VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018, Karnataka

Project Report
on

"A Privacy Protected and Federated Adaptive EdTech Learning Platform"

Submitted in partial fulfillment of the requirements for the award of the degree of
Bachelor of Engineering
in
Artificial Intelligence and Machine Learning

Submitted by

USN	Name
1BI20AI022	Kollipara Sai Sandeep
1BI20AI046	Shreyas R S
1BI20AI048	Somula Jaswanth Reddy
1BI20AI056	Vivek Pandith D V

Under the Guidance of
Dr. D G Jyothi
Professor and Head
Department of AI&ML, BIT
Bengaluru-560004

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
BANGALORE INSTITUTE OF TECHNOLOGY
K.R. Road, V.V. Pura, Bengaluru-560 004

2023-24

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY

Bengaluru-560 004

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

Certified that the project work entitled **"A Privacy Protected and Federated Adaptive EdTech Learning Platform"** carried out by

USN	Name
1BI20AI022	**Kollipara Sai Sandeep**
1BI20AI046	**Shreyas R S**
1BI20AI048	**Somula Jaswanth Reddy**
1BI20AI056	**Vivek Pandith D V**

a bonafide students of VIII semester in partial fulfillment for the award of Bachelor of Engineering in Artificial Intelligence & Machine Learning of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi** during the academic year 2023-24. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

Name & Signature of the Guide Name & Signature of the HOD Signature of the Principal

External Viva

Name of the examiners & signature with date

1

2.

ABSTRACT

The goal of providing individualised learning experiences in the field of educational technology is still to protect user privacy. This study proposes a novel solution to this problem by creating a course recommendation system based on Federated Learning. platform, which makes use of Federated Learning, examines user interactions locally on individual devices while protecting data privacy. It then compiles insights globally to produce customised course recommendations. This approach offers consumers personalised learning paths while protecting the privacy of their sensitive data by fusing the strength of machine learning with resilient privacy-preserving strategies. These results highlight how crucial it is for educational technology to strike a balance between privacy and personalisation. By adopting Federated Learning, we open the door to a more inclusive and equitable learning environment while also giving users the confidence to achieve their educational objectives.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without complementing those who made it possible and whose guidance and encouragement made our efforts successful. So, my sincere thanks to all those who have supported us in completing this Project successfully.

Our sincere thanks to **Dr. M.U. Aswath**, Principal, BIT and **Dr. D G Jyothi**, Head of the Department of Artificial Intelligence and Machine Learning, BIT for their encouragement, support and guidance to the student community in all fields of education. We are grateful to our institution for providing us a congenial atmosphere to carry out the Project successfully.

We would not forget to remember **Dr. D G Jyothi** and **Prof. Shruthiba A**, our Project Coordinators, for their encouragement and more over for his timely support and guidance till the completion of Project.

We avail this opportunity to express our profound sense of deep gratitude to my esteemed guide **Dr. D G Jyothi**, for their moral support, encouragement and valuable suggestions throughout the Project.

We wish to express our heartfelt gratitude to our mentor **Mr. Sheshadri K R** for their valuable guidance, suggestions and cheerful encouragement during the period of our Project.

We extend our sincere thanks to my department faculty members of Artificial Intelligence and Machine Learning and also non-teaching staff for supporting me directly or indirectly for the completion of this Project.

Kollipara Sai Sandeep (1BI20AI022)
Shreyas R S (1BI20AI046)
Somula Jaswanth Reddy (1BI20AI048)
Vivek Pandith D V (1BI20AI056)

