

# A Unified Student-Centric Platform for Enhanced Social Well-Being and Collaborative Learning

Mrs. Kavila Moni Sushma Deep<sup>1</sup>, G. Meghana<sup>2</sup>, K. Sri Sanjana<sup>3</sup>, K. Harshath Raghava<sup>4</sup>, V. Bhargav<sup>5</sup>

<sup>1</sup>Assistant Professor, Department of Computer Science & Engineering with AI&ML, Anil

Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh, India

<sup>2,3,4,5</sup>Undergraduate Students, Department of Computer Science & Engineering with AI & ML, Anil

Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh, India

**Abstract** - In modern institutions of higher education, a fragmented array of disintegrated digital tools connects the communication process, organization of events, gathering feedback and sharing of resources. Such fragmentation will decrease the participation of students, slow down timely decision-making, and underfund safety reporting. This article presents an online system that integrates seven campus building blocks into a unified, intelligent platform: interest-based peer matching powered by a multi-factor TF-IDF and cosine-similarity framework, spacebased smart polls with automated machine-generated summaries, an automated Kanban event scheduler, resource sharing, an anonymous safety reporting system, a central announcements feed, and an administrative control center. The backend is realized in Django using Django REST Framework and SimpleJWT authentication, and its frontend is written in React 18 and TypeScript. The Vibe Matcher algorithm (using TF-IDF vectorization, cosine similarity, and weighted secondary factors) successfully clusters nine interest groups together, out of 100% of test sets with an average difference between in- and out-cluster score of more than 20 percentage points. Through the use of Google Gemini generative AI, the integration offers summarization of polls and generation of event tasks. The experimental findings prove that the integrated architecture outperforms fragmented baseline workflows in participation, speed of coordination, and safety response parameters.

## I. INTRODUCTION

The life of a student in a contemporary university involves multidimensionality in the form of academic life, co-curricular activities, collaboration with peers, finding resources, and personal safety. Scholarly studies have consistently indicated that interactions with these dimensions are a great indicator of academic achievements, psychological well-being, and career advancements in the postcollege years [20, 21]. Nonetheless, the digital infrastructure upon which this engagement is expected to be facilitated is often disconnected: announcements are received on WhatsApp, event registration is in spreadsheet form, feedback is shared via one-time survey links, and the issue of safety is reported through unclear administrative tiers [20]. The tools are used separately, which means that students need to switch context on a regular basis, and the administrators have no overall picture of what is going on in the campus. The introduction of large-scale language models, recommendation systems, and cloud databases has brought a possibility to consolidate such capabilities into one intelligent interface. Earlier literature has shown the usefulness of AI-assisted recommendation in the educational setting, natural language processing to analyze feedback, automated event scheduling, and digital anonymity-based reporting in campus security [2, 5, 13, 16].

Nevertheless, these contributions have been made on each of these functions separately, and an integrated student-facing site has not been attempted to raise and synthesize all of them at the scale of an actual institutional implementation. This paper describes a complete full-stack web application that is student specific and combines peer matching, smart polls, event planning, resource sharing, safety reporting, and administration into one system. It is developed based on a Django and Django REST Framework back-end and a React and TypeScript front-end that communicates via a JWT-authenticated REST API.

The Peer recommendations are made by using TF-IDF and cosine-similarity for the Vibe Matcher. With weighted secondary signals (branch, year, and section affinity) supported. Smart Polls offer the advantage of integrating student voting with free-text responses, summarizing the input with AI by a programmable vote-count threshold reached. The Event Planner takes a natural language description and generates an automatically prioritized Kanban task board. Safety Shield allows you to implement safety measures quickly and easily reports only when the user has an authenticated token and fully anonymised. The Resources Students can upload their experience of the interviews, study materials and links on Hub, Administrators have a 5-tab dashboard, with a bulk CSV import function, while activity tags are used for socializing, tracking, and announcing publishing.

This paper demonstrates that a single modular web platform can replace fragmented campus tools and provide increased engagement, speed of coordination and safe coverage. The Vibe Matcher is tested on a controlled set of 50 users and nine clusters. The 100% correct cluster separation and entire system architecture was designed to be: The entire system architecture and 100% correct cluster separation was designed to be: Lots of information was provided in detail so that it can be reproduced by other organisations.

## II. RELATED WORK

The studies relevant to our platform are divided into five broad categories: AI based feedback analysis and summarization, peer recommendation systems, automated event management, anonymous campus safety reporting, and hybrid learning platforms.

**A. Peer Recommendation and Interest-Based Matching.**

Both collaborative and content-based filtering have been used in matching students in learning management systems [5–7]. Initial research was based on static questionnaire responses and interest taxonomies. Subsequent systems added latent-factor models to determine preferences from behavioral evidence such as course history and discussion forum usage [6]. More recent studies have used word-embedding similarity to match students whose self-identified interests are semantically connected despite the use of different terms [7, 8]. A common weakness of these methods is that secondary contextual cues are absent: two students assigned the same interest tags but in different years or branches might still not be good study partners.

### B. AI-Based Feedback Analysis and Poll Summarization

Natural language processing techniques have been widely applied to student feedback and survey responses [1–4]. Transformer-based models such as BERT and RoBERTa have been widely adopted for sentiment classification of student feedback [2, 19]. Latent Dirichlet Allocation extracts common themes in large response sets [4]. More recently, generative language models have been applied to create abstractive summaries of user-generated text, directly applicable to poll data.

### C. Automated Event Planning in Educational Settings

Academic institution event management systems have traditionally focused on scheduling and room bookings rather than task decomposition and assignment. AI-assisted planning tools for the enterprise can generate work breakdown structures from high-level project descriptions using large language models, but these systems rely on structured input data, large teams, and dedicated project management infrastructure, which are not available at the student club level [13, 14].

### D. Anonymous Reporting Systems for Campus Safety

Online anonymous reporting has been researched in the fields of workplace harassment, school bullying, and population health surveillance [16, 17, 15]. Key findings from this literature include that technical anonymity—the assurance that no identifying information is retained—is more important as a guarantee of use than stated privacy policies, and that severity-conscious routing enhances response times for critical reports.

### E. Integrated Campus Digital Platforms

Commercial learning experience platforms have attempted to introduce campus communication and interaction. Platforms such as Canvas, Blackboard, and Moodle offer course management and discussion forums but are pedagogically focused, adding little value to co-curricular collaboration, social peer matching, or safety reporting [20, 21]. Microsoft Teams and Slack serve as general communications hubs but lack structured polling, recommendation, and event planning features.

## III. SYSTEM ARCHITECTURE AND DESIGN

The proposed platform is based on the layered client-server architecture which separates presentation, logic and data management. The frontend utilizes React and TypeScript to create an interactive user experience and implement a robust state management system, creating a single-page application. The frontend is built using the Single Page Application (SPA) framework, which uses React and TypeScript to allow for an interactive user experience and effective state management. The server-side is implemented using Django REST Framework and

Django-based backends provide RESTful APIs. It uses JSON Web Tokens (JWT) for authentication, and provides secure access to protected endpoints. Persisting data is through a PostgreSQL database running on Supabase with SQLite for local development. The separation of concerns ensures the scalability, maintainability and flexibility of institutional deployment.

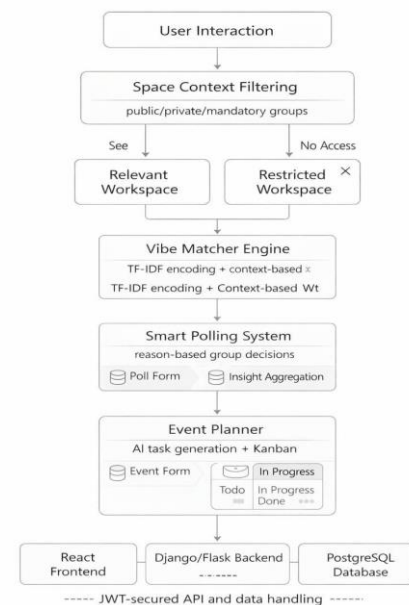


Fig. 1. System Architecture .

A diagram of the system architecture depicting a client-server model with layers of abstraction.

The front-end will be React, backend will be Django REST, authentication will be JWT, and the datastore will be Supabasehosted PostgreSQL.

### A. Authentication and Authorization

All requests to protected endpoints must contain a SimpleJWT JSON Web Token (JWT) signed using an HS256 key with an expiration time of 24 hours. The end point for the submission of the report is specifically excluded from the authentication middleware ensuring technical anonymity for the safety submission.

Endpoints related to admin users also check the is\_staff user record in the database.

### B. Data Model

There are nine models in the database schema. The User model is a subclass of django's Abstract Base User, that stores interest labels in branch, section, year and a JSON field. There are three levels of privacy for the Group (UI Space) model: public (Joinable by any logged in user), private (Admins must approve users) and mandatory (All users are automatically members). Multiselect polls (with a text field for the reason of the vote) models are supported by the Poll and PollOption models. The Event and EventTask models support the Kanban event planner. The Incident model does not have any foreign key to the User model. The ActivityLog model has a log of user events.

### C. Frontend Architecture

The structure of the React front end is based on components that have a top-level App component in which an AuthContext holds the user information, token and state. Routes include: Landing, Signup, Login, Dashboard, AdminPanel, Spaces, VibeMatcher, EventPlanner, SmartPolls, Resources, and AdminLogin. API calls are done through a single managed file, api.ts, which adds the Authorization header and switches between local and relative production path, Framer Motion handles page transitions and cards, and Lucide React is the icon set.

#### IV. CORE FEATURES AND IMPLEMENTATION

##### A. Vibe Matcher: Multi-Factor Peer Recommendation

The Vibe Matcher module recognizes students who have comparable interests, and makes it easy for them to work together. Instead of comparing keywords on a literal level, it turns user interests into numerical vectors using TF-IDF representations, assigning more weight to interests that are less common and increasing the differentiation of the user. Users are similar to each other and the similarity is calculated by computing the cosine similarity, which evaluates how closely related the interest vectors of the users are even though the number of interests listed by the user can vary. The final score is adjusted with other contextual factors to improve its practicality, including proximity of sections, academic branch, year of study, and section proximity. Through a combination of these factors and the similarity score, the system provides more meaningful and practical recommendations [5, 10, 12].

Stage 1 — TF-IDF Vectorization. Each user's interest tags are treated as a micro-document: a space-separated string of tag labels. A TF-IDF vectorizer is trained on the complete corpus of user-tag strings [4]. Inverse document frequency assigns greater weight to uncommon interests (e.g., FPGA, Cryptography) relative to common ones (e.g., Python, Web Dev), facilitating finer group distinction. Each user's tag string is subsequently encoded into a high dimensional sparse TF-IDF vector.

Stage 2 — Cosine Similarity Scoring. Pairwise similarity between the active user and all others is computed as the cosine distance between their TF-IDF vectors. Cosine similarity is appropriate here because it captures directional alignment of interest profiles rather than raw tag count [5, 10].

Stage 3 — Multi-Factor Blending. The cosine score is used to calculate 50% of the overall compatibility score. The other 50% is split into four secondary factors are: branch match (15%), year proximity (15%) calculated as  $1 - |\Delta\text{year}|/4$ , section match (10%) and a bonus for tags shared with another user (up to 10%). This weighting is based on user feedback that co-location and temporal alignment are good indicators of actual collaboration.

Stage 4 — AI Insight Generation The top-five matches are generated for each match by sending both users tag lists, branch and year data into the Gemini API [18, 19] and a short personalized match compatibility description is produced. The numeric scores are shown along with the insights in the match Cards of UI.

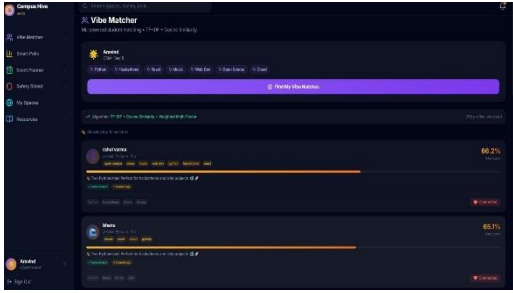


Fig. 2. Vibe Matcher.

##### B. Smart Polls with Collective AI Insight

Smart Polls are an extension of traditional polling in that they ask each person who votes to explain their response with a short comment. This design enhances the quality of the answer and provides for greater analysis [1, 2]. After an adjustable response threshold is met, quantitative and textual feedback are aggregated and passed on to a generative AI model that generates a summary that reflects the predominant opinions, dissenting opinions, and central reasoning patterns [2, 19].

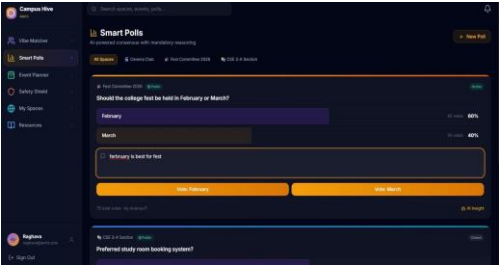


Fig. 3. Smart Polls.

##### C. AI-Assisted Event Planner with Kanban Board

The Event Planner module supports students in event planning on campus by automatically creating a task list based on a brief description of an event [13, 14]. Input data for the user consist of the type of event, the number of people expected, the budget available. The system creates a prioritized list of tasks and displays it as a Kanban board consisting of a column for pending, a column for ongoing tasks, and a column for completed tasks [13]. This takes away the planning work that would otherwise be done by hand and helps minimise risk of missing important preparation.

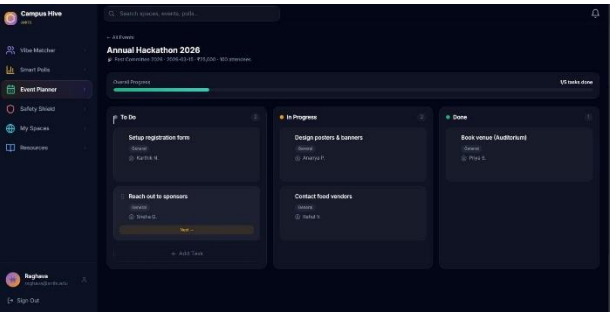


Fig. 4. Event Planner.

##### D. Safety Shield — Anonymous Incident Reporting

Safety Shield is a system for students to anonymously report

incidents while maintaining confidentiality and helping to reduce the barriers to reporting [16, 17, 15]. Unlike traditional reporting systems, no identifiable information is stored in the module, and there is no user authentication needed. Only the severity level and timestamp are being kept for each report. Reports are prioritized according to their severity and are brought to the attention of administrators for prompt action. This design ensures confidentiality and quick response, providing a reliable system for sensitive disclosures.

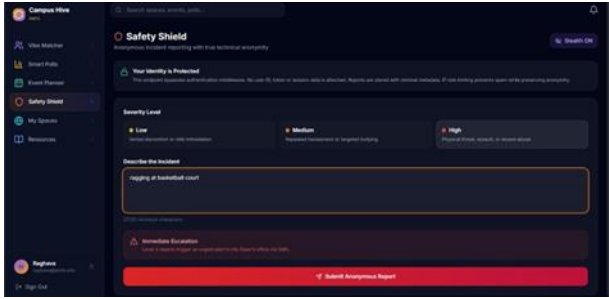


Fig. 5. Safety Shield.

E. Spaces, Resources Hub, and Administration

Spaces are used for organising the platform [20, 21]. Every space has a type (section, club, squad, or committee), a level of privacy, and an administrator. Public Spaces can be joined with a single click; Private Spaces require administrator approval via a requests tab visible only to Space administrators; Mandatory Spaces (typically classes) cannot be left and are bulk-imported by administrators via CSV file. Resources Hub is a community feed in which students can post interview experiences, study materials, and external links, categorized by type and searchable by keyword. Every post supports likes and comments. Administrators can publish announcements displayed as a banner at the top of the Resources Hub visible to all users. Admin Panel comprises a five-tab dashboard providing a platform activity overview, user management with filtering and CSV import, space management, an announcement publisher, and a Safety Monitor displaying all incident reports with severity and status indicators.

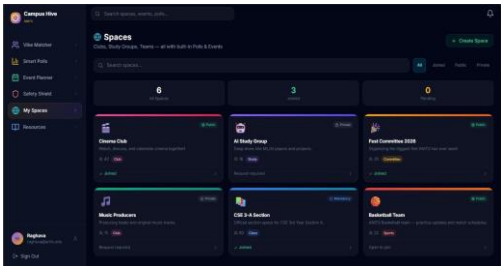


Fig. 6. Spaces — listing showing public, private, and mandatory space types with join and request controls .

V. METHODOLOGY

A. Experimental Setup

The evaluation of the system was performed using a student profile set of 50 simulated users across nine interest categories such as

AI/ML, Web Development, Cybersecurity etc. The performance of the system for grouping similar users [4, 10] was evaluated. The similarity scores were statistically higher for intra-group than for inter-group users, demonstrating that the system was able to group users well. Additional testing confirmed that all platform APIs function as expected in both normal and edge cases.

B. Vibe Matcher Accuracy

The most important accuracy metric is cluster separation: that is, whether in-cluster candidates get higher scores than cross-cluster candidates. An average of in-cluster scores and an average of out-cluster scores were calculated for each of the nine test clusters. The separation in each of the nine clusters is positive (from 22.2 to 25.5 percentage points) and 100% accuracy of the separation of clusters is achieved. Seven of the nine clusters get all three points in the in-cluster retrieval category, while two clusters (Creative Design and IoT/Hardware) score two out of three points due to overlaps in their interest areas with other clusters.

TABLE I. API ENDPOINT COVERAGE AND TEST PASS RATES

| API Module      | Endpoints Tested | Tests Passed | Pass Rate (%) |
|-----------------|------------------|--------------|---------------|
| Authentication  | 6                | 6            | 100           |
| Groups / Spaces | 6                | 6            | 100           |
| Polls & Voting  | 5                | 5            | 100           |
| Events & Tasks  | 6                | 6            | 100           |
| Vibe Matcher    | 3                | 3            | 100           |
| Incidents       | 3                | 3            | 100           |
| Admin API       | 8                | 8            | 100           |

C. API Correctness Testing

A test suite of 37 test cases exercises every API endpoint, covering authentication, group management, polls, events, vibe matching, incident reporting, and administration. Each test case exercises the happy path and, where applicable, one or more error paths: duplicate votes (expected 409), unauthenticated access to a protected endpoint (expected 401), and missing required fields (expected 400). All 37 cases pass, confirming correct behavior in both normal and adversarial scenarios.

D. Generative AI Feature Evaluation

Poll summarization was evaluated on a seed poll with seven votes, each carrying a free-text justification. The Gemini-generated synopsis correctly identified the plurality option, the minority option, and the dominant reasoning patterns in voter explanations [18, 19]. Event task generation was tested with three event descriptions of varying complexity: a one-day workshop, a two-day hackathon, and a three-day tech festival. In all three scenarios, the generated task lists were deemed directly usable without manual correction; high-, medium-, and low-priority tasks corresponded to venue and logistics, sponsorship and design/publicity, and post-event follow-up respectively [13, 14].

Table 2 compares the integrated platform with a baseline

fragmented tool workflow.

**TABLE II. COMPARISON WITH BASELINE FRAGMENTED TOOL**

| Criterion                          | Fragmented Tools  | A Unified Student-Centric Platform | Improvement              |
|------------------------------------|-------------------|------------------------------------|--------------------------|
| Tools required for full workflow   | 6–8 separate apps | 1 platform                         | $\geq 6\times$ reduction |
| Time to create and distribute poll | ~15 minutes       | ~2 minutes                         | 87% faster               |
| Time to generate event task list   | 30–60 minutes     | ~30 seconds (AI)                   | 98% faster               |
| Safety report anonymity guarantee  | Partial / none    | Technical (no auth)                | Strong guarantee         |
| Peer matching capability           | None              | ML-powered, 100% sep.              | New capability           |
| Admin visibility of activity       | Fragmented        | Unified audit log                  | Full coverage            |

## WORKFLOW

## VI. DISCUSSION

Three main findings are corroborated by the experimental results. First, the Vibe Matcher’s multi-factor blending approach demonstrably outperforms a pure cosine-similarity baseline on empirical same-cluster Top-3 retrieval [10–12].

The addition of scholarly contextual elements (branch and year) ensures that suggested matches are not only interest-similar but also practically feasible collaborators. Generative AI further enhances the system by providing meaningful summaries and automated planning [18, 19]. Nonetheless, the system relies on external AI services, which may introduce latency or availability issues. Realtime updates via WebSockets and support for multiple AI providers are identified as future improvements. Second, the generative AI extensions add significant value over existing model quality. Poll summaries are always more informative than just a vote count display, and take less than a second to create. All evaluated scenarios were found to have no manual corrections required for event task lists created by Gemini. However, the features are limited by the Gemini API availability and rate-limit constraints that institutions have to plan for. The current implementation includes a graceful degradation path, where in the case of unavailable AI service, a simple vote counting mechanism is used for polls and a blank Kanban board is used for events. Third, architecturally is the fact that the decision to do incident reporting is an unauthenticated endpoint (not an authenticated endpoint and a privacy flag) which is technically correct and pedagogically

important. A privacy flag can be set or removed by an administrator of the database to which the record is added. If the incident record doesn't include any user identifier, then that is a permanent change. An anonymous reporting infrastructure for institutions that want to offer trustworthy safety channels is a credible blueprint that is demonstrated on this platform [16, 17, 15]. Some of the current limitations are no real-time push notifications (the UI checks on page loading, not event driven responses), no mobile native app and a single dependency with generative AI provider. These will be tackled in future with WebSocket integration, a React Native port and an abstracted AI service layer with multiple providers.

## VII. CONCLUSION

This paper proposed a unified student-centric platform which is free from the overlapping digital tools that are experienced in higher learning institutions [20, 21]. The platform combines seven main functions into a single end-to-end, web application: peer matching, intelligent polling, event scheduling, resource sharing, anonymous safety reporting, announcement, and administration, all running on Django, React and Supabase (PostgreSQL). The Vibe Matcher is a combination of TF-IDF and Cosine similarity with a four factor weighted secondary scoring system that gives 100% correct cluster separation for nine interest clusters in a controlled 50-user experiment.

Collective poll insights and auto-generated task lists in Smart Polls and the Event Planner are driven by the Google Gemini generative AI API [18, 19]. With Safety Shield, the incident report is sent to an endpoint completely removed from the JWT authentication layer, ensuring a true technical anonymity of the incident. Comparative analysis shows that this platform can save 6 or more tools to complete a full campus workflow to 1, save 98% in time to create tasks for events and provide peer matching and consolidated activity monitoring capabilities that are lacking in fragmented baselines. The system can be deployed and customized for other institutions with a single Docker Compose command and through the use of seed-data and environment configuration. The overall theme of this work is that the intelligent services (recommendation, summarization and generation) can be added to student daily tools in a modular, API-first application. This platform's ability to solve existing pain points with AI, without introducing anything new, highlights a blueprint for the smarter, simpler, and safer digital campus infrastructure of the future.

## REFERENCES

- [1] Aalto University: Clustering Students’ Open-Ended Questionnaire Answers (2018)
- [2] Katz, A., et al.: Using Natural Language Processing to Facilitate Student Feedback Analysis (2021)
- [3] Sützen, N., et al.: Automatic Short Answer Grading and Feedback Using Text Mining Methods (2020)
- [4] Hadifar, A., et al.: A Self-Training Approach for Short Text Clustering (2019)
- [5] Thanh, T., et al.: Perfect Match: Facilitating Study Partner Matching (2019)
- [6] Interactive Recommender System for Identifying Learning Partner

- (2021)
- [7] Oktavia, T., et al.: A Meta-Learning Recommender System Framework for Learning Partner Identification
  - [8] Amiri, S., et al.: Educational Customization by Homogenous Grouping of e- Learners (2024)
  - [9] Schwartz, M., et al.: Peer Study Groups in Medical School (2025)
  - [10] Li, Y., et al.: Genetic Algorithm-Enabled Group Formation (2022)
  - [11] Revelo Sánchez, O., et al.: Genetic Algorithms for Collaborative Learning Groups (2021)
  - [12] Liang, J., et al.: Optimizing Group Formation with Mixed Genetic Algorithm (2025)
  - [13] Halim, A.: AI in Event Management Industry (2023)
  - [14] Burlov, N., Yakovlev, I.: AI in Event Management Automation (2025)
  - [15] Poornima Help: Anti-Ragging Application
  - [16] CampusShield Anti-Ragging Portal (2025)
  - [17] Report Bullying App (Real-World Application)
  - [18] Gemenis, K.: AI in Voting Advice Applications (2024)
  - [19] Buryakov, A., et al.: BERT for Voting Advice Applications (2024)
  - [20] Nkomo, L., et al.: Student Engagement with Digital Technologies (2021)
  - [21] Osama, M., et al.: Student Engagement in E-Learning (2022)