activities), and **relatedness** (feeling connected to others). In the context of the Track-CA platform, SDT informed the design of features that support student autonomy through self-paced interaction, competence through timely performance feedback, and relatedness via communication features integrated within the learning system.

2.4.2. Task-Technology Fit (TTF)

The Task-Technology Fit model evaluates how well a technology supports the tasks it is intended to assist [26]. It suggests that a system will positively impact performance only when the capabilities of the technology align with the requirements of the task. In designing the Track-CA platform, TTF guided the structuring of navigation, layout, and features to ensure that each component directly supported user goals—such as course tracking, assignment submission, and feedback analysis—thereby enhancing usability and effectiveness.

3. Methodology

This section outlines the approach and systematic processes adopted to design and develop an optimized wireframe frontend user interface for real-time tracking of student continuous assessment performance. The methodology follows a user-centered design approach, ensuring that the resulting interface meets the needs, expectations, and interaction preferences of its primary users—students and lecturers.

3.1. Student Interface Wireframe Design

The student interface was broken down into several functional screens representing key tasks within the real-time continuous assessment tracking workflow. These screens were developed in sequential order to mirror the actual user flow. **Figure 1** shows the tree diagram for the student interface wireframe design.

Each component is described as follows:

(i). Onboarding and Login Process Flow: Upon launching the *Track-CA* application, students are presented with a clean and minimalistic login interface requiring their *Matric No./Registration No.* and *password*. A "*Forgot Password*?" link facilitates account recovery, enhancing usability. Students enter their credentials and proceed by clicking the "*Login*" button, while forgotten credentials initiate a password recovery process to ensure smooth access and improve user retention.

(ii). First-Time Users: Sign-Up Process: First-time users complete a guided sign-up form requesting personal and academic information, including name, email, institution name, matriculation number, and an optional mobile number. They must also set and confirm a password. Upon submission, the system validates the input and creates a new user account.

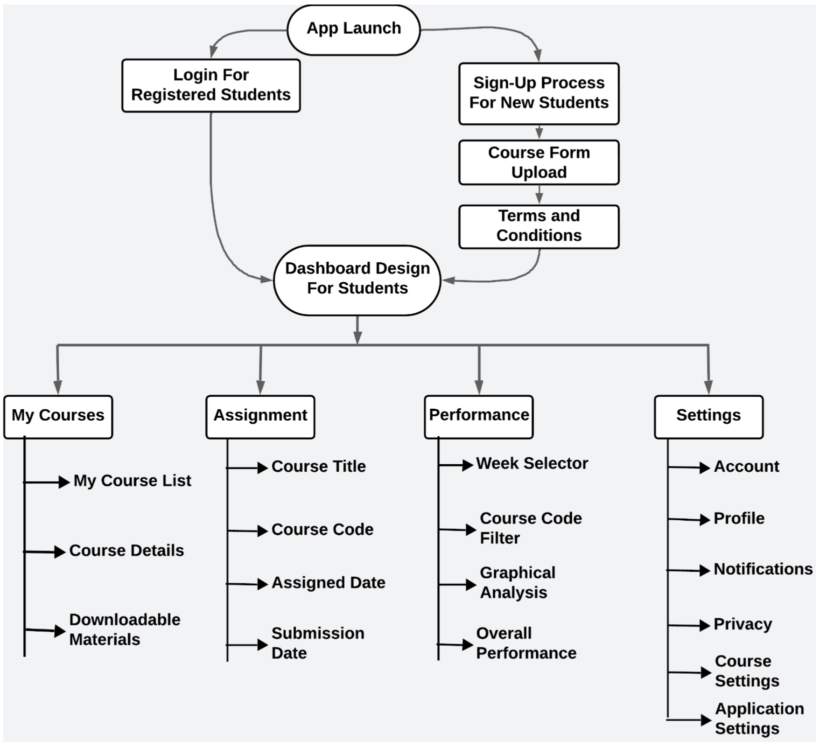


Figure 1. Tree Diagram Showing Major Sections of the Student Frontend Interface.

(iii). Course Form Upload: After successful registration, students are prompted to upload their course form, which enables the system to map and track their CA scores. Users select the file from local storage and upload it through a dedicated interface, where the system securely stores the course data for subsequent reference and monitoring.

(iv). Terms and Conditions Agreement: Before gaining full access to the application, students are required to review and accept the terms and conditions. They indicate their consent by checking the "*I agree*" box and clicking the continue button to proceed.

(v). Student Dashboard Design: Upon successful login, students are directed to a central dashboard that provides a comprehensive overview of their academic activities. A left-aligned sidebar offers navigation to key sections including Dashboard, My Courses, Assignments, Performance, and Settings, while the main area displays enrolled courses, performance summaries, and upcoming assignment deadlines.

(vi). Course Overview and Progress Tracking: Under the "*My Courses*" tab, students can view all their enrolled courses, with each course card displaying the title, course code, lecturer name, and progress percentage. A sorting feature enables users to organize courses either alphabetically or based on their performance progress.

(vii). Detailed Course View: Selecting a course provides students with detailed performance metrics, such as current scores, continuous assessment breakdowns, and overall course progress. Additionally, course materials are available for direct download from this view.

(viii). Assignment Management and Notifications: The assignment page lists all active and past assignments linked to the student's courses, displaying each assignment's title, due date, and submission status. It also provides direct links for submitting new work or reviewing past submissions.

(ix). Settings Page: The settings page allows students to customize their accounts by updating profile information, changing passwords, and managing notification preferences. All changes are saved in real-time to enhance the personalized user experience.

(x). Performance Tracking Interface: The performance tracking page offers students a graphical and data-driven overview of their academic progress. It features tools such as a week selector, course code filter, and performance charts that visualize continuous assessment trends, highlighting areas of strength and weakness.

3.2. Wireframe Design for the Lecturer Interface

This section details the methodology used to design the frontend user interface for lecturers on the Track-CA application. The primary goal is to provide lecturers with an intuitive platform for managing courses, tracking student performance, and overseeing assignments in real time. All wireframes were also designed using **Figma**, aligning with usability principles to ensure an efficient and streamlined user experience. These screens were developed in sequential order to mirror the actual user flow. **Figure 2** shows the tree diagram for the lecturer's interface wireframe design.

Each component is described as follows:

(i). Sign-Up Process: The onboarding process for lecturers begins with a secure and structured sign-up flow, which ensures authenticated access to the Track-CA system and supports profile personalization. During registration, lecturers are required to provide their full name, email address, institution name, and an optional mobile number, along with a chosen password. Once the form is submitted, the system automatically validates the provided information and creates a unique lecturer profile.

(ii). Terms and Conditions Agreement: To ensure compliance and uphold platform accountability, lecturers are required to agree to the system's terms and conditions before gaining access to any Track-CA functionalities. This step is a mandatory part of the onboarding process and serves to establish mutual understanding of usage policies and data responsibilities. Lecturers indicate their consent by checking the "I Agree" box and clicking the corresponding confirmation button, after which they are granted access to the full suite of platform features.

(iii). Login Process: Following successful registration, lecturers can log in to the Track-CA platform using their registered email and password. A "Forgot Password?" option is available to facilitate account recovery when needed. Upon successful credential verification, the system grants access and redirects the lecturer to their personalized dashboard.

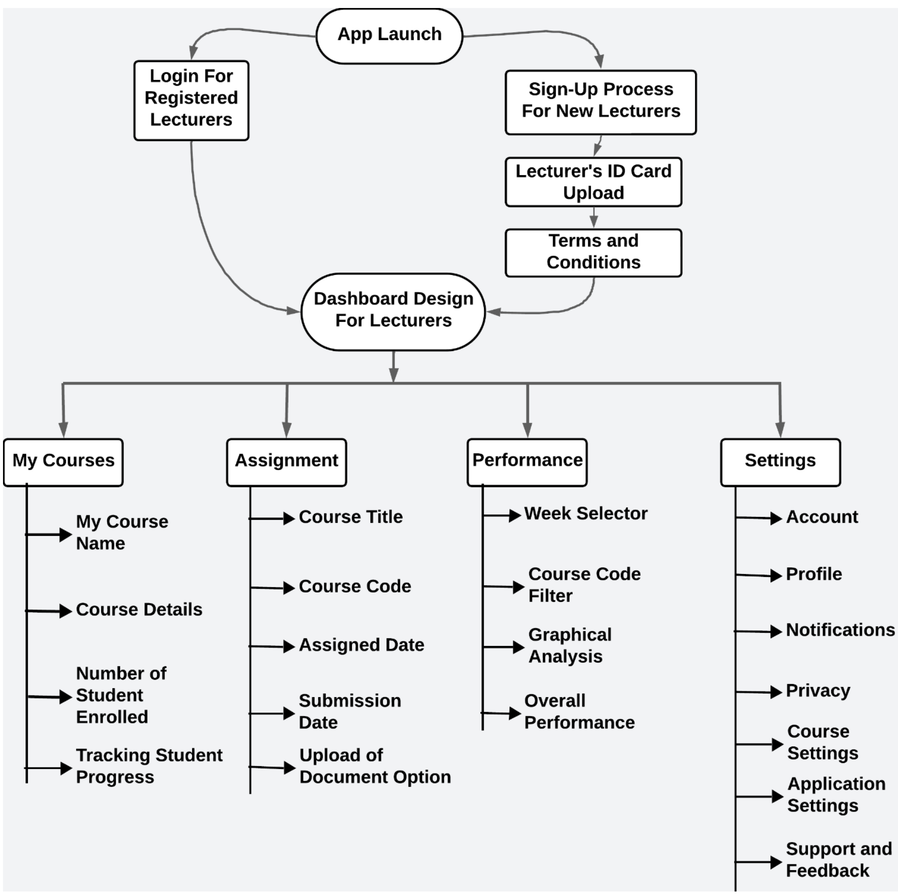


Figure 2. Tree Diagram Showing Major Sections of the Lecturer's Frontend Interface.

(iv). Dashboard Overview: The dashboard functions as the central hub for lecturers, providing quick access to academic information, platform announcements, and weekly teaching schedules. Its layout features a top navigation bar with notification and profile icons, while the main display area highlights current announcements and scheduled activities for the week.

(v). Notifications: The system delivers real-time alerts for key events such as assignment submissions, schedule updates, and administrative announcements. Each notification is interactive—lecturers can click to view more details or dismiss them as needed, ensuring they stay informed and responsive to time-sensitive tasks.

(vi). My Courses Page: Lecturers can manage all their active courses through this dedicated interface, which is easily accessible via the sidebar navigation. Each course entry displays key details, including the course name, code, unit count, and the number of enrolled students. From this page, lecturers can perform essential functions such as editing course information, uploading instructional materials, and entering or updating student assessment records.