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION **1-5**

1.1 Overview 1

1.2 Objectives 2

1.3 Purpose , Scope and Applicability 3

1.4 Organization of Report 4

1.5 Purpose, Scope and Applicability 3

CHAPTER 2: LITERATURE SURVEY **6-15**

2.1 Introduction 6

2.2 Summary of papers 7

2.3 Drawbacks of Existing System 14

2.4 Problem statement 15

2.5 Proposed system 15

CHAPTER 3: REQUIREMENT ENGINEERING **16-29**

3.1 Software and Hardware Tools used 16

3.1.1 Software Tools 16

3.1.2 Hardware Tools 18

3.2 Conceptual/Analysis Modelling 19

3.2.1 Use Case Diagram 19

3.2.3 Activity Diagram 21

3.3 Software Requirement Specification 24

3.3.1 Functional Requirements 24

3.3.2 Non-Functional Requirements 26

3.3.3 Domain Requirements 27

CHAPTER 4: PROJECT PLANNING **30-31**

4.1 Project Planning and Scheduling 30

CHAPTER 5: SYSTEM DESIGN **32-41**

5.1 System Architecture 32

5.2 Module Decomposition 33

5.3 Interface Design 35

5.4 Data Structure Design 36

5.5 Algorithm Design 38

CHAPTER 6: IMPLEMENTATION **42-46**

6.1 Implementation Approaches 42

6.1.2 Machine Learning Model 42

6.1.1 Encryption for edge prediction 43

6.2 Coding Details 39

6.2.1 ML Model 39

6.2.2 Homomorphic Encryption 45

6.2.3 Aggregation and recommendation 45

CHAPTER 7: TESTING **47-50**

7.1 Testing Approach 47

7.1.1 Unit Testing 47

CHAPTER 8: RESULTS DISCUSSION AND PERFORMANCE ANALYSIS **51-50**

8.1 Recommendation Engine 52

8.1.1 Support Vector Machine 52

8.1.2 Linear Regression 53

8.2 User Documentation : LearnSync 56

8.3 Snapshots 58

CHAPTER 9: CONCLUSION, APPLICATIONS AND FUTURE WORK **65**

9.1 Conclusion 65

9.2 Applications 66

9.3 Limitation and Future Scope of Work 68

REFERENCES

LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	Use case diagram	19
3.2	Activity diagram	21
4.1	Project planning Gantt Chart	30
5.1	System Architecture	32
5.2	Module Decomposition Diagram	34
5.3	Interface Design Diagram	35
5.4	Algorithm Working Explanation	39
5.5	CKKS Homomorphic Encryption Algorithm	40
6.1	Random Forest algorithm working	42
8.1	Training and Validation Accuracy Graph	55
8.2	Landing Page of LearnSync	58
8.3	Features Section of landing page	58
8.4	Login Page of LearnSync	59
8.5	Home Page of LearnSync	60
8.6	Stats Page of LearnSync	61
8.7	Test Page of LearnSync	62
8.8	Account Page of LearnSync	63
8.9	Schedule Page of LearnSync	63
8.10	White Theme	64

LIST OF TABLES

Table No.	Title	Page No.
7.1	Unit Testing for Prediction Model	47
7.2	Unit Testing for Homomorphic Encryption	49
7.3	Unit Testing for User Interface	50
8.1	SVM Classifier Average accuracy	53
8.2	Gradient Boosting Classifier Average accuracy	54

CHAPTER 1
INTRODUCTION

Chapter - 1

INTRODUCTION

1.1 Overview

In an age marked by the prolific generation and consumption of data, coupled with rapid technological advancements, the realm of machine learning stands at the forefront of innovation. Within this landscape, Federated Learning (FedML) emerges as a groundbreaking paradigm, redefining the traditional model of centralized data processing by distributing model training across decentralized edge devices. FedML embodies the fundamental principles of privacy preservation, facilitating collaborative learning without compromising the confidentiality of sensitive user data. This federated learning-based educational platform harnesses the transformative potential of FedML to reimagine personalized education, providing learners with tailored learning experiences while upholding their privacy rights.

At the nucleus of this platform lies a meticulously crafted user interface powered by React.js, designed to deliver a seamless and intuitive experience. From the moment users land on the platform's homepage, they are greeted with a wealth of information about the project and its team, setting the stage for an immersive learning journey. The platform's login and signup pages offer users a gateway to a diverse array of courses tailored to their interests and educational goals, fostering a sense of empowerment and agency in their learning pursuits. With its user-friendly interface and streamlined navigation, the platform serves as a gateway to a world of knowledge and discovery.

Central to the platform's architecture is a sophisticated federated learning system that integrates robust privacy-preserving techniques at its core. By decentralizing model training and leveraging local datasets stored on edge devices, the system minimizes the need for centralized data processing, thereby mitigating privacy and security risks. This distributed approach not only enhances data privacy but also fosters a collaborative learning environment where users can contribute to model training without divulging sensitive information. Through this innovative approach, the platform aims to establish a secure infrastructure for collecting, storing, and processing student data, ensuring data integrity and confidentiality at every step.

