

PROFESSIONAL TRAINING REPORT - II

entitled

STUD GRAPH: STUDENT PERFORMANCE VISUALIZATION DASHBOARD

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering with
specialization in Cyber Security

by

Harrish K [41614032]

Mohanakrishna S [41614057]

Kamesh D [41614041]

Heartlin Gardon B F [41614033]



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

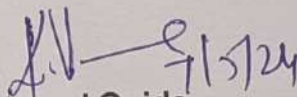
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

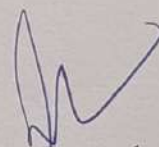
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

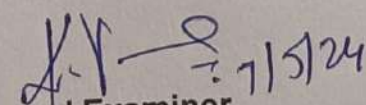
This is to certify that this Professional Training Report is the bonafide work of Mr.Harrish K (41614032), Mr.Kamesh D (41614041), Mr.Heartlin Gardon B F (41614033), Mr.Mohanakrishna S (41614057) who carried out the project entitled "STUD GRAPH: STUDENT PERFORMANCE VISUALIZATION DASHBOARD" under my supervision from January 2024 to May 2024.

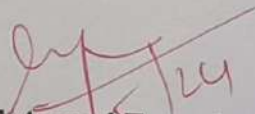

Internal Guide

Dr. K. Veena, M.E., Ph.D.,


Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 07.05.2024


Internal Examiner

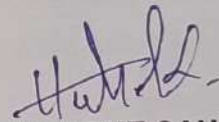

External Examiner

DECLARATION

I, **HARRISH K (41614032)**, hereby declare that the Professional Training Report-II entitled **"STUD GRAPH: STUDENT PERFORMANCE VISUALIZATION DASHBOARD"** done by me under the guidance of **Dr. K. Veena, M.E., Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Cyber Security.

DATE: 07/05/2024

PLACE: CHENNAI



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean, School of Computing, Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Dr. K. Veena, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-2 Professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

COURSE CERTIFICATE



Elite

NPTEL Online Certification

(Funded by the MoE, Govt. of India)

This certificate is awarded to

for successfully completing the course

Cloud Computing and Distributed Systems



with a consolidated score of %

Online Assignments		Proctored Exam	
--------------------	--	----------------	--

Total number of candidates certified in this course:

Batish

Prof. B. V. Batish Kumar
Chairman, Centre for Continuing Education
IIT Kanpur

Jan-Mar 2024

(8 week course)

Satyaki

Prof. Satyaki Roy
NPTEL Coordinator
IIT Kanpur



Indian Institute of Technology Kanpur



Roll No. NPTEL24CS09S353401108

To verify the certificate



No. of credits recommended: 2 or 3

ABSTRACT

The "Stud Graph: Student Performance Visualization Dashboard" is a groundbreaking web-based application meticulously crafted to revolutionize education by amplifying student learning outcomes and bolstering educator support through optimized teaching strategies. Seamlessly integrating advanced data visualization techniques with intuitive interfaces, this project serves as a beacon of innovation, furnishing students and educators alike with invaluable insights into academic performance across diverse subjects and semesters. Tailored to cater to the individualized needs of students, the dashboard offers a personalized experience, enabling them to meticulously monitor their progress, identify areas necessitating improvement, and establish ambitious academic goals. Through the utilization of interactive graphs and personalized performance messages, students gain unparalleled visibility into their achievements and receive tailor-made recommendations to enrich their learning journey. For educators, the dashboard emerges as a potent instrument, facilitating in-depth analysis of class performance, pinpointing curriculum gaps, and delivering targeted support to students. By aggregating and visualizing student performance data, educators are empowered to make well-informed decisions regarding instructional strategies, resource allocation, and intervention measures. Key features of the Student Performance Visualization Dashboard encompass interactive graphs providing visual representations of student performance data, personalized insights through performance messages tailored to individual student needs, tools and analytics facilitating data-driven decision-making for educators, and a collaborative platform fostering communication and collaboration among students, educators, and administrators to nurture a culture of continuous improvement in education. With its unwavering commitment to empowering both students and educators with actionable insights, the Student Performance Visualization Dashboard aspires to cultivate a culture of continuous improvement, thereby propelling education towards enhanced learning outcomes and unprecedented student success.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	vi
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 Overview	
2	LITERATURE SURVEY	2
	2.1 Survey	
3	REQUIREMENTS ANALYSIS	
	3.1 Objective	4
	3.2.1 Hardware Requirements	
	3.2.2 Software Requirements	6
4	DESIGN DESCRIPTION OF PROPOSED PRODUCT	7
	4.1.1 Ideation Map/Architecture Diagram	7
	4.1.2 Various stages	8
	4.1.3 Internal or Component design structure	8
	4.1.4 Working Principles	12
	4.2 4.2.1 Novelty of the Project	16
5	RESULTS AND DISCUSSION	18
6	SUMMARY AND CONCLUSION	
	REFERENCES	19
	APPENDIX	
	A. Research Paper	20
	B. Source Code	23