(vii). Tracking Student Progress: This feature offers detailed insights into each student's academic performance within a specific course. The overview includes the student's name, matriculation number, a visual CA performance bar, and a breakdown of weekly metrics such as attendance, assignments, and test scores. Performance data is presented in a structured assessment table divided into Student Score (SS) and Lecturer Score (LS) columns, summarizing Raw CA, Moderated CA, and Final Exam scores. Lecturers also have the flexibility to modify individual weekly records or update the entire table to ensure accurate reporting.

(viii). Assignment Management: The assignment page enables lecturers to efficiently create, update, and delete assignments as needed. It supports the upload of relevant documents, such as PDFs and marking schemes, while also providing a clear overview of each assignment's submission status and due dates. This centralized interface streamlines assignment handling and enhances instructional organization.

(ix). Settings and Profile Management: This module offers user-specific customization options and access to support services. Lecturers can update their personal information under account management, adjust notification preferences, and configure how courses are displayed through the course settings panel. Additionally, the module provides channels to contact the helpdesk or submit feedback, ensuring continuous improvement and user satisfaction.

(x). Performance Tracking Dashboard: This feature equips lecturers with tools to analyze both overall class performance and individual student metrics. It includes filters for selecting specific date ranges and courses, as well as comparative charts that display individual student performance against the class average. A built-in search bar allows for quick lookup of specific students. This dashboard supports timely intervention by helping lecturers identify underperforming students early and respond with appropriate academic support.

3.3. Administrator Interface and Role Integration in Student and Lecturer Frontends

The Administrator plays a critical role in the ***Track-CA* platform** by managing core backend and frontend operations that facilitate the interactions of both students and lecturers. While students and lecturers interact primarily with user-facing functions, administrators ensure the proper configuration, control, and validation of these interfaces to maintain data integrity, workflow efficiency, and user accountability. The Administrator functionalities are integrated into the student and lecturer modules but accessed via a secure backend interface, with limited visibility to standard users. However, several functions and actions reflected in the student and lecturer frontends originate from the administrator's configurations and validations.

3.3.1. Administrator Functions Relevant to the Student Interface

The following functions are managed or facilitated by the Administrator to ensure smooth operation and consistency of the student-facing platform.

(i). Student Account Verification and Activation: Student registration data including name, email, matriculation number, and institution is reviewed to verify authenticity. Accounts are then approved, rejected, or flagged for edits to ensure that only verified students gain access to the platform.

(ii). Matric Number Mapping and Cohort Assignment: The admin maps registered students to their respective academic year, department, and registered courses using institutional records to ensure accurate cohort assignment and data organization.

(iii). Course Enrollment Authorization: The administrator verifies that students are enrolled exclusively in courses appropriate for their program level. This is managed by uploading approved course lists or integrating the system with the institution's database via API.

(iv). Timetable and Weekly Schedule Configuration: The administrator configures the academic calendar, weekly schedules, and continuous assessment milestones, which are subsequently displayed on students' dashboards for effective time and activity management.

(v). Student Performance Audit Trail: The administrator maintains a secure audit log to track all changes to student continuous assessment records, enabling detection of unauthorized edits or anomalies in scores.

(vi). Access and Privilege Management: The administrator has the authority to suspend or restrict student access to the platform in cases of disciplinary actions or academic probation.

(vii). Helpdesk and Support Coordination: The administrator handles student complaints, password reset requests, and technical issues through a support ticket system or chatbot integration to ensure timely assistance.

3.3.2. Administrator Functions Relevant to the Lecturer Interface

The administrator ensures that lecturer functionalities are standardized, monitored, and compliant with institutional policies. Key roles include:

(i). Lecturer Account Approval and Institution Validation: The administrator verifies lecturer registration details, including institutional affiliation, email, and course responsibilities, before granting access privileges aligned with their verified teaching roles.

(ii). Assignment Workflow Oversight: The administrator oversees assignment uploads, deadlines, and submission reports, employing flagging mechanisms to detect plagiarism or irregular grading patterns for timely intervention.

(iii). CA Score Moderation Workflow: The administrator reviews continuous assessment scores submitted by lecturers and moderates them according to departmental policies by applying set thresholds and moderation coefficients before granting final approval.

(iv). Communication Management: The administrator manages system-wide notifications and announcements targeted at lecturers, which may include training alerts, audit reminders, and updates on institutional policies.

(vii). Feedback and Evaluation Analysis: The administrator reviews feedback from lecturers and students on course effectiveness, platform usability, and system issues, prioritizing critical concerns for escalation to development teams or institutional management.

(viii). Role-Based Access Control (RBAC): The administrator assigns role-based permissions to lecturers, granting varied access levels such as CA input-only, full grading rights, or view-only access for assistant lecturers.

3.3.3. Cross-Platform Administrative Features

In addition to student- and lecturer-specific roles, the admin also manages global platform functions:

(i). System Configuration Settings: Academic sessions, semester activation, maintenance windows.

(ii). User Activity Logs: Tracks login sessions, data changes, and dashboard access patterns for audit and security purposes.

(iii). Backup and Data Recovery: Regular export of student and lecturer data to avoid loss due to server failure.

(iv). Report Generation: Produces departmental-level or institution-wide reports on CA progress, course completion rates, and performance summaries.

(v). Security and Compliance Monitoring: Enforces password policies, data encryption standards, and platform compliance with educational data protection laws.

3.4. Breadth-First Search (BFS) Integration for Student Interface Optimization

To improve student interaction efficiency and responsiveness in the *Track-CA* platform, the Breadth-First Search (BFS) algorithm was implemented for structured content traversal. The student interface is modeled as a multi-level directed graph where pages (nodes) are linked via user actions (edges). BFS traversal supports level-wise access, ensuring the system prioritizes early visibility of major components before diving into nested pages. This methodology enhances: Rendering and loading efficiency, User experience through reduced navigation complexity, and Logical data fetching order. The tree structure of the student interface, shown in **Figure 1**, was used to simulate BFS traversal and guide frontend design decisions.

Six major features in the student interface, as shown in **Table 1**, were examined as a subgraph, and BFS was applied to ensure minimal click-depth and optimal UI flow.

Table 1. Sections of the Student Interface Optimized Using BFS.

Section	BFS Optimization Role
Onboarding and Login	BFS determines the shortest and smoothest path to full app access for new users.
Dashboard Navigation	Sidebar/menu structure was designed using BFS over a graph of features and quick-links.
Course Form Upload	BFS ensures users reach upload page in minimal clicks from login or dashboard.
Course Tracking and View	Nested navigation (e.g., Course → Details → Materials) benefits from level-order access.
Assignment Management	Optimal routes from dashboard to submission screens are modeled using BFS traversal.
Performance Visualization	BFS ensures charts and analytics are easily reachable via fewest steps.

Below are conceptual BFS traversal graphs for key interface sections:

(i). Onboarding and Login

Nodes: A: Launch App → B: Sign-Up/Login → C: Terms Agreement → D: Dashboard

BFS Vertex Order: A → B → C → D

Edges: {‘A’,’B’; ‘B’,’C’; ‘C’,’D’}

(ii). Dashboard Navigation

Nodes: D: Dashboard → E: My Courses, F: Assignments, G: Analytics

BFS Vertex Order: D → E, F, G

Edges: {‘D’,’E’; ‘D’,’F’; ‘D’,’G’}

(iii). Course Form Upload

Nodes: D: Dashboard → E: My Courses → H: Upload Form

BFS Vertex Order: D → E → H

Edges: {‘D’,’E’; ‘E’,’H’}

(iv). Course Tracking and Details

Nodes: E: My Courses → I: Course Details → J: Materials → K: Announcements

BFS Vertex Order: E → I → J, K

Edges: {‘E’,’I’; ‘I’,’J’; ‘I’,’K’}

(v). Assignment Management

Nodes: D: Dashboard → F: Assignments → L: Submit Assignment

BFS Vertex Order: D → F → L

Edges: {‘D’,’F’; ‘F’,’L’}

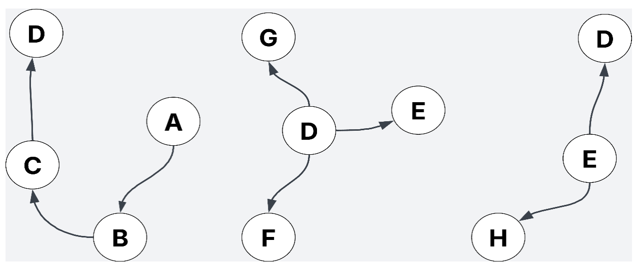
(vi). Performance Visualization

Nodes: D: Dashboard → G: Analytics → M: Score Chart → N: Progress Graph

BFS Order: D → G → M, N

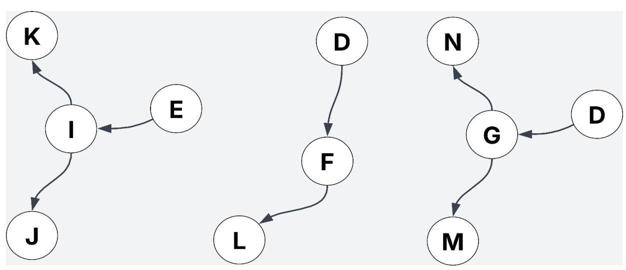
Edges: {‘D’,’G’; ‘G’,’M’; ‘G’,’N’}

The BFS graph diagrams for *Onboarding and Login* section, *Dashboard Navigation* section, and *Course form upload* section of the student interface are shown in **Figures 3 a, b** and **c** , while Figures **4 a, b** and **c** shows the BFS graph diagrams for *Course Tracking and Details* section, *Assignment Management* section and *Performance Visualization* section.



(a) (b) (c)

Figure 3. BFS Diagrams for; **(a)**. Onboarding and Login **(b)**. Dashboard Navigation **(c)**. Course Form Upload.



(a) (b) (c)

Figure 4. BFS Diagrams for; **(a).** Course Tracking and Details **(b).** Assignment Management **(c).** Performance Visualization.

3.5. Breadth-First Search (BFS) Integration for Lecturer Interface Optimization

To ensure a seamless and efficient experience for lecturers interacting with the Track-CA platform, the Breadth-First Search (BFS) algorithm was employed for user interface (UI) traversal and workflow structuring. The lecturer interface, was designed as a directed graph where: Nodes represent key pages and actions and Edges represent possible navigation steps. Applying BFS provided level-wise, structured UI navigation that allows: Immediate visibility of all course-related functionalities, efficient transition from login to task execution, predictable access patterns for core and extended features, and balanced loading and performance for complex forms and datasets.