Furthermore, the platform leverages the power of federated learning to adapt educational content recommendations and assessments to the unique needs and preferences of individual learners. By analysing performance metrics and user engagement patterns, the platform delivers personalized course recommendations and assessments tailored to each user's proficiency level and learning style. This adaptive learning approach not only enhances user engagement but also fosters a deeper understanding of complex concepts and topics. With its ability to dynamically adjust to the evolving needs of users, the platform represents a significant leap forward in the realm of educational technology, promising to revolutionize the way we learn and grow in the digital age.

1.2 Objectives

1. **Implement Robust Privacy-Preserving Techniques:** Develop and integrate advanced privacy-preserving techniques within the platform's framework to address the inherent privacy challenges associated with Federated Learning. This includes leveraging techniques such as homomorphic encryption and differential privacy to ensure the confidentiality of user data during model training and aggregation processes.
2. **Distributed Model Training Across Edge Devices:** Enable distributed model training across decentralized edge devices to minimize the reliance on centralized data processing, thereby reducing privacy and security risks associated with data sharing. This objective aims to empower users to participate in model training without compromising the privacy of their personal data.
3. **Build a Robust and Secure Infrastructure:** Establish a robust and secure infrastructure for collecting, storing, and processing student data, ensuring data integrity and confidentiality throughout the platform. This involves implementing stringent security measures and encryption protocols to safeguard sensitive user information from unauthorized access or breaches.
4. **Utilize Federated Learning for Personalization:** Harness the power of Federated Learning to deliver personalized educational content recommendations and assessments tailored to individual students' abilities and preferences. By analyzing performance metrics and user engagement patterns, the platform aims to provide customized learning experiences that optimize student engagement and learning outcomes.

5. **Create a User-Friendly Interface:** Design and develop a user-friendly interface that simplifies interaction with the federated learning platform, enhancing accessibility and usability for users of all levels. This objective focuses on creating intuitive navigation, visually appealing layouts, and responsive design elements to ensure a seamless and engaging user experience.

1.3 Purpose , Scope and Applicability

The purpose of this federated learning-based educational platform is multifaceted, aiming to revolutionize traditional learning paradigms by leveraging cutting-edge technologies to enhance personalized learning experiences while safeguarding user privacy. At its core, the platform seeks to address the growing concerns surrounding data privacy and security in educational settings while catering to the diverse learning needs and preferences of individual users.

The scope of the platform extends beyond conventional learning management systems, encompassing a holistic approach to educational technology that integrates federated learning, homomorphic encryption, and user-centric design principles. By adopting a federated learning approach, the platform enables distributed model training across edge devices, minimizing the need for centralized data processing and mitigating privacy risks associated with data sharing. This distributed model training paradigm not only enhances data privacy but also facilitates adaptive learning experiences tailored to individual users' performance and preferences.

In terms of applicability, the platform offers a versatile solution that can be deployed across various educational contexts, including K-12 education, higher education, corporate training, and lifelong learning initiatives. By harnessing federated learning techniques, the platform can adapt to diverse learning environments and user demographics, making it suitable for both traditional classroom settings and remote learning scenarios. Furthermore, the platform's user-friendly interface and personalized recommendations cater to users with varying levels of technical proficiency, ensuring accessibility and inclusivity for all learners.

The platform's applicability extends beyond educational institutions to include corporate training programs, professional development initiatives, and lifelong learning platforms. By leveraging federated learning techniques, organizations can harness the collective

intelligence of their workforce or user base to deliver targeted training programs and personalized learning experiences that align with individual goals and organizational objectives. Additionally, the platform's emphasis on data privacy and security makes it suitable for industries with stringent regulatory requirements, such as healthcare, finance, and government, where confidentiality and data protection are paramount.

Overall, the purpose, scope, and applicability of this federated learning-based educational platform underscore its potential to transform traditional learning environments, empower learners, and facilitate lifelong learning journeys. By embracing innovative technologies and user-centric design principles, the platform offers a scalable and adaptable solution that meets the evolving needs of today's learners and educational stakeholders.

1.4 Organization of Report

The subsequent chapters of the project report will delve into various aspects of the implementation of our federated learning-based educational platform. In Chapter 2, an extensive literature survey will be conducted to explore existing research and technologies in the fields of federated learning, educational technology, and data privacy. This chapter will provide valuable insights into the current state of the art, identify gaps in existing literature, and lay the groundwork for our proposed methodology.