LIST OF FIGURES

Figure No.	Figure Name	Page No.
4.1.(i)	Ideation Map for Stud Graph	7
4.1.3.(i)	Code for uploading Students semester-wise marks	9
4.1.3.(ii)	Code for creating the graph	9
4.1.3.(iii)	Code for styling the graph	10
4.1.3.(iv)	Code for adding buttons and labels	10
4.1.4(i)	Graphical representation of marks	13
4.2.(i)	Linear Graphical representation of overall marks	14
4.2(ii)	Student Information	15
Fig – 1	System Architecture	22
Fig – 2	Implementation	22

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The "Stud Graph - Student Performance Visualization Dashboard" project offers a comprehensive solution for visualizing and analyzing student academic performance across multiple semesters. With a user-friendly interface and interactive graphs, it aims to provide valuable insights to students, educators, and academic institutions.

At its core, the project comprises two main components: the graph container and the sidebar. The graph container serves as the canvas where subject-wise marks and overall CGPA are displayed in visually appealing formats. Utilizing HTML, CSS, and JavaScript, along with the Chart.js library, the project ensures responsiveness and interactivity, enabling users to explore data with ease.

The sidebar complements the graph container by offering essential features such as student information, performance messages, and navigation buttons. Student information, including name, registration number, and profile picture, adds a personalized touch to the interface. The performance message section provides tailored feedback based on the student's academic performance, guiding them towards areas of improvement. The semester buttons allow users to select specific semesters of interest, while the overall button provides an overview of the student's CGPA trend across all semesters.

Underpinning the project is a robust dataset comprising sample data for subject marks and CGPA across various semesters. While the provided data serves as a foundation, the project's flexibility allows for seamless integration with real-time data from academic records or databases. This ensures that the visualizations accurately reflect current student performance, making them more relevant and actionable.

The technology stack employed in the project underscores its modern and versatile nature. HTML, CSS, and JavaScript provide the foundation for building the user interface and handling interactivity, while the Chart.js library facilitates the creation of dynamic and visually appealing graphs. This technology stack not only ensures a smooth user experience but also enables future enhancements and customizations to meet evolving needs.

Looking ahead, the project holds significant potential for further development and expansion. Future enhancements could include integration with student databases for real-time data updates, additional features such as trend analysis and peer comparisons, and customization options for users to personalize their viewing experience.

CHAPTER 2

LITERATURE REVIEW

2.1 SURVEY

The visualization of student academic performance has gained significant attention in educational research and practice. Through the use of graphical representations, educators and stakeholders can gain insights into student progress, identify areas of improvement, and make data-driven decisions to enhance learning outcomes. This literature review explores key studies and trends related to the visualization of student academic performance, focusing on the role of graphs in facilitating data interpretation and decision-making.

Visual Analytics in Education: Visual analytics, a multidisciplinary field combining visualization, human-computer interaction, and data analytics, has emerged as a powerful tool for understanding complex educational data. In their paper "Visual Analytics in Education: Enabling Researcher Understanding of Student Behaviour Through Large-Scale Data Visualization," Pea et al. (2014) discuss how visual analytics techniques can help researchers and educators make sense of large-scale educational datasets. By visualizing student interactions, performance metrics, and learning trajectories, researchers can gain deeper insights into student behavior and engagement, leading to more informed instructional strategies.

Graphical Representations of Student Performance: Graphical representations, including line graphs, bar charts, and scatter plots, are commonly used in educational contexts to visualize student performance data. In their study "Using Visualizations to Analyze Student Performance," Smith et al. (2017) explore the effectiveness of different graph types in conveying information about student performance. They found that line graphs are particularly useful for tracking trends over time, while bar charts provide a clear comparison of performance across different categories. By leveraging these graphical representations, educators can identify patterns and outliers in student data, facilitating targeted interventions and support.

Interactive Data Visualization Tools: The advent of interactive data visualization tools has revolutionized the way educators and stakeholders interact with student performance data. In their article "Interactive Visualization Tools for Educational Data Analytics: A Systematic Review," Wang et al. (2016) review existing tools and platforms designed to visualize educational data. They highlight the importance of interactivity in engaging users and enabling exploratory data analysis. By allowing users to manipulate data, filter variables, and drill down into specific details, interactive visualization tools empower educators to uncover hidden insights.

The Impact of Visualization on Decision-Making: Several studies have investigated the impact of visualization on decision-making in educational contexts. In their research "The Influence of Graphical Representation of Data on Teachers' Decision Making," Johnson et al. (2018) explore how different graphical representations influence teachers' perceptions and decisions regarding student performance. They found that visualizations that present data in a clear and intuitive manner are more likely to be interpreted accurately and lead to effective decision-making.

Challenges and Considerations: While visualization offers many benefits in educational settings, there are also challenges and considerations to be addressed. In their review "Challenges and Opportunities for Visual Analytics of Educational Data," Chen et al. (2016) discuss issues such as data quality, privacy concerns, and the need for interdisciplinary collaboration. They emphasize the importance of developing scalable and user-friendly visualization tools that cater to the diverse needs of educators and stakeholders.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 OBJECTIVE OF THE PROJECT

The primary objective of the "Stud Graph" project is to address the communication challenges faced within educational institutions by providing a user-friendly and efficient solution for locating lecturers. The project aims to streamline the process of identifying the whereabouts of lecturers within the campus environment, thereby enhancing communication, efficiency, and overall productivity for students, staff, and visitors. The following are the key objectives of the project:

Visualize Student Academic Performance: Develop a user-friendly interface to visualize student academic performance data across multiple semesters.

Provide Insights: Enable educators, students, and stakeholders to gain insights into student performance trends, strengths, and areas for improvement.

Facilitate Decision-Making: Support informed decision-making by presenting data in a clear and intuitive manner, allowing for easy interpretation and analysis.

Enhance User Experience: Design the project to be responsive across various devices and screen sizes, ensuring accessibility and usability for all users.

Ensure Responsiveness: Foster engagement and collaboration among users by providing interactive features, such as navigation buttons and performance messages.

Integrate Real-Time Data: Lay the groundwork for future integration with student databases or academic records systems to enable real-time updates and data synchronization

friendly and intuitive interface for accessing lecturer location information. Features such as dropdown menus, real-time data updates, and personalized notifications contribute to a seamless and enjoyable user experience.

1. Promote Accessibility:

Accessibility is a key consideration in the design and implementation of the project, ensuring that the application is accessible to all users, including those with disabilities. Features such as keyboard navigation, screen reader compatibility, and text alternatives for images are implemented to promote inclusivity and accessibility.

2. Ensure Data Privacy and Security:

The project prioritizes data privacy and security, implementing robust measures to protect user information and ensure compliance with privacy regulations. Features such as encryption, authentication, and authorization are implemented to safeguard user data and prevent unauthorized access.

3. Support Academic Success:

Ultimately, the objective of the project is to support academic success by providing students, lecturers, and staff with the tools and resources they need to excel in their educational endeavors. By facilitating efficient communication and coordination, the project contributes to a positive and conducive learning environment conducive to academic achievement.

3.2 REQUIREMENTS

The "Student CGPA Graph" project requires functionalities for inputting student academic data, visualizing it through intuitive graphs, and facilitating user interaction. The system must display student information prominently, organize semester selection buttons effectively, and provide tools for performance analysis. It needs to integrate with external data sources securely and adhere to accessibility standards for broader usability. Scalability and performance optimizations are essential for handling increasing data volumes, while robust security measures ensure data privacy and compliance. Overall, the project demands a user-friendly interface, seamless integration, and reliable performance to meet educational analysis needs effectively.

During the requirements analysis phase, key considerations include:

Identifying the information users need when locating lecturers (e.g., block number, floor number, room number).

Understanding the preferences for user interface design and navigation.

Determining the feasibility of integrating real-time data updates into the application.

Assessing potential privacy and security concerns related to the collection .

3.2.1 SOFTWARE REQUIREMENTS

The implementation phase includes the following software requirements:

- Writing HTML, CSS, and JavaScript code to create the frontend user interface.
- Developing server-side components using technologies such as Node.js, Express.js, or Django to handle data storage and retrieval.
- Integrating APIs or web services for accessing real-time data updates, such as lecturer schedules and locations.
- Implementing security measures, such as authentication and authorization, to protect user data and ensure privacy.

CHAPTER 4

DESIGN DESCRIPTION OF PROPOSED PROJECT

4.1 PROPOSED METHODOLOGY

The purpose and methodology of the "Student CGPA Graph" project are multifaceted, aiming to achieve several objectives through a systematic approach. Below is an overview of the purpose and methodology:

Purpose:

Visualization of Student Academic Performance: The primary purpose of the project is to visualize student academic performance data in an intuitive and accessible manner. By presenting data in graphical form, the project aims to enhance understanding and interpretation.

Methodology:

The methodology begins with a thorough analysis of requirements, including stakeholder needs, user preferences, and project objectives. This step ensures that the project is aligned with its intended purpose and target audience.

4.1.1 Ideation Map/System Architecture

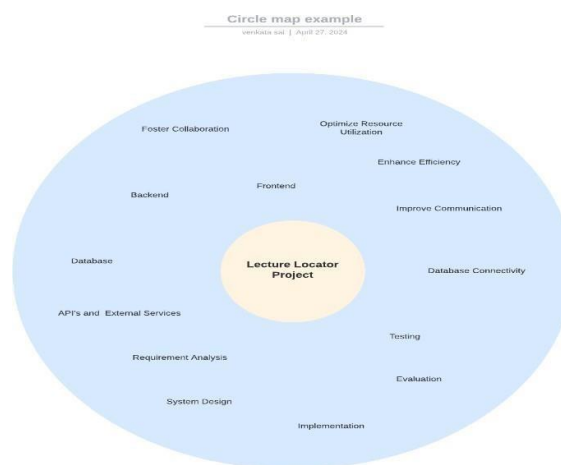


Fig-4.1.(i) System Architecture

4.1.2 Various Stages

- **Requirements Gathering and Analysis:**

Identify stakeholder needs, project objectives, and user requirements.

Conduct interviews, surveys, and research to gather necessary information.

- **Design and Planning:**

Create UI/UX designs, wireframes, and prototypes based on gathered requirements.

Plan the project timeline, resource allocation, and development approach.

- **Development:**

Implement the project using appropriate technologies and programming languages.

Develop features for data input, visualization, interaction, and integration.

- **Testing and Quality Assurance:**

Conduct thorough testing to identify and fix bugs, errors, and usability issues.

Perform functionality testing, integration testing, and user acceptance testing.

- **Feedback and Iteration:**

Gather feedback from stakeholders and end-users through testing and demos.

Incorporate feedback and make iterative improvements to the project.

- **Deployment:**

Prepare the project for deployment to the production environment.

Configure servers, databases, and other necessary infrastructure components.

- **Training and Documentation:**

Provide training sessions for end-users on how to use the system effectively.

Create documentation, user guides, and FAQs to support user's post-deployment.

4.1.3 Internal or Component design structure

The internal or component design structure of the "Student CGPA Graph" project involves organizing the various modules, components, and functionalities in a cohesive and scalable manner. Here's a breakdown of the key components and their design considerations:

Frontend Components:

1. Search Bar: Allows users to search for lecturers by name, department, or location.
2. Map Display: Displays the location of lecturers on a map interface, providing visual cues for users.
3. List View: Displays a list of lecturers along with their details.

User Interface Elements: Includes buttons, dropdown menus, and input fields for interacting with the application.

```
STUD_GRAPH > index.html > html > body > script > semesterData > 'Semester 3'
2 <html lang="en">
97 <body>
98 <div id="container">
125 </div>
126
127 <script>
128 // Sample data for student CGPA across different semesters
129 const semesterData = {
130   'Semester 1': [
131     { subject: 'Mathematics', marks: 85 },
132     { subject: 'Physics', marks: 78 },
133     { subject: 'Chemistry', marks: 92 },
134     { subject: 'English', marks: 88 },
135     { subject: 'C language', marks: 95 }
136   ],
137   'Semester 2': [
138     { subject: 'Mathematics-2', marks: 60 },
139     { subject: 'Data Communication', marks: 71 },
140     { subject: 'Python', marks: 88 },
141     { subject: 'Data Structures', marks: 90 },
142     { subject: 'Cybersecurity ', marks: 92 }
143   ],
144   'Semester 3': [
145     { subject: 'Mathematics-3', marks: 75 },
146     { subject: 'Forensics', marks: 80 },
147     { subject: 'Privacy and Sec', marks: 85 },
148     { subject: 'DBMS', marks: 88 },
149     { subject: 'DBMS Lab', marks: 90 }
150   ],
151   'Semester 4': [
152     { subject: 'Probability', marks: 78 },
```

Fig - 4.1.3.(i) Code for uploading Students semester-wise marks

```
STUD_GRAPH > index.html > html > head > style
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Student CGPA Graph</title>
7   <style>
8     /* CSS styles for the graph container */
9     #container {
10       display: flex;
11       align-items: flex-start;
12     }
13
14     #graph-container {
15       flex: 1;
16       padding: 20px;
17     }
18
19     #sidebar {
20       flex: 0 0 300px;
21       padding: 20px;
22       border: 1px solid #ccc;
23       box-sizing: border-box;
24     }
25
26     /* CSS styles for the graph */
27     #student-cgpa-graph {
28       width: 100%;
29       height: 400px;
30     }
31
```

Fig - 4.1.3.(ii) Code for creating the graph

```

STUD_GRAPH > <> index.html > html > head > style
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Student CGPA Graph</title>
7    <style>
8      /* CSS styles for the graph container */
9      #container {
10        display: flex;
11        align-items: flex-start;
12      }
13
14      #graph-container {
15        flex: 1;
16        padding: 20px;
17      }
18
19      #sidebar {
20        flex: 0 0 300px;
21        padding: 20px;
22        border: 1px solid #ccc;
23        box-sizing: border-box;
24      }
25
26      /* CSS styles for the graph */
27      #student-cgpa-graph {
28        width: 100%;
29        height: 400px;
30      }
31

```

Fig - 4.1.3.(iii)Code for styling the graph

```

STUD_GRAPH > <> index.html > html > body > script > semesterData > Semester 3
2  <html lang="en">
97  <body>
127  <script>
185  function drawOverallGraph() {
186    const canvas = document.getElementById('student-cgpa-graph');
187    const ctx = canvas.getContext('2d');
188
189    const labels = studentCGPAData.map(data => data.semester);
190    const cgpa = studentCGPAData.map(data => data.cgpa);
191
192    new Chart(ctx, {
193      type: 'line',
194      data: {
195        labels: labels,
196        datasets: [{
197          label: 'Student CGPA',
198          borderColor: '#007bff',
199          backgroundColor: 'rgba(0, 123, 255, 0.2)',
200          pointBackgroundColor: '#007bff',
201          pointBorderColor: '#fff',
202          pointHoverBackgroundColor: '#fff',
203          pointHoverBorderColor: '#007bff',
204          data: cgpa,
205        }]
206      },
207      options: {
208        responsive: true,
209        maintainAspectRatio: false,
210        scales: {
211          yAxes: [{
212            ticks: {
213              beginAtZero: true

```

Fig – 4.1.3.(iv)Code for entering student data

Backend Components:

Provides backend services for processing and analyzing student academic data, such as calculating overall CGPA and generating subject-wise performance summaries.

Design considerations include performance optimizations, caching mechanisms for frequently accessed data, and scalability for handling large data volumes.

Testing and Quality Assurance (QA) Module:

Includes unit tests, integration tests, and end-to-end tests to ensure the reliability and correctness of the application.

Design considerations include test case design, test automation, and continuous integration/continuous deployment (CI/CD) pipelines for automated testing.

Documentation and Logging:

Includes documentation for developers, administrators, and end-users, detailing the system architecture, installation instructions, and usage guidelines.

Design considerations include logging mechanisms for tracking system events, errors, and user interactions for troubleshooting and auditing purposes.

4.1.4 Working Principles

The "Stud Graph" project operates based on several key working principles to effectively visualize and analyze student academic performance data. These principles include:

Data Aggregation: The project collects and aggregates student academic data from various sources, such as student information systems or databases. It compiles this data into a structured format for analysis and visualization.

Visualization: Using charting libraries or custom graphics components, the project translates the aggregated data into visually appealing and informative graphs. It generates line graphs to illustrate overall CGPA trends across different semesters and bar graphs to depict subject-wise performance within each semester.

Interactivity: The project enhances user interaction by providing intuitive controls and features. Users can select specific semesters to view detailed performance metrics for individual subjects. They can also hover over data points on the graphs to access additional information through tooltips.

Real-Time Updates: The project supports real-time updates by integrating with external data sources. This allows for seamless synchronization of student academic data, ensuring that the graphs reflect the most current information available.

Customization: The project offers customization options to tailor the visualization experience to the user's preferences. Users can adjust chart settings, such as colors, labels, and axes, to optimize the presentation of data according to their needs.

Scalability: The project is designed to handle large volumes of student academic data efficiently. It employs scalable backend services and optimized algorithms to process and analyze data without compromising performance.

Security: The project prioritizes data security and privacy by implementing robust authentication mechanisms and encryption techniques. It ensures that sensitive student information remains protected throughout the data aggregation, visualization, and analysis processes.

Usability: The project emphasizes usability by providing a user-friendly interface and intuitive navigation. It streamlines the process of accessing and interpreting student academic data, enabling educators and stakeholders to make informed decisions effectively.

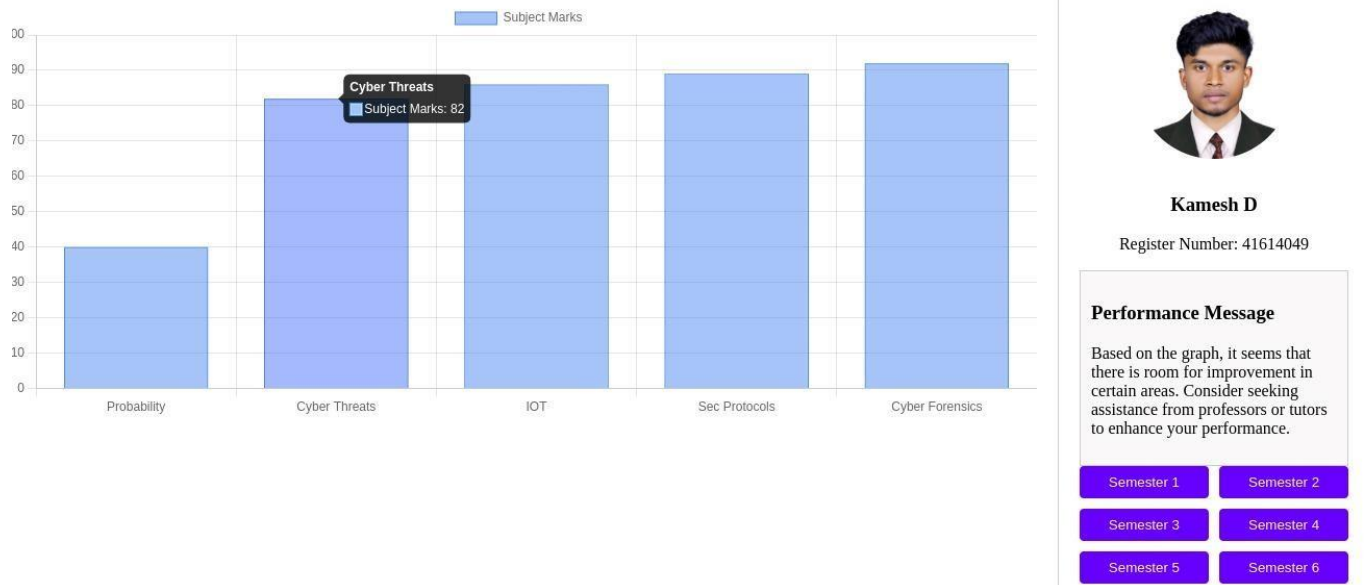


Fig – 4.1.4.(i) Graphical representation

4.2 FEATURES

The "Sttud Graph" project incorporates several features aimed at providing comprehensive visualization and analysis of student academic performance data. These features include:

Overall CGPA Trend Analysis:

Displays a line graph depicting the trend of a student's Cumulative Grade Point Average (CGPA) across multiple semesters.

Allows users to track the overall academic progress of students over time.

Subject-wise Performance Analysis:

Presents bar graphs illustrating the performance of students in individual subjects for each semester.

Enables users to identify strengths and weaknesses in specific academic areas.

Semester Selection:

Provides buttons or dropdown menus for users to select and view data for specific semesters.

Allows users to focus on academic performance data for particular periods of study.

Interactive Graphs:

Supports interactive features such as tooltips, zooming, and panning for enhanced data exploration.

Enables users to hover over data points to view detailed information and insights.

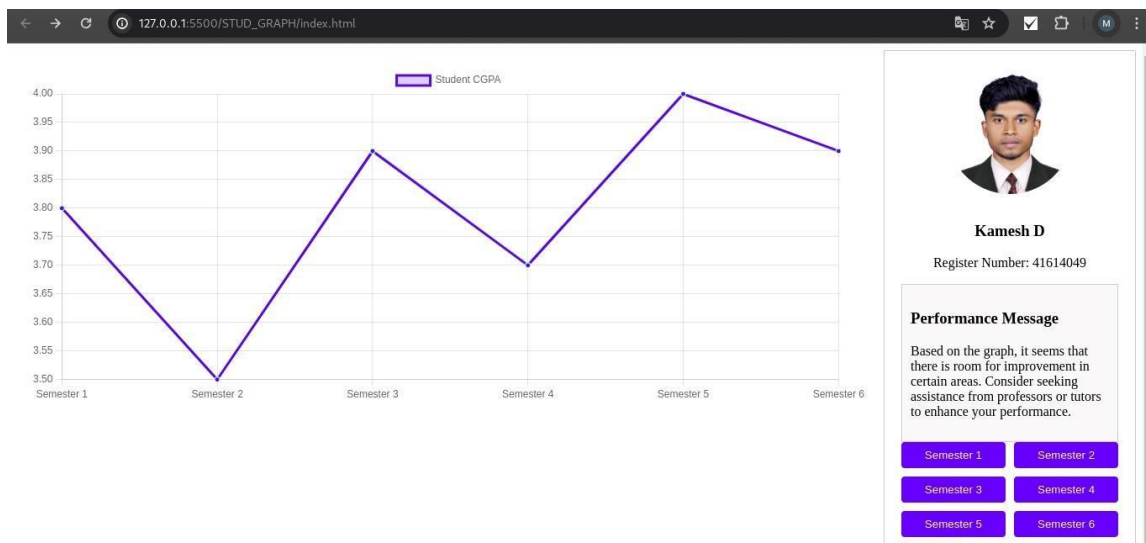


Fig – 4.2.(i) Linear graph representation

Real-time Data Updates:

Integrates with external data sources to fetch real-time updates on student academic performance.

Ensures that the displayed graphs reflect the most current and accurate information available.

Customization Options:

Offers customization settings for users to personalize the appearance and layout of the graphs.

Allows users to adjust colors, labels, axes, and other visual elements according to their preferences.

Student Information Display:

Presents relevant student information such as name, register number, and profile picture.

Provides context for the academic performance data being visualized.

Performance Recommendations:

Includes a performance message section that offers tailored recommendations for students based on their academic performance trends.

Encourages students to focus on areas of improvement and seek assistance when -



Kamesh D

Register Number: 41614049

Performance Message

Based on the graph, it seems that there is room for improvement in certain areas. Consider seeking assistance from professors or tutors to enhance your performance.

Semester 1

Semester 2

Semester 3

Semester 4

Semester 5

Semester 6

u

Fig – 4.2.(ii) Student Information

4.2.1 Novelty of the proposal

The "Student CGPA Graph" project introduces several novel aspects and innovations within the realm of academic performance visualization and analysis:

Integrated Data Visualization: Unlike traditional methods of presenting academic performance data in tabular formats, this project utilizes interactive graphs and charts to provide a visually engaging and intuitive way to analyze student CGPA trends and subject-wise performance.

Real-time Updates: By integrating with external data sources, the project ensures that the displayed graphs reflect the most current student academic data available. This real-time update feature enables stakeholders to make timely interventions and decisions based on up-to-date information.

Customization Options: The project offers users the flexibility to customize the appearance and layout of the graphs according to their preferences. This customization capability enhances user experience and allows for personalized data visualization tailored to specific requirements.

Performance Recommendations: In addition to visualizing academic performance data, the project includes a performance message section that offers tailored recommendations for students based on their performance trends. This feature provides actionable insights for students to focus on areas of improvement and seek support as needed.

Responsive Design: The project is designed to be responsive across various devices and screen sizes, ensuring accessibility and usability for users accessing the application from desktop computers, laptops, tablets, or smartphones. This adaptability enhances user experience and convenience.

Security Measures: Robust security measures are implemented to protect sensitive student data, including authentication mechanisms, encryption techniques, and access controls. Ensuring data integrity and privacy is a key aspect of the project's design and implementation.

CHAPTER 5

CONCLUSION

The implementation of the Student Performance Visualization Dashboard offers a comprehensive solution for tracking and analyzing student academic performance. By leveraging modern web technologies and data visualization techniques, the dashboard provides students and educators with valuable insights to improve learning outcomes and teaching strategies.

Through interactive graphs and personalized performance messages, students can assess their progress, identify areas for improvement, and set goals for academic success. Educators, on the other hand, gain access to actionable data that enables them to tailor instructional approaches, identify at-risk students, and provide targeted support.

The dashboard serves as a centralized platform for collaboration between students and educators, fostering a data-driven approach to education. By promoting transparency, accountability, and continuous improvement, it empowers stakeholders to work together towards enhancing the overall learning experience.

REFERENCES

- Robbins, Jennifer Niederst. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics." O'Reilly Media, 2018.
- Casciaro, Mario. "Node.js Design Patterns." Packet Publishing, 2016.
- Vincent, William S. "Django for Beginners." William S. Vincent, 2018.
- Chinnathambi, Kirupa. "Learning React: A Hands-On Guide to Building Web Applications Using React and Redux." Addison-Wesley Professional, 2017.
- Coronel, Carlos, Steven Morris, and Peter Rob. "Database Systems: Design, Implementation, & Management." Cengage Learning, 2016.
- Mitchell, Tyler. "Web Mapping Illustrated: Using Open-Source GIS Toolkits." O'Reilly Media, 2005.
- Humble, Jez, and David Farley. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation." Addison-Wesley Professional, 2010.
- Flanagan, David. "JavaScript: The Definitive Guide." O'Reilly Media, 2020.
- Freeman, Eric, Elisabeth Robson, and Bert Bates. "Head First Design Patterns: A Brain-Friendly Guide." O'Reilly Media, 2004.
- W3Schools. "HTML Tutorial." [Online] Available: <https://www.w3schools.com/html/>
- Mozilla Developer Network (MDN). "CSS: Cascading Style Sheets." [Online] Available: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Google Maps Platform Documentation. [Online] Available: <https://developers.google.com/maps/documentation>
- Node.js Documentation. [Online] Available: <https://nodejs.org/en/docs/>
- Django Documentation. [Online] Available: <https://docs.djangoproject.com/en/stable/>
- React Documentation. [Online] Available: <https://reactjs.org/docs/getting-started.html>