The lecturer interface was broken into six logical subgraphs for BFS traversal as shown in **Table 2**. Each section was optimized to minimize depth and ensure logically progressive navigation.

Table 2. Sections of the Lecturer's Interface Optimized Using BFS.

Section	BFS Optimization Role
Onboarding and Login	BFS determines shortest route to teaching dashboard, bypassing redundant inputs.
Dashboard Overview	Allows top-level visibility of all courses and quick access to primary features.
Course Material Upload	Minimizes clicks to upload materials by ensuring BFS path from dashboard to upload.
Assessment Creation and Edit	Provides structured access to create → edit → view assessments per course.
Student Performance Review	Enables quick access to charts, scores, and comments without deep nesting.
Notifications and Feedback	Ensures prompt access to send or review messages and announcements.

Below are conceptual BFS traversal graphs for key interface sections:

(i). Onboarding and Login

Nodes: A: Launch App → B: Sign-In → C: Terms Agreement → D: Lecturer Dashboard

BFS Order: A → B → C → D

Edges: {'A','B'; 'B','C'; 'C','D'}

(ii). Dashboard Overview

Nodes: D: Dashboard → E: My Courses, F: Post Announcement, G: View Submissions

BFS Order: D → E, F, G

Edges: {'D','E'; 'D','F'; 'D','G'}

(iii). Course Material Upload

Nodes: E: My Courses → H: Course Page → I: Upload Materials

BFS Order: E → H → I

Edges: {'E','H'; 'H','I'}

(iv). Assessment Creation and Edit

Nodes: H: Course Page → J: Create Assessment → K: Edit Assessment → L: View Submissions

BFS Order: H → J → K, L

Edges: {'H','J'; 'J','K'; 'J','L'}

(v). Student Performance Review

Nodes: D: Dashboard → M: Analytics → N: Score Summary → O: Progress Report

BFS Order: D → M → N, O

Edges: {'D','M'; 'M','N'; 'M','O'}

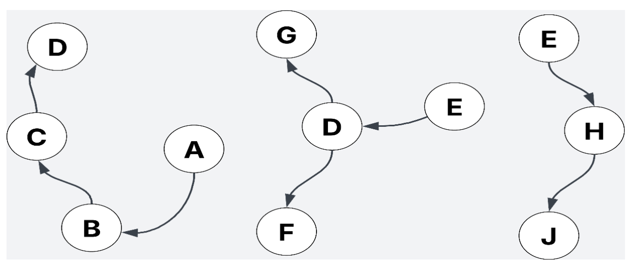
(vi). Notifications and Feedback

Nodes: D: Dashboard → P: Inbox → Q: Send Feedback → R: View Responses

BFS Order: D → P → Q, R

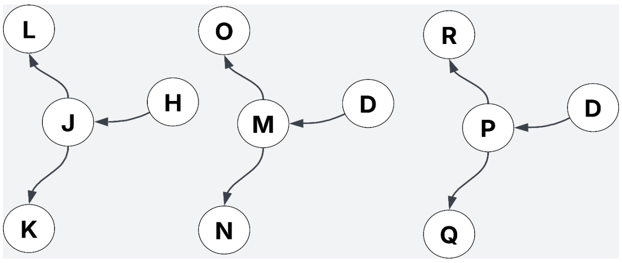
Edges: {'D','P'; 'P','Q'; 'P','R'}

The BFS graph diagrams for *Onboarding and Login* section, *Dashboard Navigation* section, and *Course Material upload* section of the lecturer's interface are shown in **Figures 5 a, b** and **c** , while Figures **6 a, b** and **c** shows the BFS graph diagrams for *Assessment Creation and Edit* section, *Student performance Review* section and *Notification and Feedback* section.



(a) **(b)** **(c)**

Figure 5. BFS Diagrams for; **(a)**. Onboarding and Login **(b).** Dashboard Overview **(c).** Course Material Upload.



(a) **(b)** **(c)**

Figure 6. BFS Diagrams for; **(a).** Assessment Creation and Edit **(b).** Student Performance Review **(c)**. Notification and Feedback.

3.6. Theoretical Underpinnings of Design Choices

To ensure that the Track-CA platform was not only technically optimized but also pedagogically impactful, two foundational theories informed our design and development approach. Self-Determination Theory (SDT) emphasizes the importance of autonomy, competence, and relatedness in fostering intrinsic motivation. This theory guided our implementation of real-time performance visualization, enabling students to receive immediate, personalized feedback—an approach known to reinforce learning engagement and motivation.

Simultaneously, the Task-Technology Fit (TTF) model shaped our decisions in UI and navigation architecture. By aligning interface components with actual user workflows (e.g., navigating from dashboard to assignment submission), we ensured that the technology appropriately supported users' task completion with minimal friction. The integration of the BFS algorithm into these navigation paths was not just a technical optimization but a strategic design choice to enhance usability as prescribed by TTF.

Together, SDT and TTF provided a dual theoretical lens that grounded our methodological choices, validating both the motivational and operational aspects of the system. This duality ensures that the platform is not only functionally robust but also responsive to the behavioral and cognitive needs of its users—contributing to broader discourse on human-computer interaction in educational technology.

4. Results and Discussion

In the context of the continuous assessment (CA) tracking app project, the results and discussion section provides the outcomes obtained during the development/design process.

4.1. Selected Wireframes Design for the Student Interface without BFS Algorithm

Figures 7 to **11** show some of the wireframes designed for the student interface without the BFS algorithm.

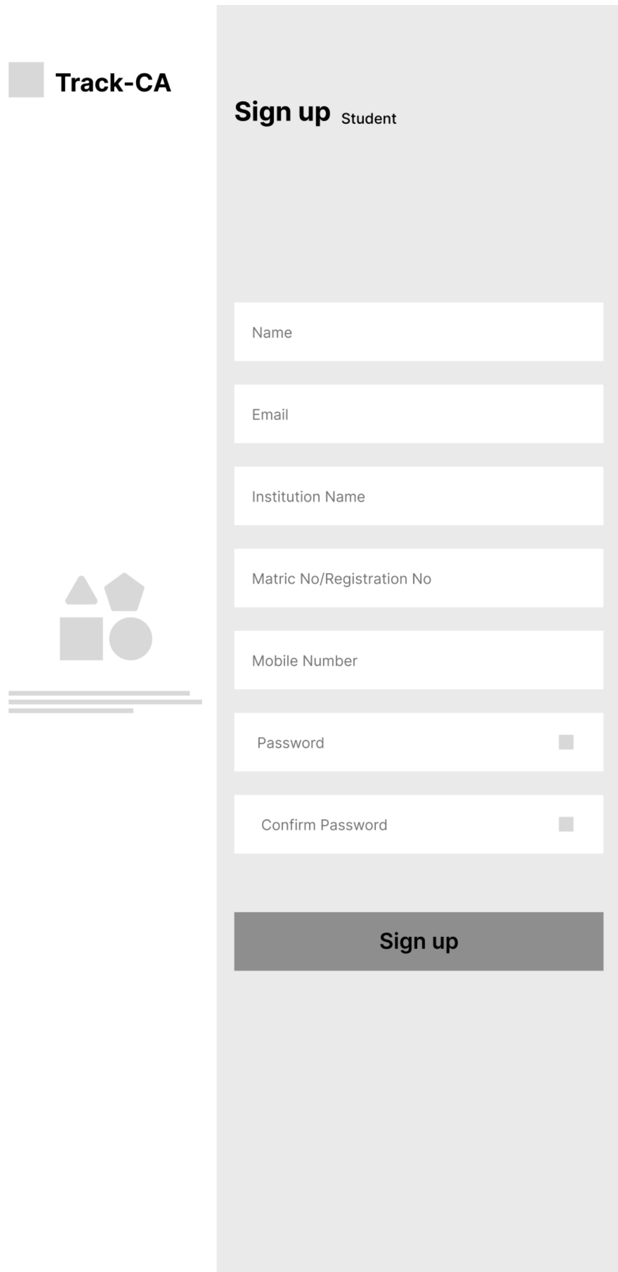


Figure 7. Wireframe for Student's Sign-Up Page.

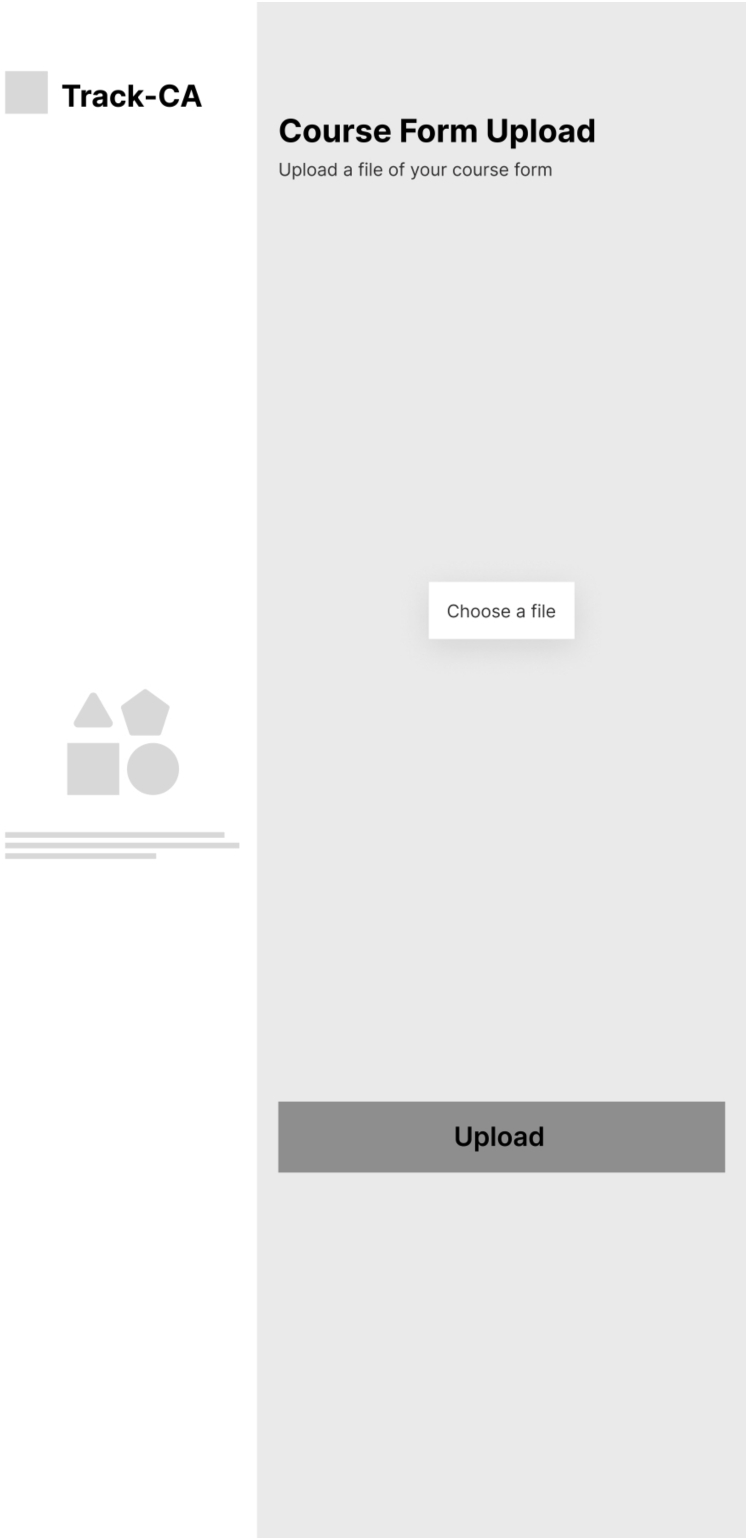


Figure 8. Wireframe For Student Course Upload Page.

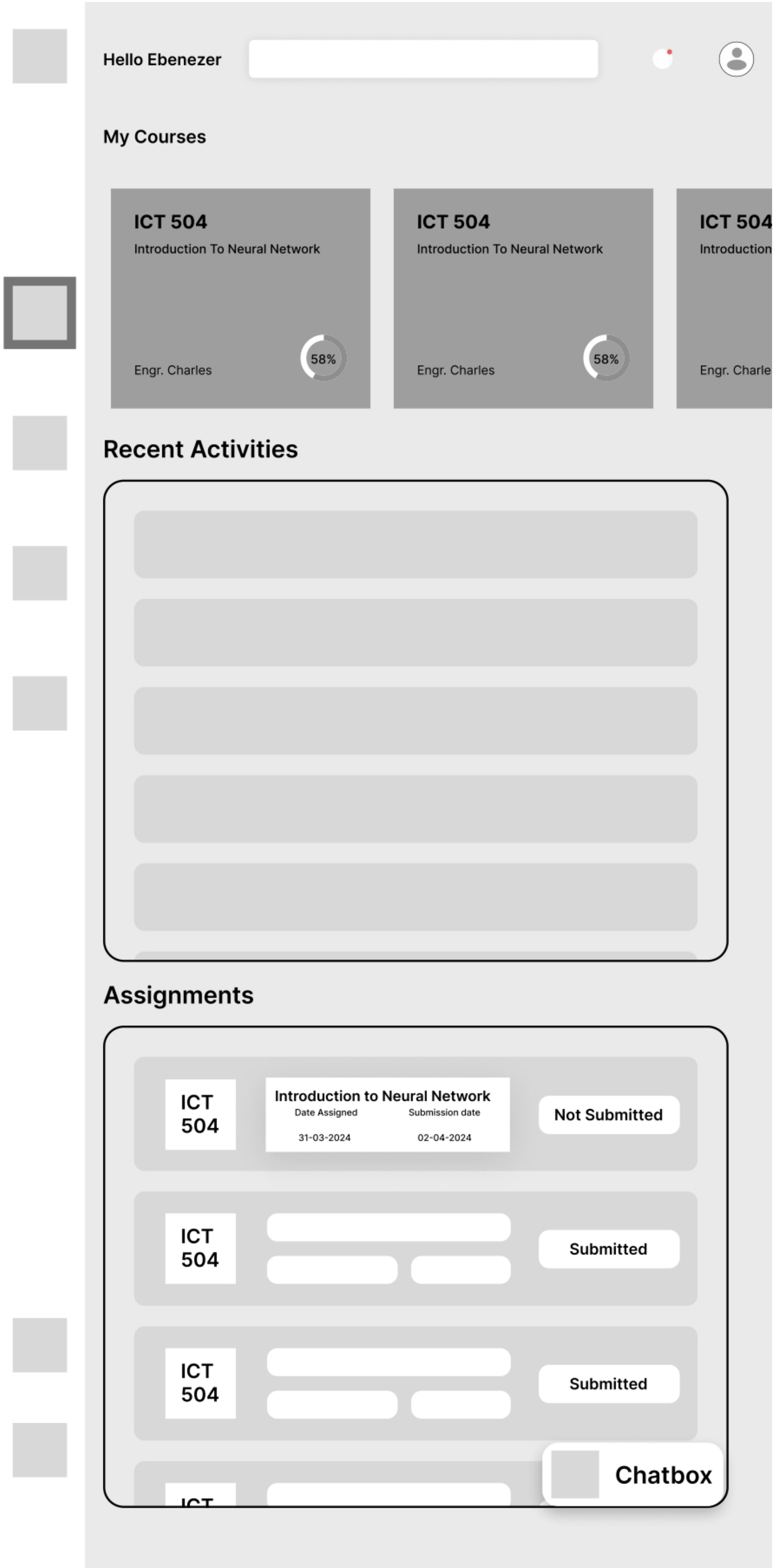


Figure 9. Wireframe for Student's Dashboard.

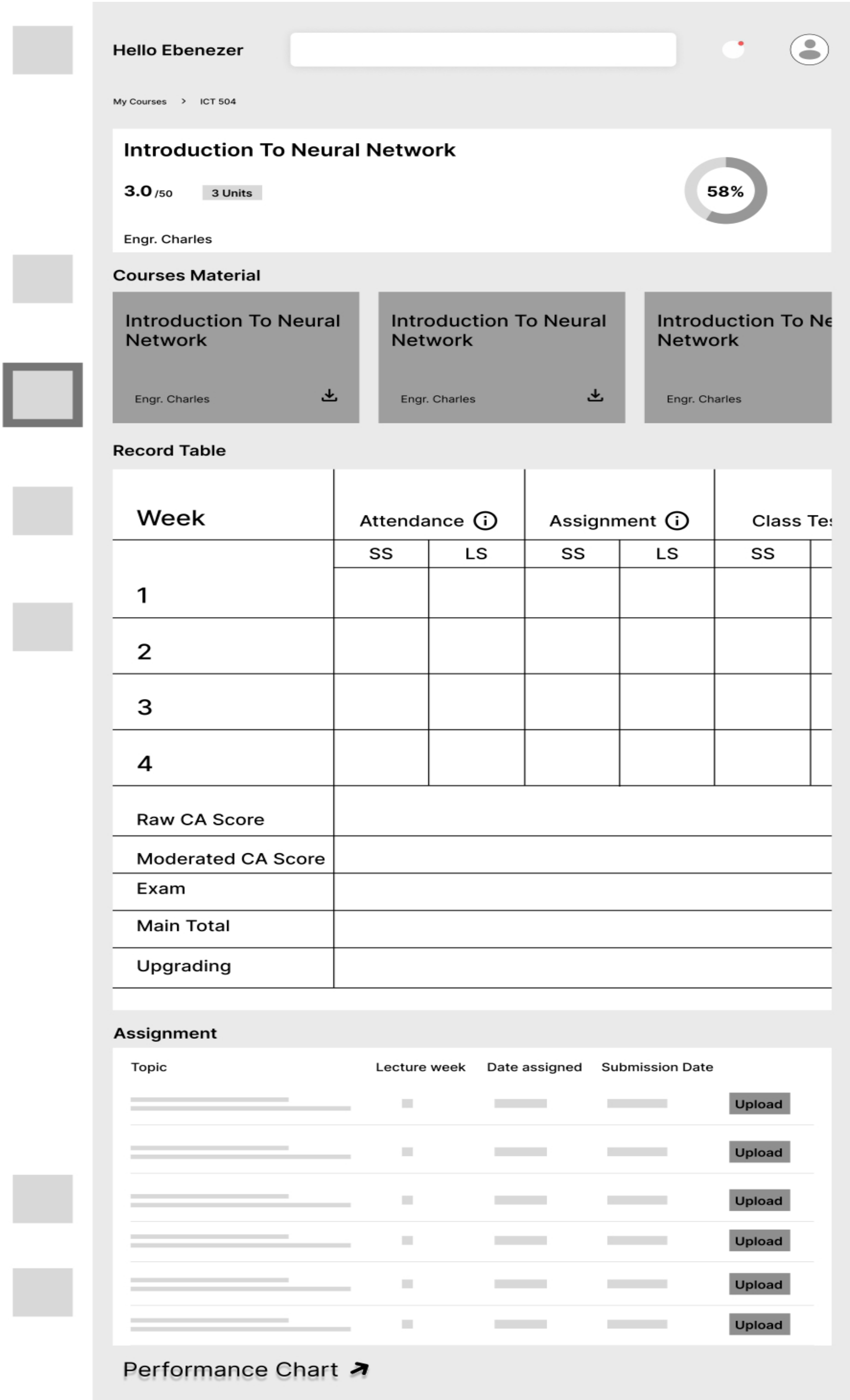


Figure 10. Wireframe for Student's Detailed Course View.

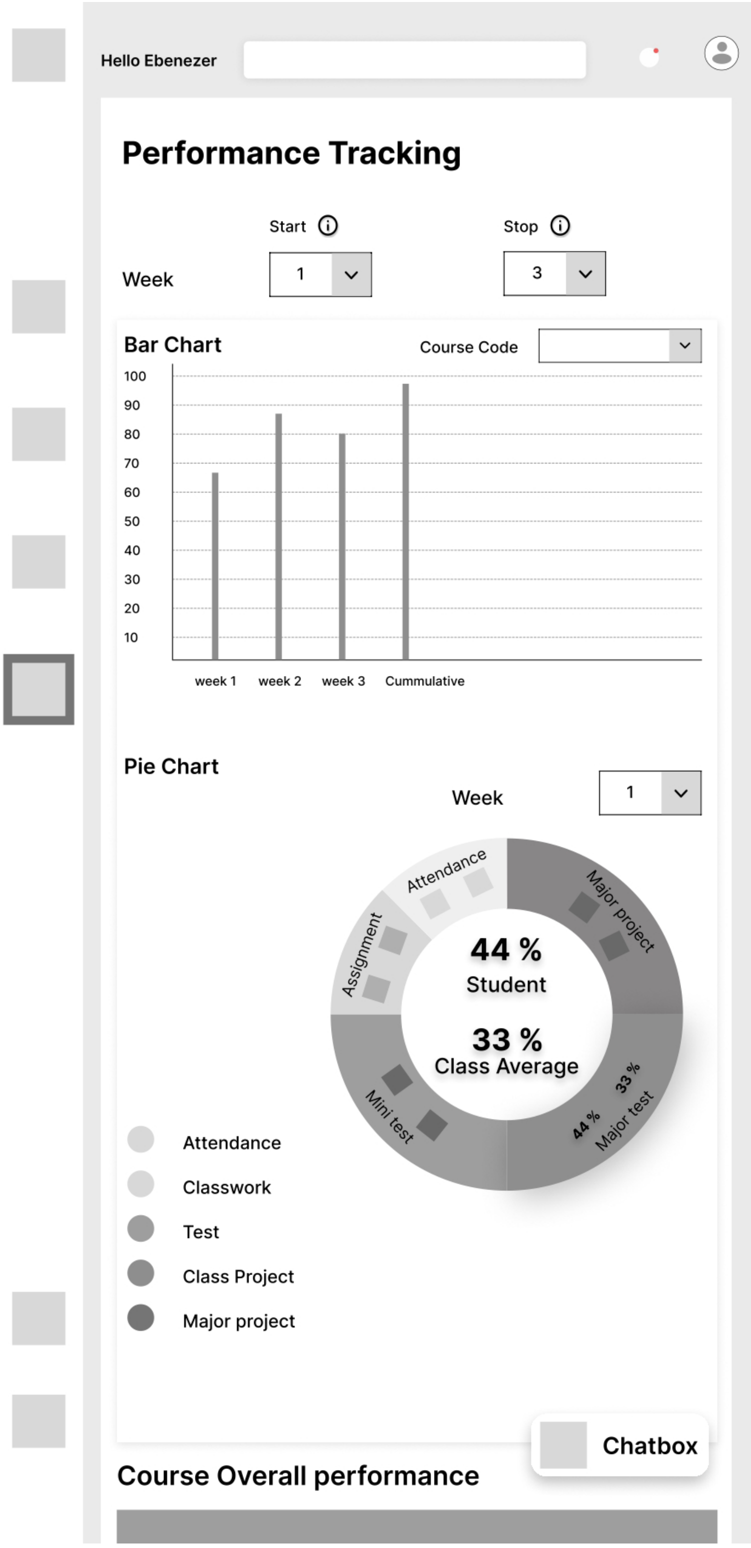


Figure 11. Student's Wireframe for Performance Tracking Page.

4.2. Selected Wireframes Design for the Lecturer's Interface

Figures 12 to **16** show some of the wireframes designed for the lecturer's interface.

The wireframes for both the student and lecturer interfaces were meticulously designed to serve as blueprints for the development of the Track-CA platform's frontend. These wireframes clearly delineated user roles and responsibilities, capturing key functionalities such as course upload, continuous assessment tracking, assignment grading, and real-time performance analytics. Although comprehensive in structure, the overall number of interface pages was substantial due to the deliberate inclusion of granular features aimed at covering every critical user interaction in the system. To preserve clarity and adhere to space constraints within this paper, only the most essential wireframes—those representing core flows and high-frequency actions—are presented in **Figures 7**–**16**. While the wireframes successfully represented the interface architecture, the actual implementation of these screens did not yet benefit from the Breadth-First Search (BFS) algorithm. As a result, the navigation across the app currently involves deeper click paths and higher user effort in reaching nested functions. This limitation highlights the need for subsequent BFS optimization in the interface development phase to reduce click depth, enhance usability, and align the live interface more closely with the system's low-friction design objectives.

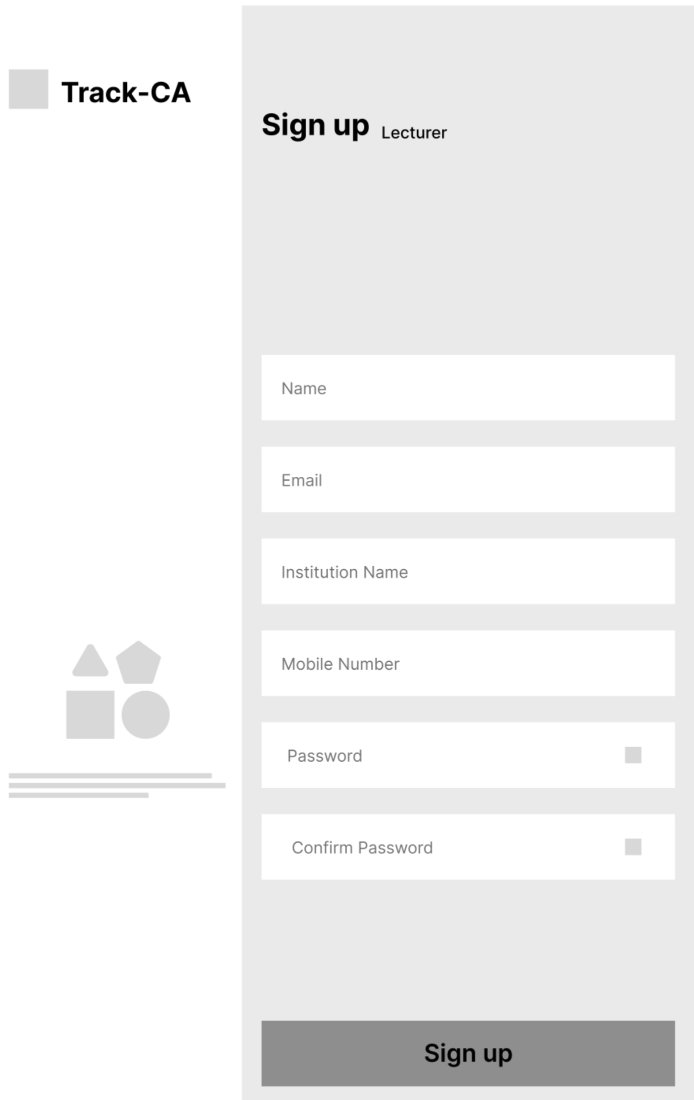


Figure 12. Wireframe for Lecturer Sign-Up Page.

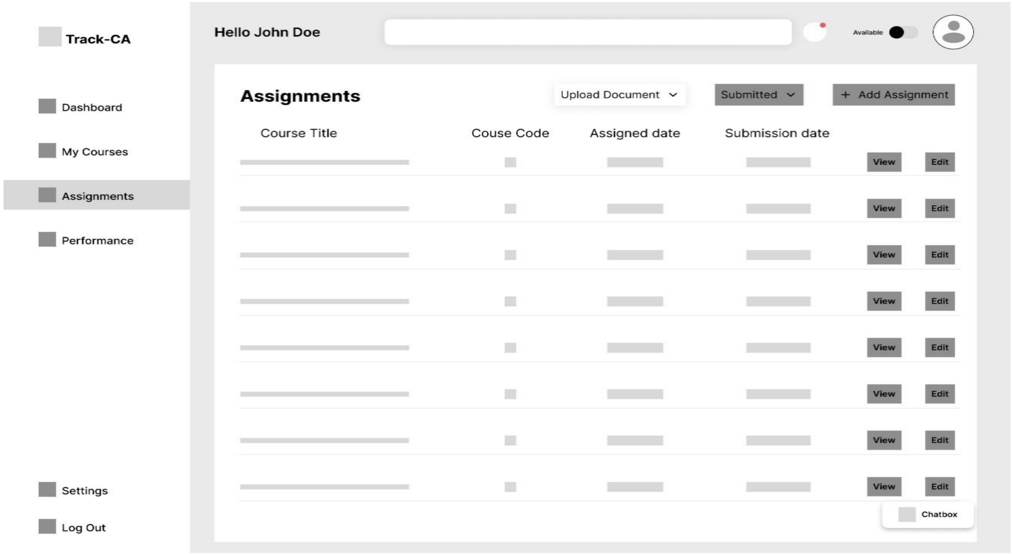


Figure 13. Wireframe for Assignment Page.

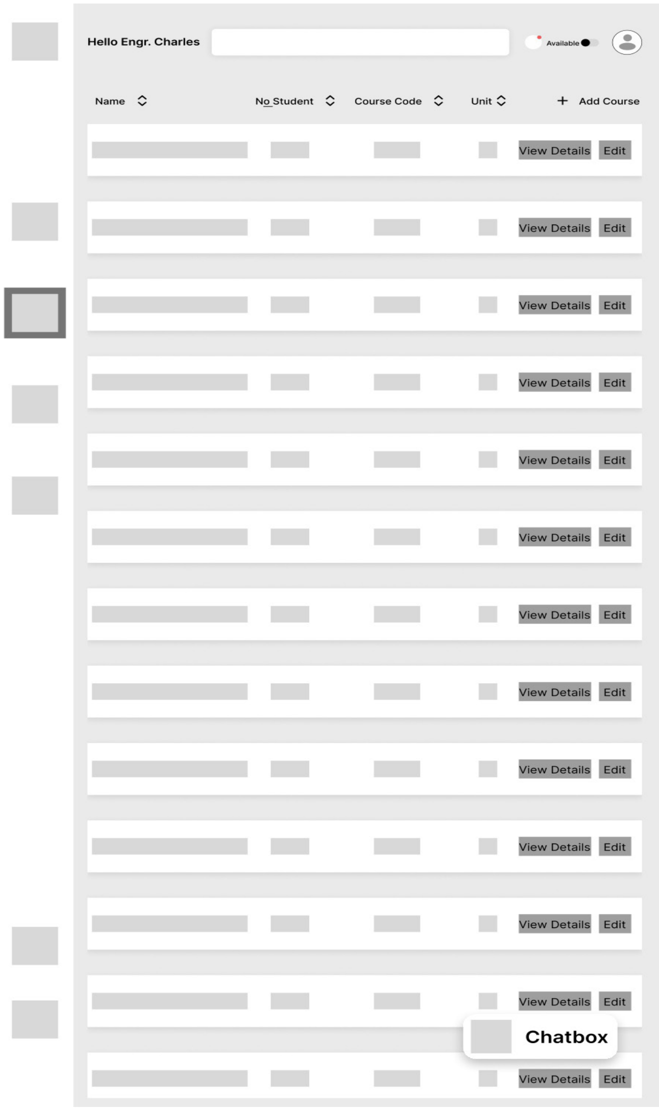


Figure 14. Lecturer's Wireframe for My Students Page.

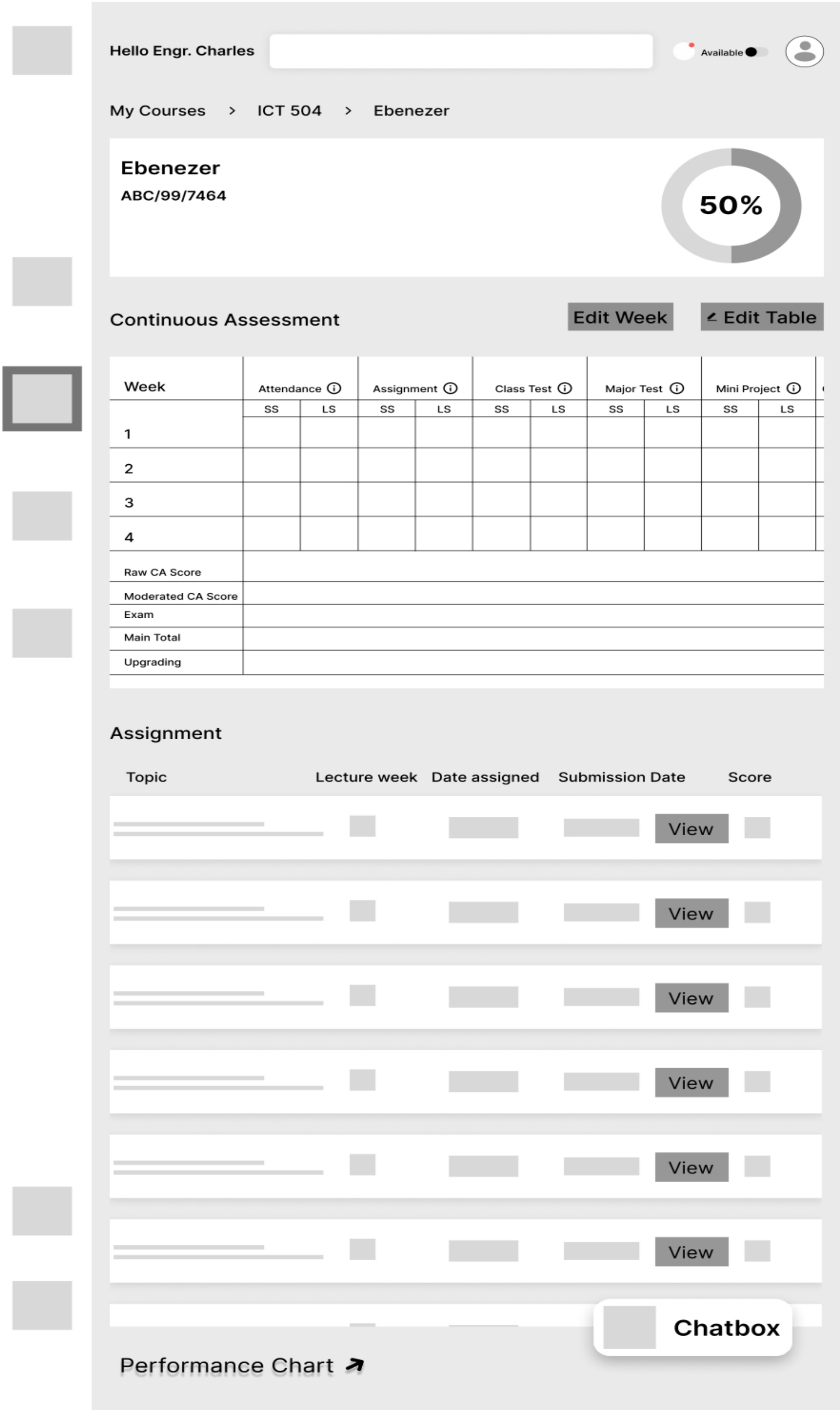


Figure 15. Lecturer's Wireframe for Tracking Student Progress.

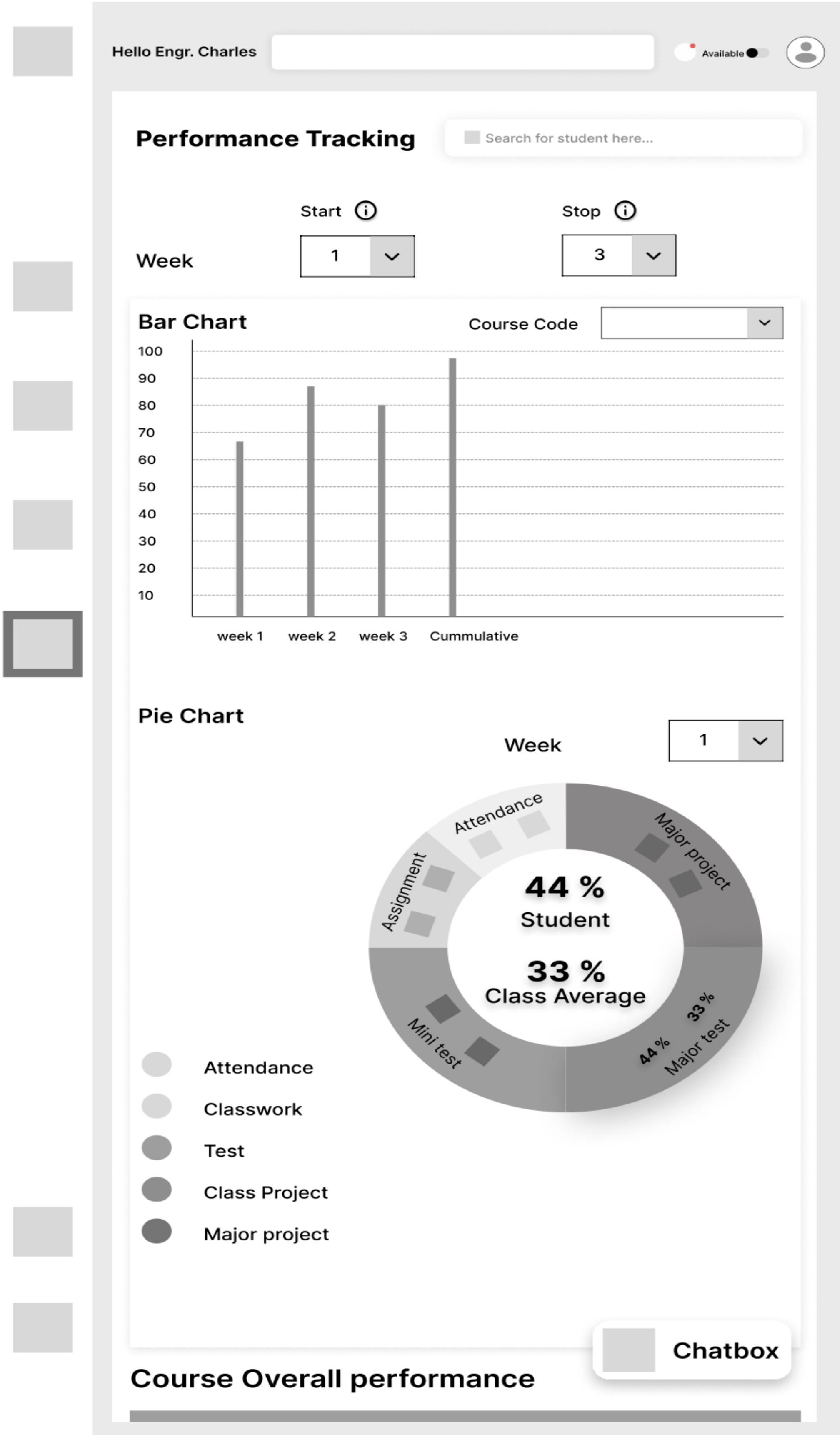


Figure 16. Lecturer's Wireframe for Performance Tracking Dashboard.

4.3. BFS-Based Optimization of the Interface

In response to the limitations identified in the initial wireframe structure in section 4.2, particularly the deep navigation layers and high click-up depth, the *Track-CA* platform was systematically redesigned to incorporate **BFS** algorithm directly into the frontend interface. This improved version was implemented and deployed on the **Vercel** platform. While retaining the comprehensive structure that supports a wide range of academic tasks, the redesigned interface now benefits from BFS-optimized navigation that ensures smoother transitions and faster access to key features.

The integration of BFS has significantly **reduced the number of clicks** required to access nested functions and **collapsed multiple related pages** into single, unified views. This approach not only enhances usability but also aligns with best practices in user-centered design, enabling users to accomplish tasks with greater efficiency. For instance, **Figures 17** to **20** illustrate the **BFS-enhanced lecturer interface** deployed on Vercel. **Figure 21** shows how these figures are presented together in a single composite layout to reflect the live user experience, where functionalities such as assignment management, weekly schedules, and notifications/announcements and course management are seamlessly accessed without redundant page transitions. This structural refinement marks a critical step in transforming Track-CA from a static wireframe prototype into a **dynamic, performance-oriented academic tool**, significantly improving interaction flow, system responsiveness, and overall user satisfaction.

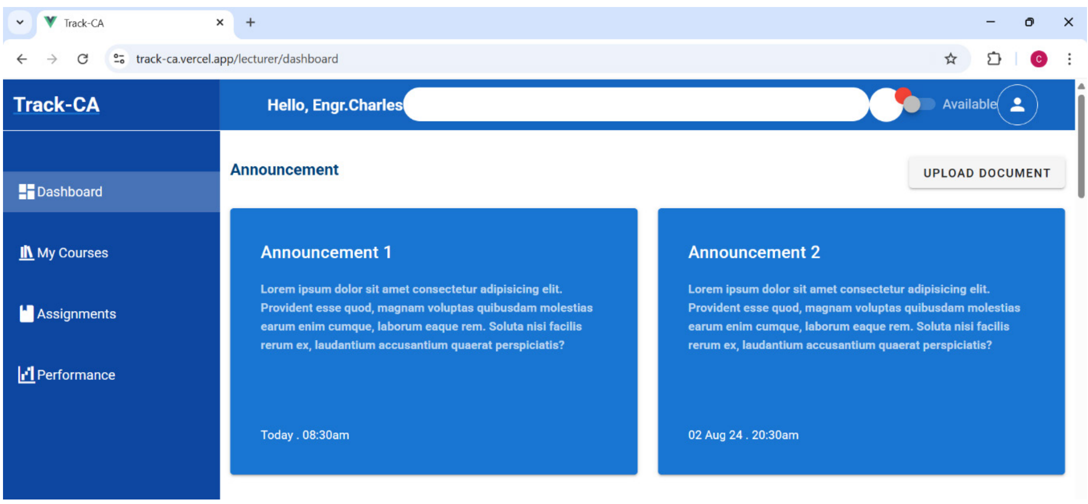


Figure 17. First View of the Lecturer's BFS-Based Frontend for Announcement Pane in Vercel Platform.

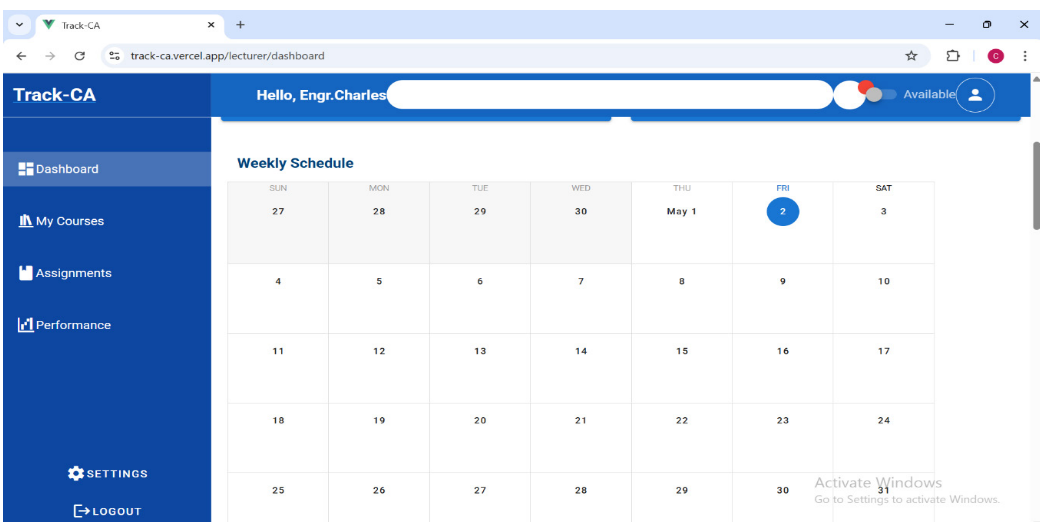


Figure 18. Second View of the Lecturer's BFS-based Frontend for Weekly Schedule Pane in Vercel Platform.

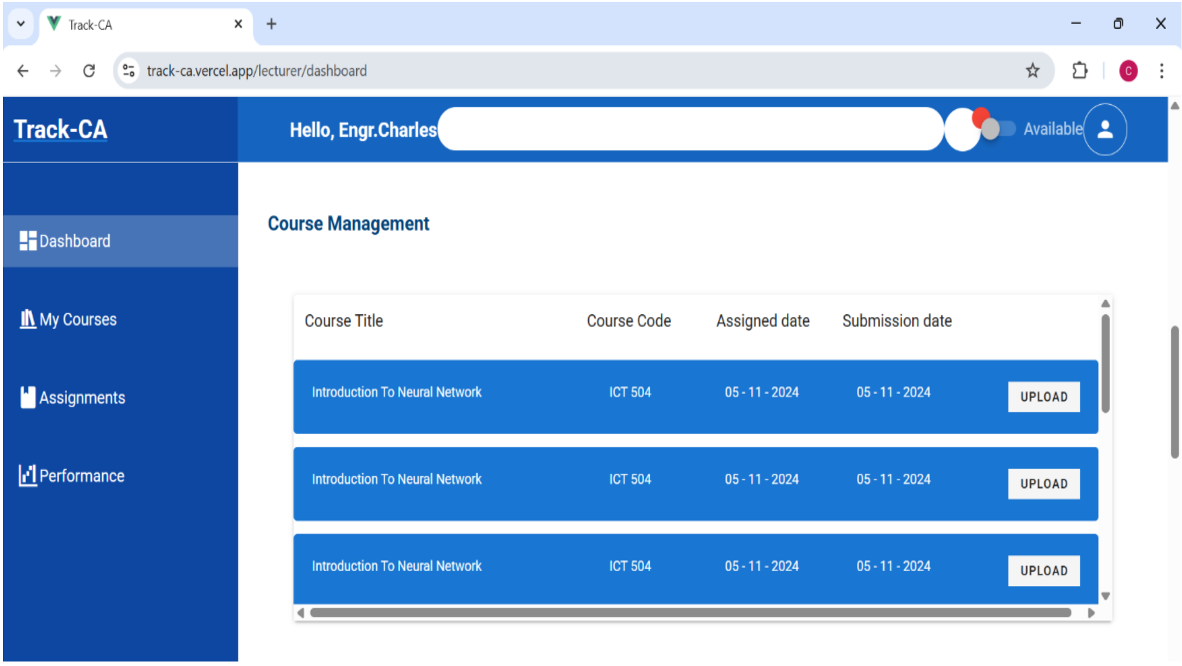


Figure 19. Third View of the Lecturer's BFS-based Frontend for Course Management Pane in Vercel Platform.

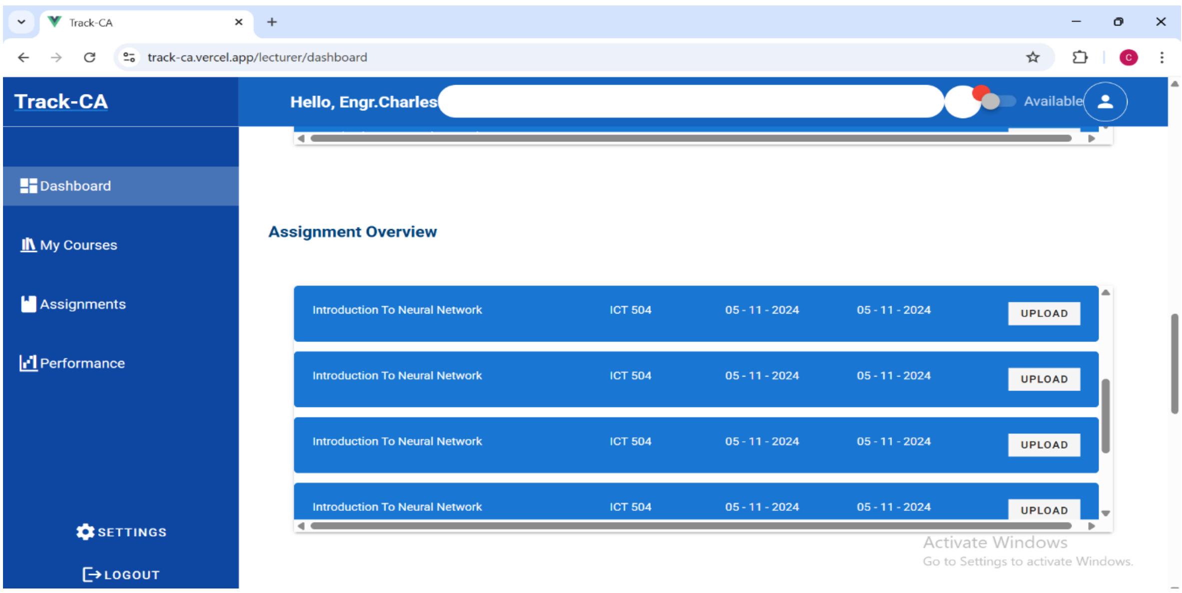


Figure 20. Fourth View of the Lecturer's BFS-based Frontend for Assignment Overview Pane in Vercel Platform.

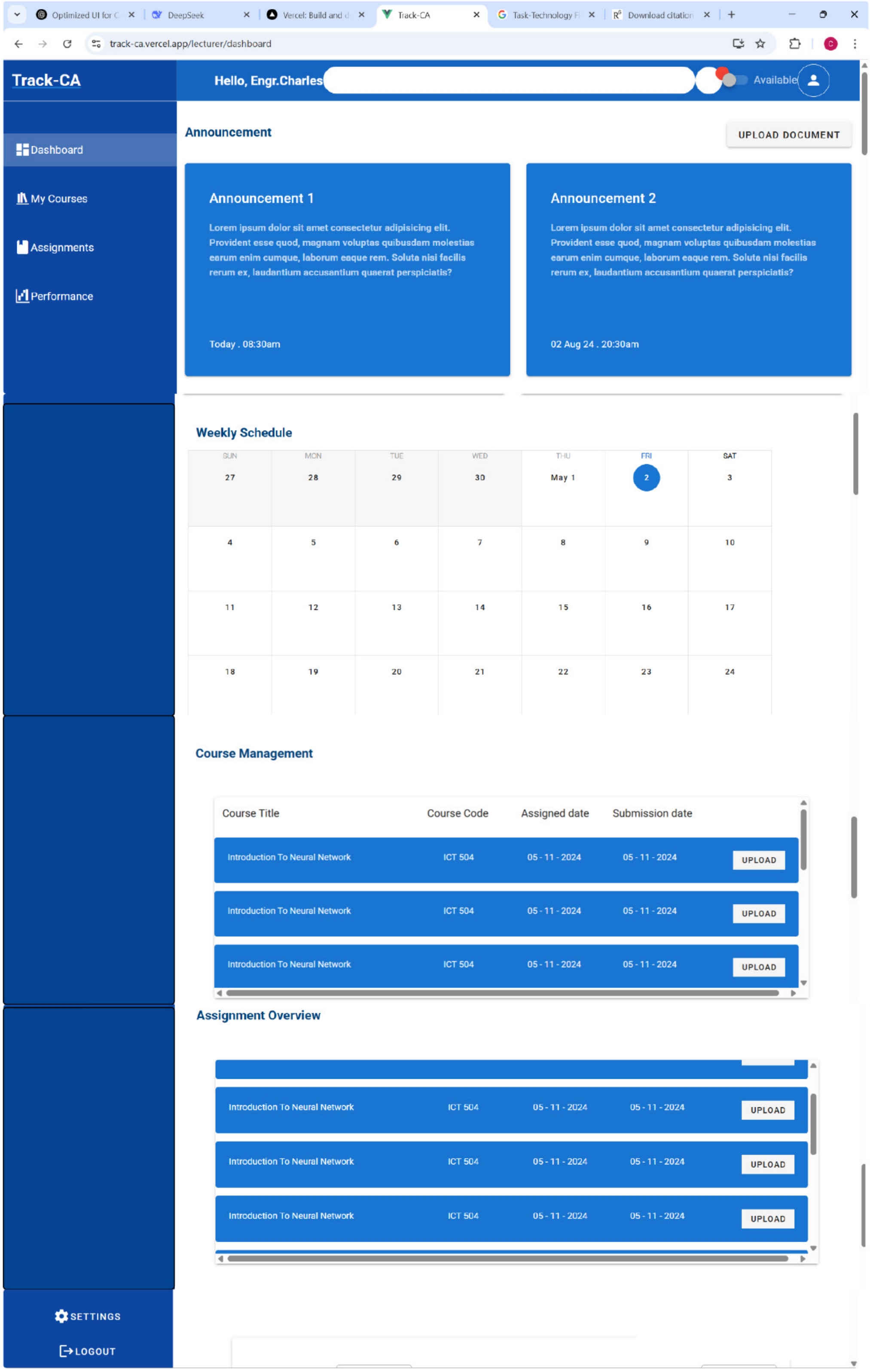


Figure 21. Single Composite Layout of the BFS-Enhanced Optimization of the Lecturer's Interface Shown in **Figures 17–20**.