Chapter 3 will focus on requirement engineering, outlining the specific needs and objectives of our project. By eliciting and analyzing user requirements, system functionalities, and design constraints, this chapter will serve as a roadmap for the subsequent stages of development. It will detail the key features, functionalities, and performance metrics that our federated learning-based educational platform aims to achieve.

In Chapter 4, the project planning process will be elucidated, including timelines, resource allocation, and risk management strategies. This chapter will outline the project milestones, deliverables, and dependencies, ensuring a structured and organized approach to implementation. By delineating clear goals and objectives, we aim to streamline the development process and mitigate potential challenges.

Chapter 5 will delve into the system design, providing an architectural overview of our federated learning-based educational platform. This chapter will detail the components, modules, and interactions within the system, elucidating the underlying infrastructure and

technology stack. By presenting a comprehensive system design, we aim to facilitate a deeper understanding of the platform's inner workings and functionalities.

The implementation process will be the focus of Chapter 6, where the technical details of developing and deploying our federated learning-based educational platform will be discussed. This chapter will cover software development methodologies, coding practices, and integration strategies employed during the implementation phase. By documenting the implementation process, we aim to provide insights into the practical challenges and solutions encountered during development.

Chapter 7 will be dedicated to testing, where the procedures and methodologies for validating the functionality and performance of our federated learning-based educational platform will be outlined. This chapter will detail the testing frameworks, test cases, and evaluation criteria used to assess the platform's reliability, scalability, and usability. By conducting rigorous testing, we aim to ensure the robustness and quality of our platform.

In Chapter 8, the results of our implementation will be discussed, and a comprehensive performance analysis will be conducted. This chapter will present empirical data, metrics, and insights gathered during testing, allowing for an objective evaluation of the platform's effectiveness and efficiency. By analyzing the results, we aim to identify strengths, weaknesses, and areas for improvement.

Finally, Chapter 9 will conclude the report by summarizing the findings, discussing the implications of our work, and outlining directions for future research and development. This chapter will highlight the significance of our federated learning-based educational platform, its potential applications, and the opportunities for further enhancement and refinement. By offering concluding remarks and recommendations, we aim to provide a comprehensive overview of our project's contributions and significance.

The report will be supplemented with a references section where all cited sources and relevant literature will be listed for further reading and academic purposes. This organizational structure aims to provide a comprehensive and systematic exploration of our federated learning-based educational platform, from conceptualization to implementation and beyond.

CHAPTER 2
LITERATURE SURVEY

Chapter – 2

LITERATURE SURVEY

2.1 Introduction

A literature survey in a project report is that section that shows the various analyses and research made in the field of your interest and the results already published taking into account the various parameters of the project and the extent of the project. A Literature survey refers to getting the content from the books that are related to the topic or a given project. It should be referred from some research paper that is related to the topic. Any materials that are related to the project from the internet which are valuable for the student and have helped the student to enhance the report status as well as the calculation, analysis, and tabulation majorly reflected in the survey. So, in this way, one can select the literature survey. It is necessary to emphasize that it is the most important part of the project report. It is the most important part of the report as it gives the students direction in the area of their research. It helps the students to set a goal for analysis - thus giving them their problem statement. When one writes a literature review in respect of the project, they have to write the research made by various analysts - their methodology (which is their abstract) and the conclusions they have arrived at. One should also give an account of how this research has influenced their thesis.

Literature surveys are needed for:

- To see what has and has not been investigated.
- To identify data sources that other researchers have used.
- To learn how others have defined and measured key concepts.
- To develop alternative research projects.
- To put one's perspective into work.
- To contribute to the field by moving research forward.
- Reviewing the literature lets one see what came before, and what did and didn't work for other researchers.
- To demonstrate one's understanding, and ability to critically evaluate research in the field.
- To provide evidence that may be used to support your own findings.

2.2 Summary of papers

1.Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao - "Insights into Federated Learning: Data Partitioning, Privacy Mechanisms, and Model Exchange" (2021)

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao provided a comprehensive review of federated learning in their 2021 paper. The study delves into various aspects of federated learning, including data partitioning, privacy mechanisms, and model exchange between local devices and a central aggregator. By analyzing existing research and methodologies, the authors offer insights into the potential of federated learning to address challenges related to data privacy and scalability in distributed machine learning systems. However, despite its promises, the paper also highlights several limitations of federated learning, such as the complexity of managing heterogeneous data sources, the need for robust privacy-preserving techniques, and the overhead associated with model aggregation and synchronization across distributed devices.

In their review of federated learning, