

National AI Institute for Exceptional Education

Guided by Professor: Jinjun Xiong, TA: Yuting Hu

Team members

Dhaval Patel, Kashyap Soni, Shubham Kulkarni, Divya Muralidharan, Attharva Kulkarni

1. Introduction & Motivations

The National AI Institute's mission to leverage artificial intelligence (AI) for exceptional education necessitates a robust and dynamic online presence. Recognizing this, our student team embarked on developing a comprehensive website featuring dedicated admin and member portals, a content management system for research papers, and portals for monthly reports, news, and events.

This platform serves as a nexus for educators, researchers, and students to engage with AI's transformative role in education. It aims to facilitate the seamless exchange of ideas, findings, and pedagogical strategies, thus fostering an environment of innovation and inclusivity in special education. By providing real-time access to research, resources, and collaborative opportunities, the website is envisioned to become a cornerstone for AI-driven educational excellence.

In pursuit of this vision, the technical architecture has been meticulously designed to ensure reliability, scalability, and user-centric functionality. The following sections will delve into the intricate technicalities of the project, the open issues that persist, the future work that beckons, and the individual contributions that made this endeavor a reality.

2. Technical Details of the Project

1. Tech Stack

1.1 Development Stack

- **Backend Language:** Python
- **Web Framework:** FastAPI
- **Database:** MySQL Server
- **Version Control:** Git
- **Libraries/Tools:**
 - Python 3.9.x
 - FastAPI 0.68.x
 - MySQL 8.x
 - Git 2.x
- **IDE:** Any code editor or IDE preferred by the development team.
- **API Testing:** Postman for API testing.
- **System Requirements:** The development machines should have at least 8GB of RAM and sufficient disk space for development purposes. Specific processor power requirements are not critical for development.

1.2 Deployment Stack

- **Production Environment:** Xlabs Kubernetes Cluster has been used to deploy the application container
- **Server Configuration:** The deployment uses a 1cpu Node and 1GB of storage
 - **Database:** MySQL Server (AWS RDS)
 - **Version Control:** GitHub
 - **Libraries/Tools:**
 - Python 3.11
 - FastAPI 0.68.0
 - MySQL 8.2
 - Git 2.x
 - **API Testing:** Postman for API testing.
- **System Requirements:** 1cpu, 4GB RAM, 1GB storage

2. Accounts and Infrastructure

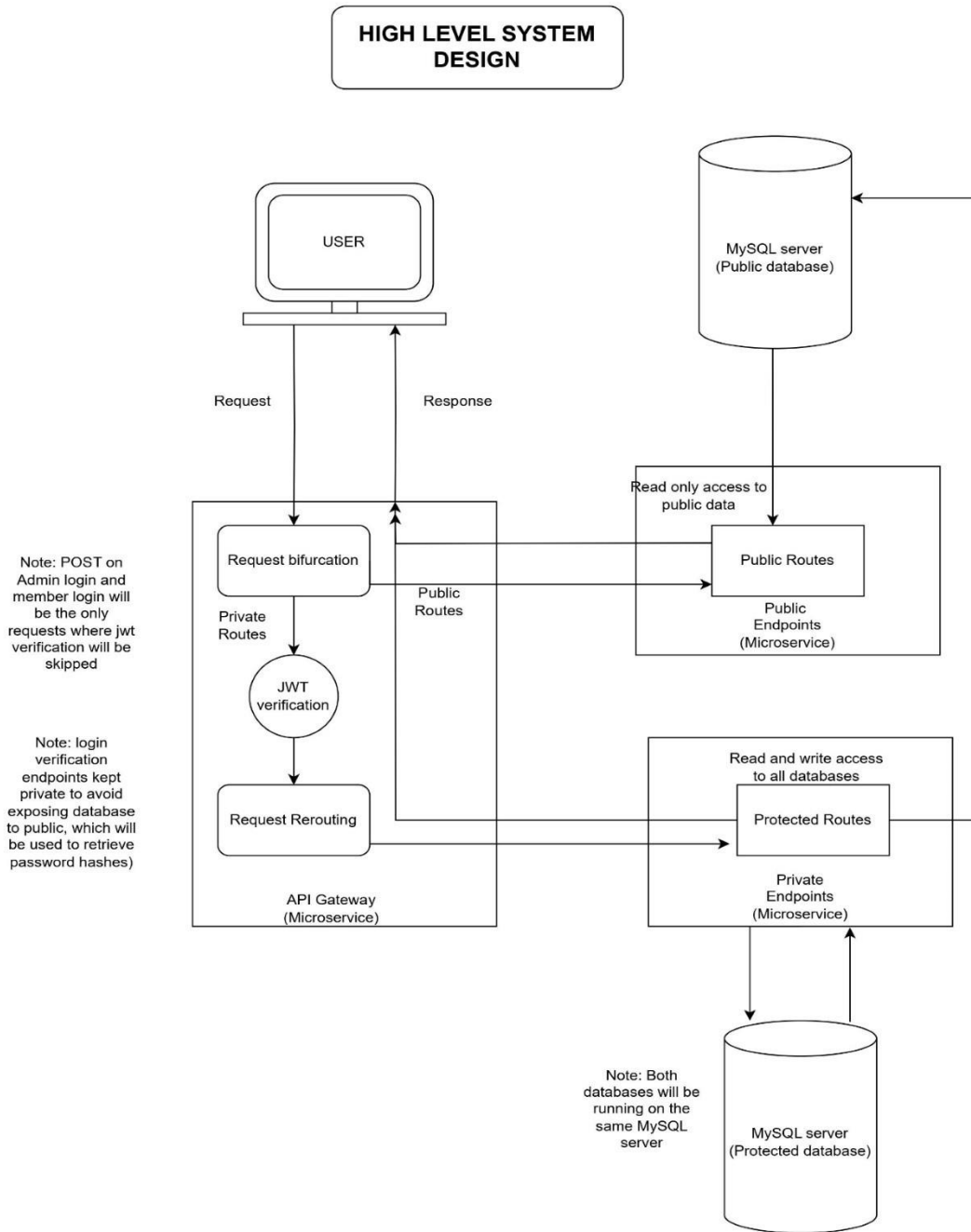
2.1 Development

- **Development Environment:** Local development machines for each developer.
- **Version Control Repository:** GitHub.
- **AWS Development Account:**
 - **URL:** [AWS Console](#)
 - **Account Name:** [Our AWS Account Name]
 - **Login Credentials:** [Developer's AWS IAM credentials]
- **Postman Account:**
 - **URL:** Postman
 - **Account Name:** [Our Postman Account Name]
 - **Login Credentials:** [Developer's Postman credentials]

2.2 Production

- **Production Environment:** AWS EC2 instances and RDS for MySQL.
- **AWS Production Account:**
 - **URL:** [AWS Console](#)
 - **Account Name:** [Our AWS Account Name]
 - **Login Credentials:** [Production Team's AWS IAM credentials]

3. System Architecture Diagram



4. Deployment Methodology

The deployment process for building and pushing a Docker image locally, and the subsequent Kubernetes deployment, follows the steps outlined below:

1. Building and Pushing Docker Image Locally:
2. Build the Docker image locally with the specified tag:
3. `docker build -t kashyap:latest .`
4. Tag the local image with the appropriate Docker Hub repository and version:
5. `docker tag kashyap kashyapsoni3010/kashyap:latest`
6. Push the tagged image to Docker Hub:
7. `docker push kashyapsoni3010/kashyap:latest`
8. Run the Docker container using the pushed image:
9. `docker run kashyapsoni3010/kashyap`
10. Kubernetes Deployment Process:
11. Upload the Docker container to Docker Hub as outlined in the previous steps.
12. Access the Kubernetes cluster using SSH.
13. Create the deployment.yaml file with the deployment configuration:
14. Apply the deployment configuration using the `kubectl apply` command:
15. `kubectl apply -f deployment.yaml -n aiinstituteweb`
16. Verify the deployment by checking the status of deployments:
17. `kubectl get deployments -n aiinstituteweb`
18. Expose the deployment by creating the service.yaml file:
19. Apply the service configuration using the `kubectl apply` command:
20. `kubectl apply -f service.yaml -n aiinstituteweb`
21. Verify the service by checking the status of services:
22. `kubectl get services -n aiinstituteweb`

This process ensures the deployment of the Docker container in a Kubernetes environment, exposing it with a LoadBalancer service for external access.

5. Postman API Testing

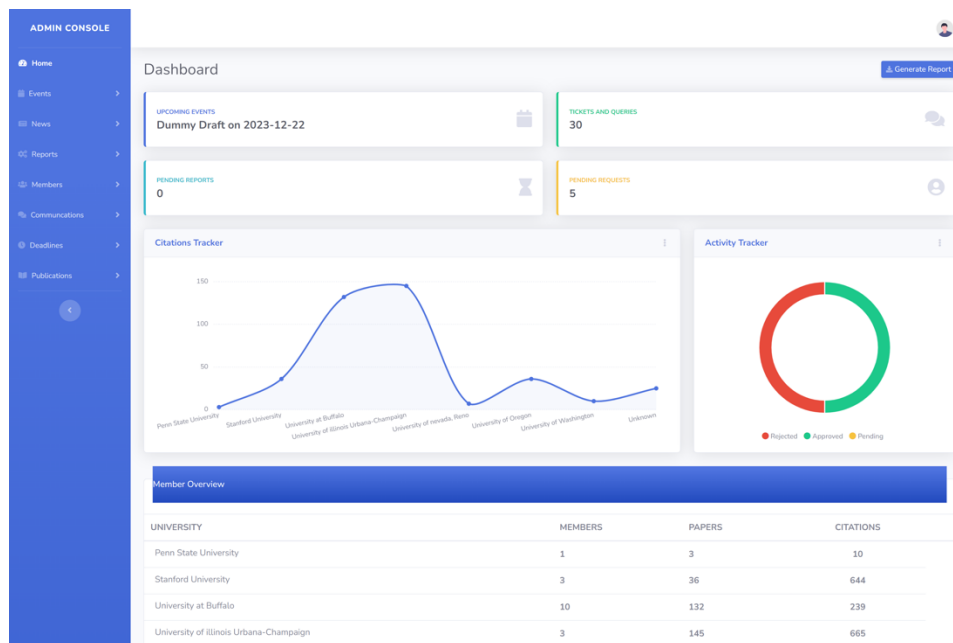
API testing for the National AI Institute website was conducted using Postman. The process involved these steps:

1. The Postman collection containing API requests and tests was imported into the Postman environment.
2. Development and production environment configurations were set up in Postman to manage different API endpoints and variables.
3. API requests were executed in Postman to verify the functionality of various endpoints, including user registration, data retrieval, and administrative operations.
4. Authorization and authentication mechanisms, including token-based authentication for secure API access, were tested using Postman.
5. Load testing and performance testing were performed using Postman to assess how the API performs under different traffic conditions.
6. Security testing was conducted with Postman, focusing on vulnerabilities like SQL injection and cross-site scripting (XSS).
7. Postman tests were automated and integrated into the CI/CD pipeline to continuously validate API functionality during development and deployment. Postman test results and reports were utilized to identify and address any issues in the API behavior, ensuring the reliability and security of the website's API endpoints.

6. Admin portal features

6.1 Admin portal homepage

This is the homepage for Admin portal. Here, admin can keep track of status of upcoming events, deadlines, pending requests, pending reports, tickets and queries and many others.

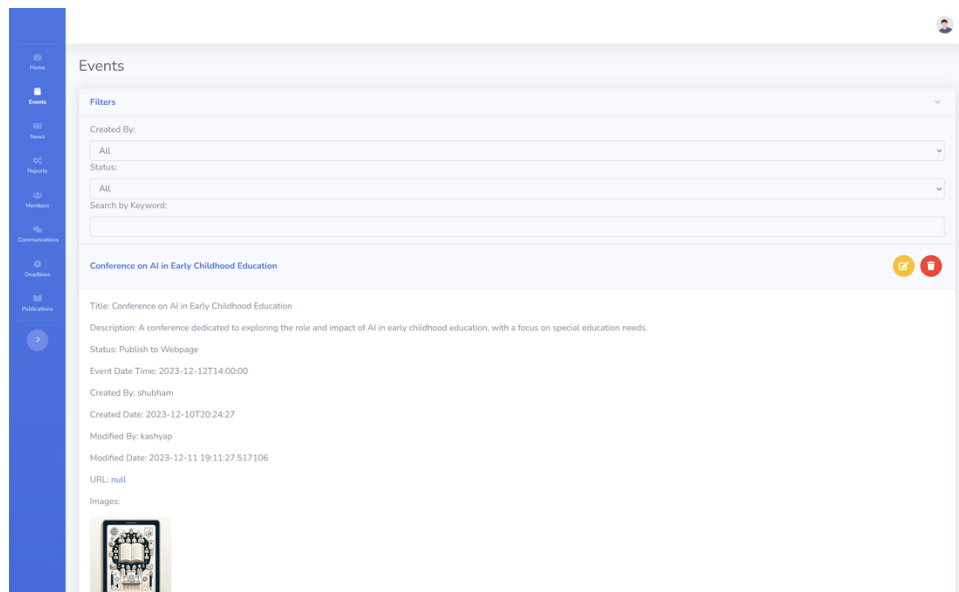


6.2 Admin portal add and view events

Here, admin can create or add new events. Admin can choose events status as draft, publish to webpage, or publish to members.