Table 3. Comparison of TrackCA Before and After Optimization.

Metric	Before Optimization	After BFS + Vercel Deployment	Percentage Improvement
Average Page Transition Time	1.8 seconds	0.9 seconds	50% faster
Average Click Depth per Task	5 steps	3 steps	40% reduction
Task Completion Rate (First Attempt)	72%	93%	+21%
Interface Load Time (First Load)	2.5 seconds	1.3 seconds	48% faster
User Reported Satisfaction (1–5 scale)	3.6	4.4	+22%

By integrating the Breadth-First Search (BFS) algorithm into interface structuring and deploying the Track-CA platform on Vercel, several tangible improvements in user experience (UX) were recorded during pilot testing with 15 students and 10 lecturers as presented in **Table 3**.

The integration of BFS-guided routing and deployment on Vercel's edge network significantly enhanced system responsiveness. Page transitions became noticeably faster, with most occurring in under one second, effectively reducing lag and creating a smoother user experience across all device types.

Additionally, the restructuring of the interface into a shallower navigation tree helped minimize user confusion. On average, users required two fewer clicks to access frequently used features such as assignment submission and course performance tracking, streamlining interaction and reducing cognitive load as shown in **Figure 21**.

Usability testing also revealed a notable improvement in task execution. Ninety-three percent (93%) of users successfully completed their intended tasks on the first attempt, marking a 21% increase in task success rate compared to initial trials. This improvement shows the effectiveness of the optimized navigation structure and interface responsiveness in supporting intuitive user behavior.

5. Conclusions

This research project introduced *Track-CA*, a performance-oriented academic platform designed to enhance course accountability, student monitoring, and timely feedback for both students and lecturers. The system's frontend interfaces were carefully structured using the Breadth-First Search (BFS) algorithm, optimizing navigation flow and improving task accessibility through level-wise content traversal. Deployment via Vercel ensured fast rendering, global accessibility, and real-time updates, which significantly enhanced usability during testing.

Furthermore, the design was grounded in two robust theoretical frameworks: Self-Determination Theory (SDT), which informed the integration of real-time feedback to support student motivation, and the Task-Technology Fit (TTF) model, which shaped interface features to align closely with user goals and academic responsibilities. Together, these frameworks validated the platform's user experience and educational relevance.

Empirical evaluation revealed key user experience (UX) improvements, including reduced navigation complexity, faster access to core features, and higher user success rates in performing academic tasks such as submitting assignments and tracking progress.

However, a key limitation of this work lies in its scope—this version of Track-CA is focused solely on the frontend interface. As such, backend functionalities such as user authentication, persistent data storage, automated grading algorithms, and system-wide analytics were not implemented in this phase. Another limitation was that few users were used for testing (15 students and 10 lecturers).

For future development, we recommend extending this project by:

(i). Building a robust backend system using frameworks such as Node.js or Django.

(ii). Integrating a secure and scalable database to store user records, submissions, and performance data.

(iii). Incorporating intelligent algorithms for automated assessments, feedback personalization, and learning analytics.

(iv). Ensuring compliance with educational data privacy standards.

(v). Increase the number of users for testing the improved design.

These enhancements will transform *Track-CA* from a functional prototype into a comprehensive academic performance platform, capable of full deployment in higher institutions.

Author Contributions

Conceptualization, C.U and E.A.; methodology, C.U.; software, C.U and E.A.; validation, C.U and E.A.; writing—original draft preparation, E.A.; writing—review and editing, C.U.; supervision, C.U.; project administration, C.U and E.A. All authors have read and agreed to the published version of the manuscript.

Funding

This work received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

For our wireframes designs, you can find them through this link: https://drive.google.com/drivr/folders/1Qikq6_3ZHkg1ug5uV5YuaBNI2vRz5-Sh.

Acknowledgments

We express our sincere gratitude to all individuals who participated in testing the Track-CA platform during its development phase. We also appreciate those who provided support and guidance in using the design and deployment tools, especially during the implementation and evaluation stages of this work. Your contributions were invaluable to the success of this project.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Hooda, M., Rana, C., Dahiya, O., et al., 2022. Artificial Intelligence for Assessment and Feedback to Enhance Student Success in Higher Education. Mathematical Problems in Engineering. 2022(1), 5215722. DOI: https://doi.org/10.1155/2022/5215722

[2] Dawson, P., Henderson, M., Mahoney, P., et al., 2019. What makes for effective feedback: staff and student perspectives. Assessment and Evaluation in Higher Education. 44(1), 25–36. DOI: https://doi.org/10.1080/02602938.2018.1467877

[3] Faulconer, E., Griffith, J., Gruss, A., 2021. The impact of positive feedback on student outcomes and perceptions. Assessment and Evaluation in Higher Education. 47(2), 259–268. DOI: https://doi.org/10.1080/02602938.2021.1910140

[4] Samiullah, I.M., Anjum, A., 2017. Effect of continuous assessment techniques on students' performance at elementary level. Bulletin of Education and Research. 39(1), 91–100.

[5] Uwaifo, V., Uddin, P., 2009. Transition from the 6-3-3-4 to the 9-3-4 System of Education in Nigeria: An Assessment of Its Implementation on Technology Subjects. Studies on Home and Community Science. 3(2), 81–86.

[6] Juanah, J.E., 2018. The State of Continuous Assessment Practices in Junior Secondary Schools in Kenema City. Global Journal of Human-Social Science. 18(G8), 11–17. Available from: https://socialscienceresearch.org/index.php/GJHSS/article/view/2651 (cited 01 January 2025).

[7] Bhattacharjee, B., Deb, K., 2016. Role of ICT in 21st century's teacher Education. International Journal of Education and Information Studies. 6(1), 1–6.

[8] Banjoko, S.O., Akindoju, O.G., Jimoh, A.S., 2010. Perceived roles of information and communication technologies in the implementation of continuous assessment in Nigerian secondary schools. African Journal of Teacher Education. 1(1), 78–90. DOI: https://doi.org/10.21083/ajote.v1i1.1577

[9] Yussuff, A.I., Goke, A., Folorunsho, H.B., et al., 2024. Development of Integrated Web-Based Continuous Assessment Management System. UNIOSUN Journal of Engineering and Environmental Sciences. 6(2), 138–147. DOI: https://doi.org/10.36108/ujees/4202.60.0221.

[10] Murugan, V., 2021. ICT Based Assessment Methods in Education. Journal of Emerging Technologies and Innovative Research. 8(7), 4.

[11] Chekwube, E., Mgbeafulike, E., 2021. An Integrated System for Continuous Assessment and Examination Management in Schools and Colleges. International Journal of Computer Applications Technology and Research. 10(4), 82–86.

[12] Alqurni, Jehad. 2023. Evaluating the User Interface and Usability Approaches for E-Learning Systems. International Journal of Information Technology and

Web Engineering. 18(1), pp.1-25. DOI: https://doi.org/10.4018/IJITWE.333638.

[13] Ibarra, S.M., Olmos, S.C., Ibáñez, J., 2017. Trends and challenges of educational assessment and evaluation in the digital society. Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality; 18–20 October 2017; Cádiz, Spain. pp. 1–4. DOI: https://doi.org/10.1145/3144826.3145364

[14] Peter, P., Matus, T., 2020. The usage of vue js framework for web application creation. Mesterséges intelligencia. 2(2), 61–72.

[15] Junhui, S., Min, Z., Hua, X.X., 2019. Design and Implementation of a Vue.js-Based College Teaching System. International Journal of Emerging Technologies in Learning. 14(13), 59.

[16] Mustaqim, A., Dinova, D.B., Fadhilah, M.S., et al., 2024. Optimizing the implementation of the BFS and DFS algorithms using the web crawler method on the kumparan site. Journal of Soft Computing Exploration. 5(2), 200–206. DOI: https://doi.org/10.52465/joscex.v5i2.309

[17] Syaichul, F.A., Tito, T.P., Rita, P., et al., 2018. Implementation of Breadth First Search Algorithm for Verification in the Informatics Engineering Education (PTI) UM. Proceedings of the 2nd International Conference on Vocational Education and Training (ICOVET 2018); 26–28 October 2018; Malang, Indonesia. pp. 229–233.

[18] Somayeh, F., Takeshi, O., 2018. Technology acceptance model in technology-enhanced OCLL contexts: A self-determination theory approach. Australasian Journal of Educational Technology. 34(4), 138–154. DOI: https://doi.org/10.14742/ajet.3629

[19] Saida, U., Ence, S., Izzul, F., et al., 2024. Task-Technology Fit Analysis: Measuring the Factors that influence Behavioural Intention to Use the Online Summary-with Automated Feedback in a MOOCs Platform. The Electronic Journal of e-Learning. 22(1), 63–77. DOI: https://doi.org/10.34190/ejel.22.1.3094

[20] Vue.js, 2025. The Progressive JavaScript Framework. Available from: https://www.vuejs.org (cited 01 January 2025).

[21] Vuetify, 2025. Vue Component Framework. Available from: https://vuetifyjs.com/en/ (cited 01 January 2025).

[22] Vercel, 2025. Your complete platform for the web. Available from: https://vercel.com/ (cited 01 January 2025).

[23] Figma, 2025. Track-CA Figma designs. Available from: https://www.figma.com/design/ (cited 01 January 2025).

[24] Holdsworth, J., 1999. The Nature of Breadth-First Search. School of Computer Science, Mathematics, and Physics James Cook University, Australia. Technical Report 99-1, 18 January 1999. Available from: http://cairns.cs.jcu.edu.au/~jason https://www.researchgate.net/publication/2727226_The_Nature_of_Breadth-First_Search. (cited 01 January 2025).

[25] Legault, L., 2017. Self-Determination Theory. In: Zeigler-Hill, V., Shackelford, T. (eds.). Encyclopedia of Personality and Individual Differences. Springer, Cham, Switzerland. pp. 1–9. DOI: https://doi.org/10.1007/978-3-319-28099-8_1162-1

[26] D'Ambra, J., Concepcion, W., Shahriar, A., 2013. Application of the Task-Technology Fit Model to Structure and Evaluate the Adoption of E-Books by Academics. Journal of the American Society for Information Science and Technology. 64(1), 48–64. DOI: https://doi.org/10.1002/asi.22757