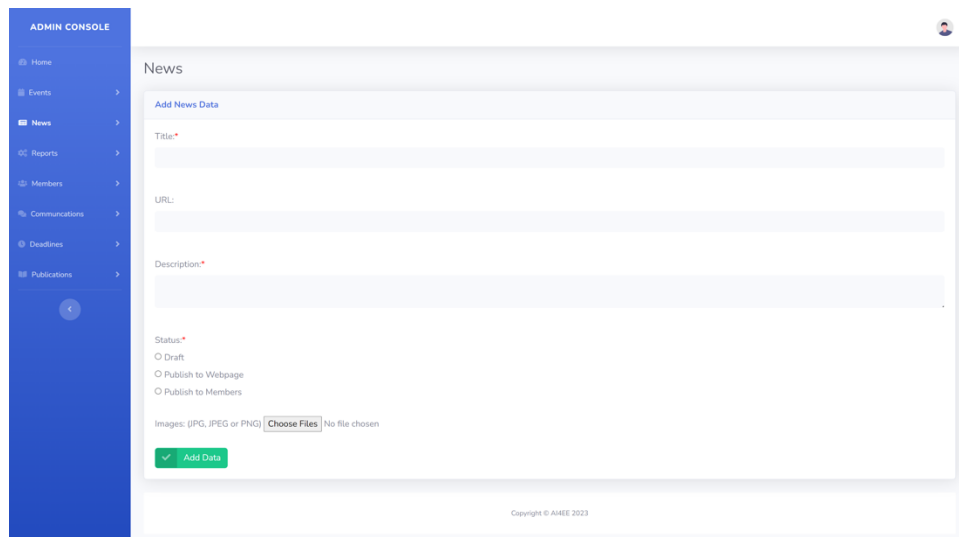
The 'Events' page in the Admin Console allows for adding new events. It features a sidebar with navigation options: Home, Events, News, Reports, Members, Communications, Deadlines, and Publications. The main content area is titled 'Events' and contains a form labeled 'Add Events Data'. The form includes fields for 'Title*', 'URL', and 'Description*'. The 'Status*' section offers three radio button options: 'Draft', 'Publish to Webpage', and 'Publish to Members'. The 'Event Date Time*' field is a date and time picker. At the bottom, there is a file upload section for 'Images: (JPG, JPEG or PNG)' with a 'Choose Files' button and a 'No file chosen' status. A green 'Add Data' button is located at the bottom of the form.

Admin can view the events which are added. Also, admin can edit or delete events.

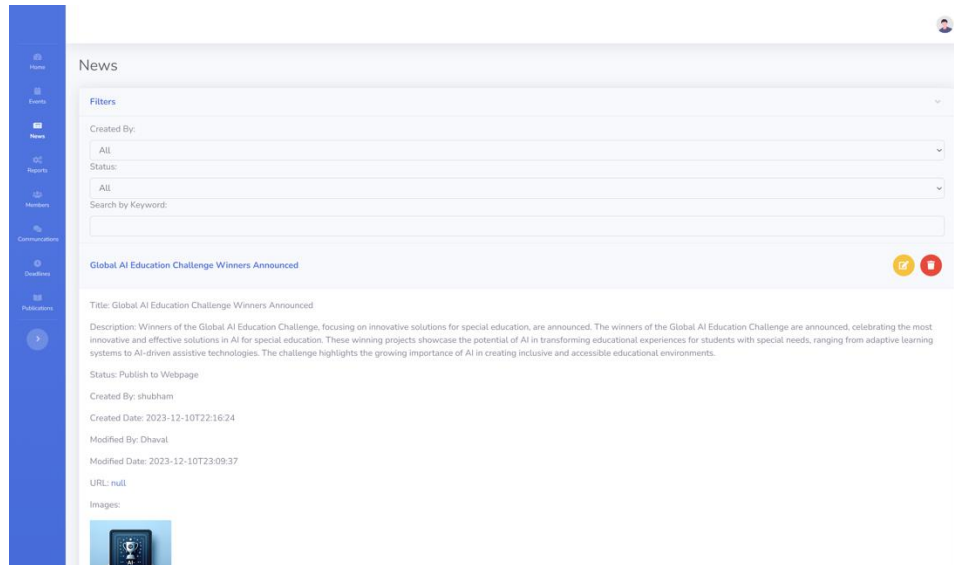


6.3 Admin portal add and view news

Here, admin can create or add news. Admin can choose news status as draft, publish to webpage, or publish to members.

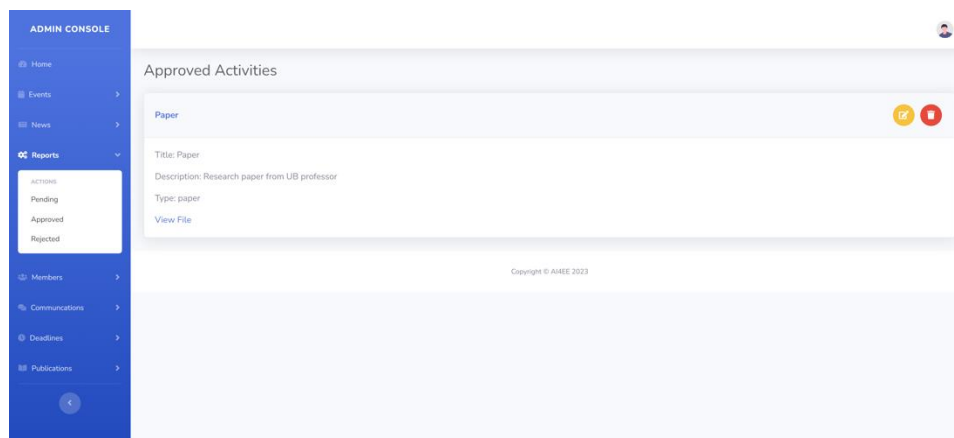


Admin can view news which is added. Also, admin can edit or delete news.



6.4 Admin portal reports

In the reports section admin can approve or reject reports which are uploaded by members. Also, admin can see pending reports.



6.5 Admin portal members

Admin can add, view, and approve new members in this section.

ADMIN CONSOLE

- Home
- Events
- News
- Reports
- Members**
 - Actions
 - Add Members
 - View Members
 - Approve Members
- Communications
- Deadlines
- Publications

Add Members

Full Name *

University *

Designation *

Copyright © AHAE 2023

6.6 Admin portal communications

In communications, admins can address to tickets and queries submitted by members,

ADMIN CONSOLE

- Home
- Events
- News
- Reports
- Members
- Communications**
 - Actions
 - Tickets
 - Queries
- Deadlines
- Publications

Tickets

Update report ✓

Name: Kashyap Soni
Description: This is a sample ticket description

aa ✓

Name: member
Description: aaaa

raising from docker ✓

Name: member
Description: docker says hello

6.7 Admin portal Deadlines

Admin can add submission deadlines of reports, research papers, news, events, and publications for members.

ADMIN CONSOLE

- Home
- Events
- News
- Reports
- Members
- Communications
- Deadlines**
- Publications

Submission Deadlines

Add Deadline Information

Title *

Description

Submission Month *

December 2023

Status *

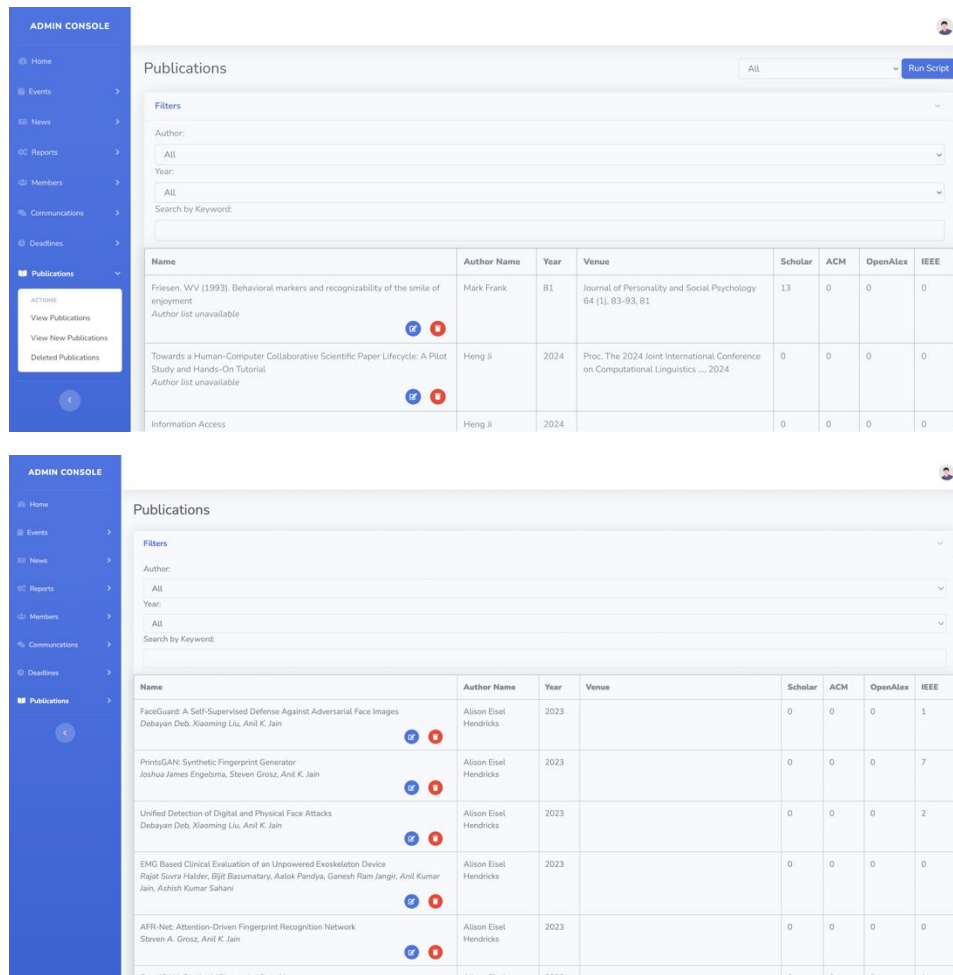
☐ Active
☐ Inactive

Deadline Date Time *

dd-mm-yyyy

6.8 Admin portal publications

Here, admin can see new publications that are scraped from Google scholar, ACM, OpenAlex and IEEE libraries. Also, admin can edit or delete the publications. Admin can also view the publications which are deleted. In addition, admin can view newly scraped entries in a separate section.



ADMIN CONSOLE

- Home
- Events
- News
- Reports
- Members
- Communications
- Deadlines
- Publications**
 - View Publications
 - View New Publications
 - Deleted Publications

Publications [All] [Run Script]

Filters

Author: [All]

Year: [All]

Search by Keyword: []

Name	Author Name	Year	Venue	Scholar	ACM	OpenAlex	IEEE
Friesen, WV (1993). Behavioral markers and recognizability of the smile of enjoyment Author list unavailable	Mark Frank	81	Journal of Personality and Social Psychology 64 (1), 83-93, 81	13	0	0	0
Towards a Human-Computer Collaborative Scientific Paper Lifecycle: A Pilot Study and Hands-On Tutorial Author list unavailable	Heng Ji	2024	Proc. The 2024 Joint International Conference on Computational Linguistics ... 2024	0	0	0	0
Information Access	Heng Ji	2024		0	0	0	0

ADMIN CONSOLE

- Home
- Events
- News
- Reports
- Members
- Communications
- Deadlines
- Publications**
 - View Publications
 - View New Publications
 - Deleted Publications

Publications [All] [Run Script]

Filters

Author: [All]

Year: [All]

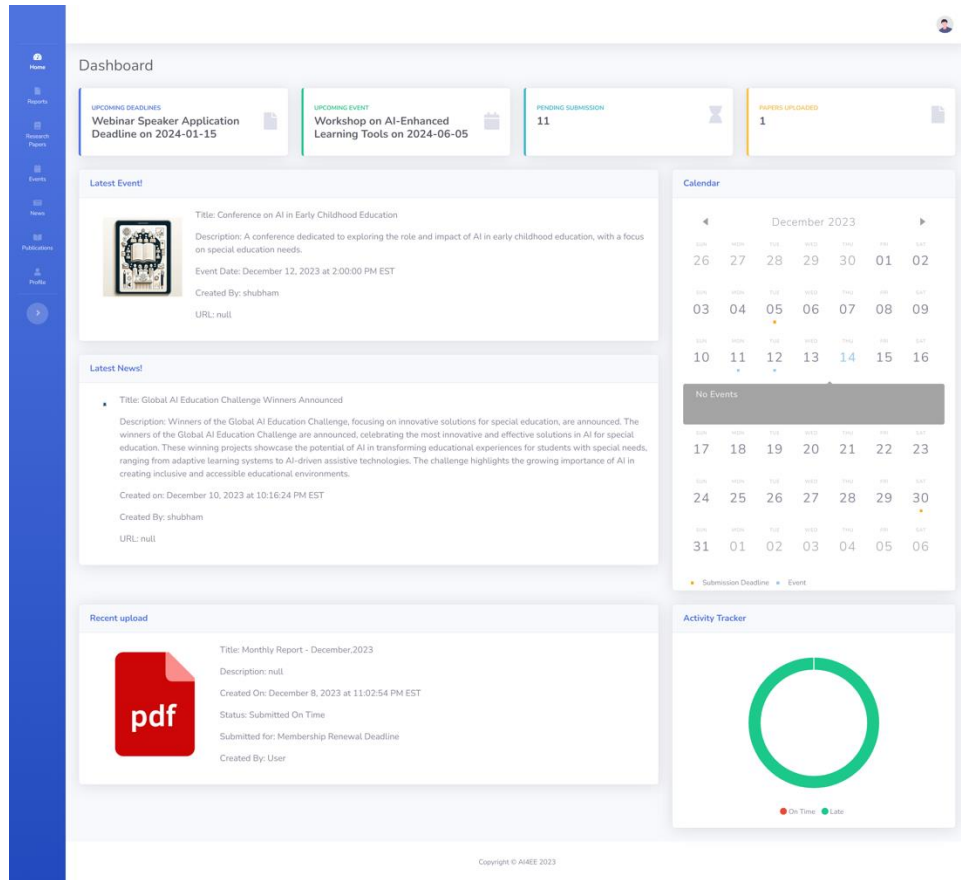
Search by Keyword: []

Name	Author Name	Year	Venue	Scholar	ACM	OpenAlex	IEEE
FaceGuard: A Self-Supervised Defense Against Adversarial Face Images Debayan Deb, Xiaoming Liu, Anil K. Jain	Alison Eisel, Hendricks	2023		0	0	0	1
PrintsGAN: Synthetic Fingerprint Generator Joshua James Engelma, Steven Grosz, Anil K. Jain	Alison Eisel, Hendricks	2023		0	0	0	7
Unified Detection of Digital and Physical Face Attacks Debayan Deb, Xiaoming Liu, Anil K. Jain	Alison Eisel, Hendricks	2023		0	0	0	2
EMG Based Clinical Evaluation of an Unpowered Exoskeleton Device Rajat Souva Halder, Bijit Basumatary, Aalok Pandya, Ganesh Ram Jangir, Anil Kumar Jain, Achish Kumar Sahani	Alison Eisel, Hendricks	2023		0	0	0	0
AFR-Net: Attention-Driven Fingerprint Recognition Network Steven A. Grosz, Anil K. Jain	Alison Eisel, Hendricks	2023		0	0	0	0
FaceGuard: A Self-Supervised Defense Against Adversarial Face Images Debayan Deb, Xiaoming Liu, Anil K. Jain	Alison Eisel	2023		0	0	0	1

7. Member portal features

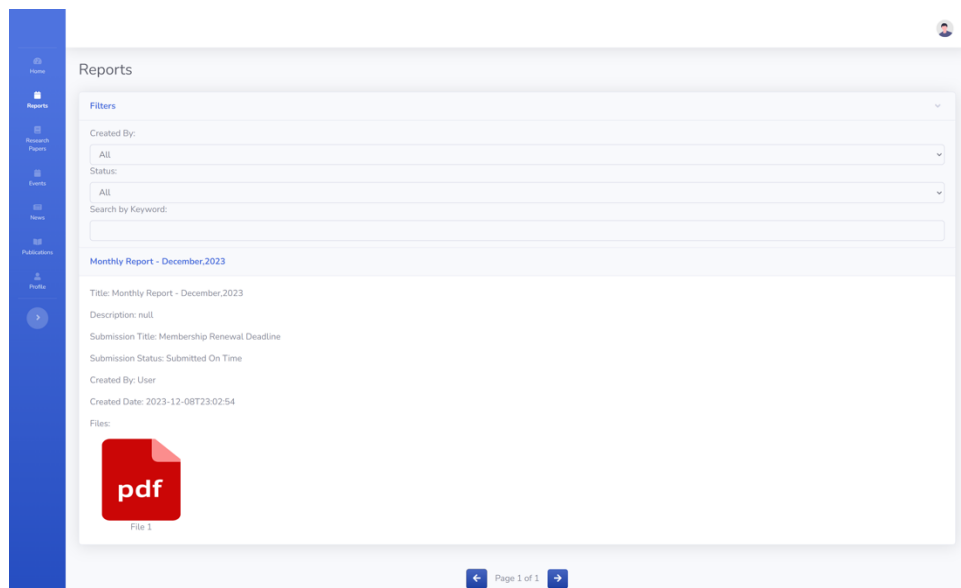
7.1 Member portal homepage

This is the homepage for the members. Here, members can see upcoming submission deadlines, upcoming events, pending submissions, papers uploaded, latest events and news, recently uploaded papers, and activity tracker.



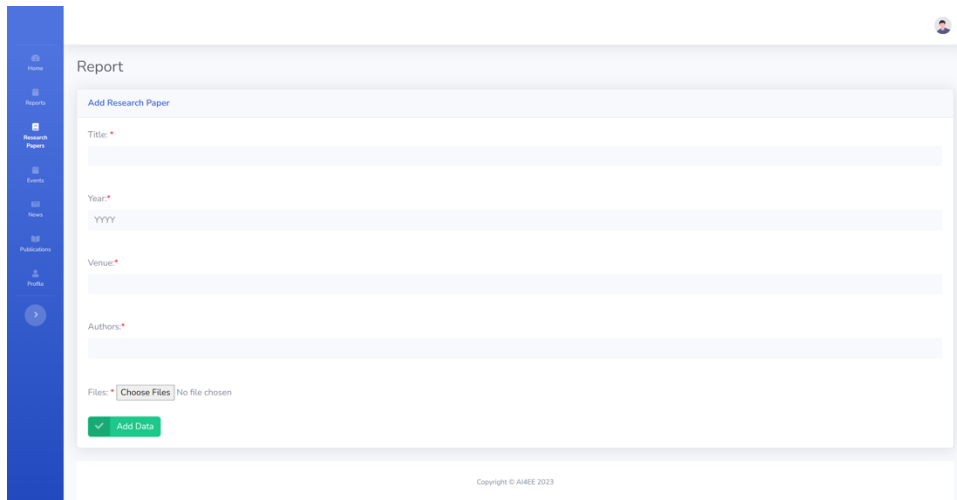
7.2 Member portal view report

Here, members can view reports which are submitted. Members cannot edit or delete submitted reports. Only admin can do it.



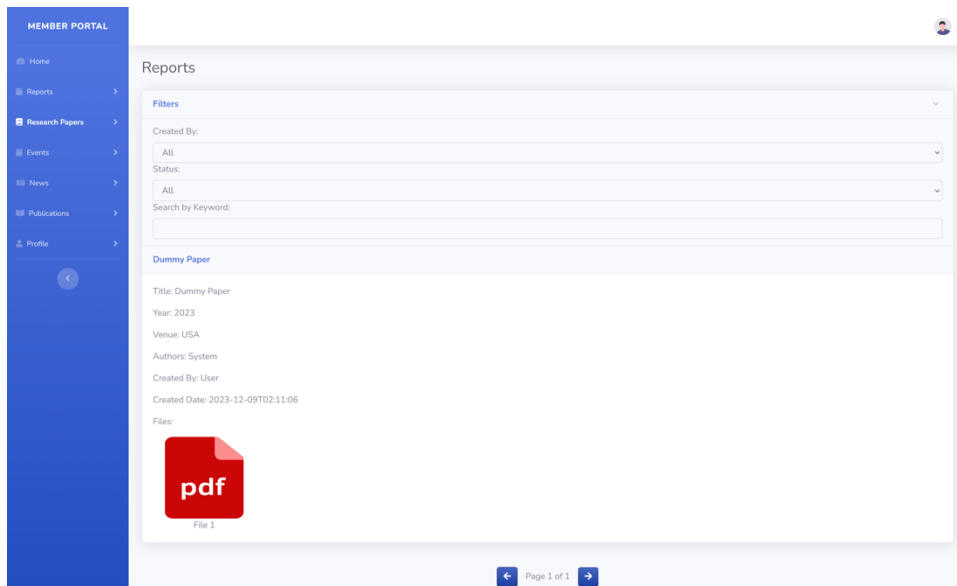
7.3 Member portal add and view research papers

Members can add new research papers.



The screenshot shows a web interface for adding a research paper. On the left is a blue sidebar with navigation links: Home, Reports, Research Papers (active), Events, News, Publications, and Profile. The main content area is titled 'Report' and contains a form titled 'Add Research Paper'. The form has fields for Title, Year (with a placeholder 'YYYY'), Venue, and Authors. Below these is a 'Files' section with a 'Choose Files' button and the text 'No file chosen'. At the bottom of the form is a green 'Add Data' button. A small copyright notice 'Copyright © AMEE 2023' is visible at the bottom of the page.

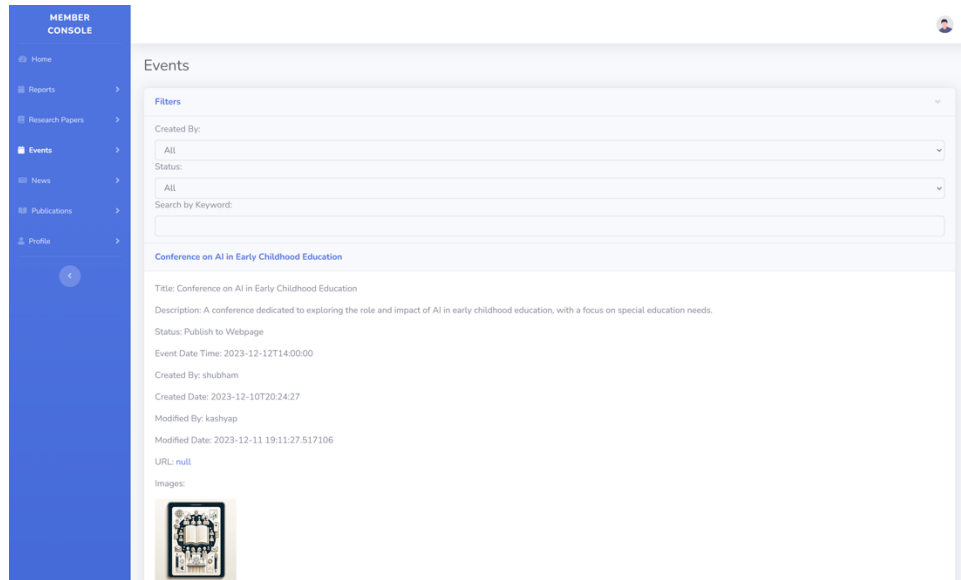
Members can view submitted research papers but cannot edit or delete the submission.



The screenshot shows the 'Reports' section of the member portal. The sidebar is the same as in the previous image. The main content area is titled 'Reports' and includes a 'Filters' section with dropdown menus for 'Created By' (set to 'All') and 'Status' (set to 'All'), and a 'Search by Keyword' input field. Below the filters is a card titled 'Dummy Paper' displaying the following information: Title: Dummy Paper, Year: 2023, Venue: USA, Authors: System, Created By: User, and Created Date: 2023-12-09T02:11:06. Under the 'Files' section, there is a red PDF icon labeled 'File 1'. At the bottom of the page, there is a pagination bar showing 'Page 1 of 1' with navigation arrows.

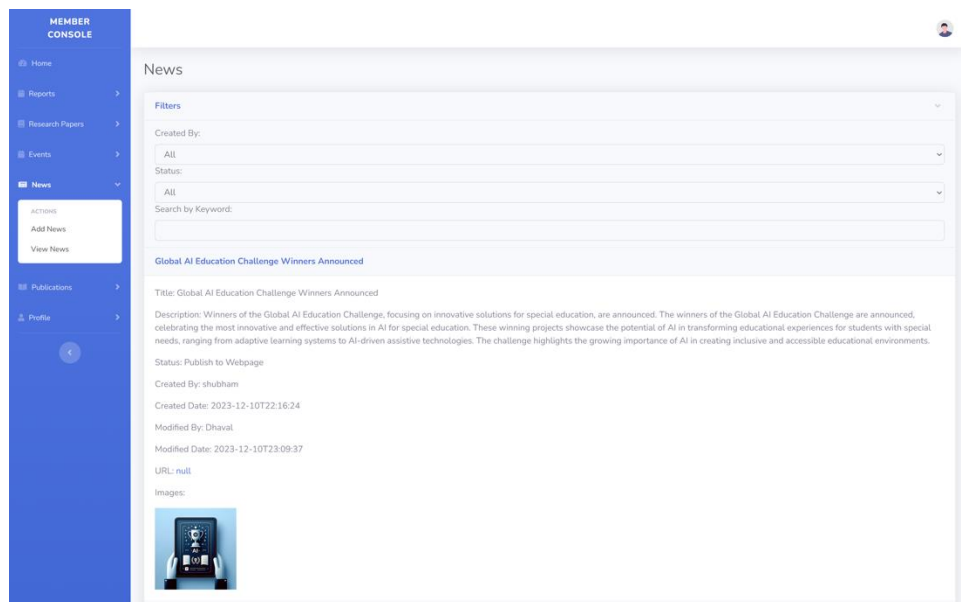
7.4 Member portal view events

Here, members can view events which are created by admins. Members cannot edit or delete events. Only admin has the authority to do it.



7.5 Member portal view news

Members can view the latest news in this section. Also, members can add news, but it needs admin approval.



7.6 Member portal view publications

Members can view new publications in this section. Members can also view their own publications and other member's publications as well.

MEMBER CONSOLE

Home

Reports

Research Papers

Events

News

Publications

Profile

View Your Publications

View Your New Publication

View Member Publications

Publications

Filters

Author:

All

Year:

All

Search by Keyword:

Name	Author Name	Year	Venue	Scholar	ACM	OpenAlex	IEEE
Friesen, W.V. (1993). Behavioral markers and recognizability of the smile of enjoyment. Author list unavailable	Mark Frank	81	Journal of Personality and Social Psychology 64 (1), 83-93, 81	13	0	0	0
Towards a Human-Computer Collaborative Scientific Paper Lifecycle: A Pilot Study and Hands-On Tutorial. Author list unavailable	Heng Ji	2024	Proc. The 2024 Joint International Conference on Computational Linguistics ... 2024	0	0	0	0
Information Access Author list unavailable	Heng Ji	2024		0	0	0	0

MEMBER CONSOLE

Home

Reports

Research Papers

Events

News

Publications

Profile

Tickets

View Your Publications

View Your New Publication

View Member Publications

Publications

Filters

Year:

All

Search by Keyword:

Name	Year	Venue	Scholar	ACM	OpenAlex	IEEE
Hear The Flow: Optical Flow-Based Self-Supervised Visual Sound Source Localization Author list unavailable	2023	Proceedings of the IEEE/CVF Winter Conference on Applications of Computer ... 2023	4	0	0	0
Deep metric learning for computer vision: A brief overview Author list unavailable	2023	Deep Learning 48, 59, 2023	1	0	0	0
NAPReg: nouns as proxies regularization for semantically aware cross-modal embeddings Author list unavailable	2023	Proceedings of the IEEE/CVF Winter Conference on Applications of Computer ... 2023	3	0	2	0
CoNAN: Conditional Neural Aggregation Network For Unconstrained Face Feature Fusion Author list unavailable	2023	arXiv preprint arXiv:2307.10237, 2023	0	0	0	0
Deep metric learning for computer vision: A brief overview Author list unavailable	2023	Deep Learning 48, 59, 2023	1	0	0	0
User Behaviour based Insider Threat Detection using a Hybrid Learning Approach Author list unavailable	2023	Journal of Ambient Intelligence and Humanized Computing 14 (4), 4573-4593, 2023	4	0	2	0

7.7 Member portal edit profile

Members can customize their profiles over here. Also, members can change password for their account.

MEMBER CONSOLE

Home

Reports

Research Papers

Events

News

Publications

Profile

View Your Publications

View Your New Publication

View Member Publications

Account settings

General

Change password

Research Profile links

Upload new photo

Reset

Username

User

First Name

Dhaval

Last Name

Patel

E-mail

dhavalja@buffa

Phone Number

000000000

University

Designation

Student

Save changes Cancel

Copyright © AIEEE 2023

8. Contributions:

Dhaval	<p>System Design and Documentation</p> <ul style="list-style-type: none">• Collaborated on High-level and Low-level system design for the application• Collaborated on documentation for high-level and low-level design• Collaborated on API Document for the backend• Collaborated on API usage documentation• Created the project plan for the project• Developed and Designed the ER Diagram for the project <p>Back-end:</p> <ul style="list-style-type: none">• Created the structure of Server routes for Admin, Member, and Public, and mounted folders for serving static files <p>Admin Portal: Back-end: Built API endpoints for the following functions:</p> <ul style="list-style-type: none">• Events: Create, view, edit and delete events• News: Create, view, edit and delete News• Deadlines: Create, view, edit and delete Deadlines <p>Member Portal: Back-end Built API endpoints for the following functions:</p> <ul style="list-style-type: none">• Home Page: Statistics, Calendar View, and all the data APIs• Reports: Create, view, edit and delete reports• Research Papers: Create, view, edit and delete research papers• Events: Create, view, edit and delete events• News: Create, view, edit and delete news <p>Landing Page: Back-end Built API endpoints for the following functions:</p> <ul style="list-style-type: none">• Static HTML Pages• Events Page• News Page <p>Front-End: Admin Portal: Front-end Created web pages for the following features:</p> <ul style="list-style-type: none">• Login and Logout Pages• Events: Create events page, view events page, edit and delete pages, ability to view uploaded images. Added functionality to change the status of events within the view page• News: Create news page, view news page, edit and delete pages, ability to view uploaded images. Added functionality to change the status of news within the view page• Deadlines: Create deadlines page, view deadlines page, edit and delete pages. Added functionality to change status of news within the view page
--------	---

	<p>Member Portal: Front-end Created pages for the following:</p> <ul style="list-style-type: none"> • Login and Logout Pages • Member Home: Designed and developed the entire Member Home page, along with the statistics and the integrated Calendar View to notify the upcoming events and deadlines • Reports: Create reports page, view events page, edit and delete pages, ability to view uploaded files • Research Paper: Create research paper page, view events page, edit and delete pages, ability to view uploaded files • Events: Create events page, view events page, edit and delete pages, ability to view uploaded images. Added functionality to change status of events within the view page • News: Create news page, view news page, edit and delete pages, ability to view uploaded images. Added functionality to change status of news within the view page • Profile Page: Update Profile information, change passwords, update research links as well as upload profile images <p>Landing page: Front-end:</p> <ul style="list-style-type: none"> • Integrated code written by the front-end developer to ensure that the pages are being sent by the public routes • Developed the Events and News pages, used JS DOM manipulation to send Events and News information to the front-end using public routes <p>For the front-end views of the member portal and admin portal, created JavaScript code to fetch, send data, and manipulate page contents using DOM manipulation to display data. Used Sweet Alerts library for displaying popups and adding functionality to the popups.</p> <p>API Testing:</p> <ul style="list-style-type: none"> • Used OpenAPI documentation of FastAPI to test every single endpoint of the API • Covered all different scenarios possible and created test cases to test them out • Used Postman to test out the endpoints using custom inputs from test cases • Fixed bugs and issues discovered during testing
Kashyap	<p>System Design and Documentation</p> <ol style="list-style-type: none"> 1. Created High level and Low-level system design for the application 2. Created documentation for high level and low-level design. 3. Designed API for the back end 4. Created API usage documentation <p>Back-end: Created the structure of Server routes, mounted folders for serving static files</p> <p>Admin Portal: Back-end Created session management, session continuation, auto session termination, and logout token management logic for the admin portal Built API endpoints for the following functions:</p>

1. Login, logout logic
2. Home Page: Member statistics, overall analysis, view all data
3. Reports: add reports, view reports, approve reports, edit reports
4. Members: add members, view members, edit members, approve members
5. Tickets: Create tickets, view tickets, approve tickets, resolve tickets
6. Queries: View queries, respond to queries
7. Files: View files, view images, add files, add images

Member Portal: Back-end

Created session management, session continuation, auto session termination, and logout token management logic for the member portal.

Built API endpoints for the following functions:

Tickets: Create tickets, view tickets, view responses

Front-End:

Admin Portal: Front-End

Created web pages for the following features:

1. Admin Home: Fetch and load data from server, created charts, and added responsiveness
2. Reports: View reports page, approve reports page, reject reports page, view report file. Added functionality to change status of reports from all the pages.
3. Members: View members page, approve new members, remove members, add new member page. Added functionality to perform the above-mentioned functions to make the pages dynamic.
4. Tickets: Created page to view unresolved tickets and added functionality to resolve the tickets and close them upon resolution.
5. Queries; Created queries page and added functionality to respond to open tickets and close them on responding.

Member Portal: Front-End

Created pages for the following:

Tickets: Created page to raise tickets, view unresolved tickets. Added functionality to view the response of a resolved ticket.

For the front-end views of member portal and admin portal, created javascript code to fetch, send data and manipulate page contents to display data. Used SweetAlerts library for displaying popups and adding functionality to the popups.

API testing:

1. Used OpenAPI documentation of FastAPI to test every single endpoint of the API.
2. Covered all different scenarios possible and created testcases to test them out.
3. Used Postman to test out the endpoints using custom inputs from testcases.
4. Fixed bugs and issues discovered during testing.
5. Used Grafana K6 library to simulate load on the server and tested performance of the API under varied rates of requests hitting the server per second.

Deployment:

1. Created Dockerfile and docker-compose files to containerize the application.
2. Fixed key issues in containerizing the application
3. Built docker image and pushed it to DockerHub

	<ol style="list-style-type: none"> Created a deployment and a service on the Xlabs Kubernetes cluster Configured the service to pull the docker image and run the deployment Tested the functionality of deployed service. Fixed bugs and created documentation for building image and deploying it on Kubernetes cluster.
Shubham	<p>Documentation:</p> <ol style="list-style-type: none"> Collaborated on the creation of requirement drafts, technical design documents, and the final project report to ensure alignment with frontend development goals and requirements. <p>Backend:</p> <ol style="list-style-type: none"> Worked closely with the backend team to define and consume API endpoints for the member portal, ensuring seamless communication between the frontend and backend systems. Collaborated in designing and developing the member portal events page, allowing users to add and view events, ensuring a responsive and user-friendly interface. Contributed to the design and layout of the homepage for the member portal, focusing on features like calendar views, upcoming events, and upcoming deadlines. Implemented the front-end components for login and logout functionality, ensuring a smooth and secure user experience in member portal. Facilitated the integration of PDF files within events, enabling users to input PDF files as part of event details. <p>Database:</p> <ol style="list-style-type: none"> Collaborated with the database team to design and implement the database schema for events, news, deadlines, and reports, ensuring efficient data storage and retrieval. Assisted in database cleanup efforts to remove unnecessary data, improving overall database performance. Generated dummy data for events, news, and deadlines to support frontend development and testing. <p>Testing:</p> <ol style="list-style-type: none"> Actively participated in functional testing for the member portal, verifying the proper functionality of each frontend feature in collaboration with the testing team. Supported performance testing for the member portal to assess the responsiveness and efficiency of front-end components. Engaged in usability testing to evaluate the user-friendliness of the UI on various devices, providing valuable feedback for improvements. <p>Frontend:</p> <ol style="list-style-type: none"> Designed and developed the events page for the member portal, enabling users to add and view events with an emphasis on a responsive and intuitive user interface. Contributed to the design and layout of the member portal homepage, incorporating calendar views, upcoming events, and upcoming deadlines to enhance the user experience.

Divya	<p>Documentation:</p> <ul style="list-style-type: none"> Collaborated on the requirement document, technical design, and final report to ensure alignment with goals and requirements. <p>Scraping:</p> <ul style="list-style-type: none"> Researched and experimented with available libraries that can be used to implement scraping features. Implemented python scripts to crawl publication details for institution members from the following sources: Google Scholar, IEEE, ACM, OpenAlex, DBLP. Implemented separate scripts for each source mentioned above. Modified python scripts as required when change of requirements occurred. <p>Admin Portal: Front-End:</p> <ul style="list-style-type: none"> Implemented front-end views for displaying scraped publication data in admin portal. Created the following views: <ul style="list-style-type: none"> View all publications View deleted entries Edit publication details Run all scripts from home page View newly scraped publications Delete publications with comments to justify deletion <p>Member Portal: Front-End:</p> <ul style="list-style-type: none"> Created the following views for publications section in member portal: <ul style="list-style-type: none"> View user's publications View all publications Edit publication details View newly scraped publications. Delete publications with comments to justify deletion <p>Back-End:</p> <ul style="list-style-type: none"> Built API endpoints for the following functions in both member and admin portal, tested the API endpoints using Postman: <ul style="list-style-type: none"> View Publications Hide Publications Get Publication Data Edit Publication Data View Deleted Publications Get New Publication Data Run Publication Scripts. <p>Testing:</p> <ul style="list-style-type: none"> Contributed to functional testing of member portal, verifying functionality and reporting issues as and when detected.
Atharva	<p>User Interface and Theme Research</p> <ul style="list-style-type: none"> Conducted extensive research on the user interfaces of 15 different AI Institute websites from various universities. Analysed these websites to determine effective design elements, colour palettes, and fonts. Selected the most suitable colour palette and fonts that would resonate with the intended audience and enhance the institute's online presence.

	<ul style="list-style-type: none"> This research informed the final design decisions, ensuring a professional and appealing interface for the website. <p>Frontend Development</p> <p><i>UI Design and Implementation</i></p> <ul style="list-style-type: none"> Developed the user interface of a 10-12 pages website using HTML, CSS, JavaScript, and Bootstrap, focusing on creating an intuitive and visually appealing experience. Paid special attention to the layout and colour scheme to align with the website's branding and user experience objectives. <p><i>Theme Options</i></p> <ul style="list-style-type: none"> Integrated two theme options to choose, enhance user experience and provide visual variety. Enabled easy customizable viewing experience. <p><i>Responsive Design and Media Queries</i></p> <ul style="list-style-type: none"> Implemented responsive design principles to ensure smooth operation across various devices, including desktops, tablets (like iPads), and mobile phones. Used media queries in CSS to dynamically adapt the layout and content based on the device's screen size, ensuring consistency across different platforms. <p><i>Transitions and Animations</i></p> <ul style="list-style-type: none"> Incorporated CSS transitions and JavaScript animations to add visual interest and interactivity to the website. Ensured that these elements were engaging yet did not detract from the website's overall performance. <p><i>Google Maps Integration</i></p> <ul style="list-style-type: none"> Embedded Google Maps to display relevant geographic locations, adding a practical element for users. Customized the map feature to align with the site's responsive and aesthetic design. <p><i>JSON DOM File for Events and News</i></p> <ul style="list-style-type: none"> Created a JSON DOM file to manage and display dynamic content such as events and news. Implemented efficient data handling and DOM manipulation for real-time content updates, enhancing the site's interactivity and relevance. <p><i>Image Gallery and Team Sections</i></p> <ul style="list-style-type: none"> Designed and implemented an image gallery and team sections, highlighting visuals and team profiles attractively. Employed caching and lazy loading techniques for the image gallery to ensure swift and seamless loading of images, enhancing the user experience without compromising site speed. <p><i>Product Section</i></p> <ul style="list-style-type: none"> Developed a product section to highlight two key products of the AI institute. Designed this section to be visually compelling and informative, providing essential product details and images. <p><i>Podcast Section</i></p> <ul style="list-style-type: none"> Created a podcast section where users can view and listen to podcasts. Ensured this feature was user-friendly, with easy navigation and controls for an enjoyable listening experience.
--	--

	<p>Testing</p> <ul style="list-style-type: none"> • Focused on frontend-specific testing to ensure every aspect of the user interface worked seamlessly across different devices and browsers. • Conducted tests for responsiveness, ensuring that the layout, images, and text adapted correctly to various screen sizes, especially for desktops, tablets, and mobile phones. • Verified the functionality of interactive elements such as transitions, animations, and the Google Maps integration. • Tested the dynamic loading and rendering of content from the JSON DOM file, ensuring accurate and timely updates of events and news. • Tested the lazy loading of gallery images and podcast section. • Utilized browser developer tools and automated testing frameworks to simulate different environments and identify any potential issues. • Addressed all identified issues to enhance the overall user experience and ensure the website met its design and functionality goals. <p>Challenges and Solutions</p> <p>Challenge: Cross-Device Compatibility and Performance</p> <ul style="list-style-type: none"> • Problem: Balancing a responsive design for various devices, including desktops, iPads, and mobile phones, while ensuring smooth performance, especially with integrated animations and Google Maps. • Solution: Employed CSS media queries for responsive layouts and optimized JavaScript animations for performance. <p>Challenge: Dynamic Content Management</p> <ul style="list-style-type: none"> • Problem: Efficiently managing dynamic content from a JSON DOM file without impacting website performance. • Solution: Implemented effective data handling and DOM manipulation strategies for News and Events. <p>Search Engine Optimization (SEO)</p> <ul style="list-style-type: none"> • Implemented SEO best practices to improve the website's visibility and ranking on search engines. • Optimized metadata, keywords, and content structure for enhanced search engine indexing and relevance. • Created a sitemap XML file to facilitate more efficient crawling and indexing by search engines, ensuring that all pages of the 10-page website were recognized and properly categorized. This inclusion aids in improving the site's overall SEO performance.
--	--

9. Open issues and future work:

To further enhance user engagement and provide timely, interactive content, the News and Events pages will undergo a significant redesign. The future version will implement:

- **Interactive Features:** Real-time updates, event countdowns, and live notifications to keep users informed and engaged.
- **Modern Front-end Frameworks:** Transition to React for more dynamic and responsive page components that can handle real-time data without reloading.

- **Analytics Integration:** Tools to analyze user behavior and content popularity, driving data-informed improvements.
- **Community Interaction:** Features to allow community contributions, comments, and discussion threads related to news and events.

These enhancements will aim to create an immersive, informative, and community-centric experience on the AI4EE website.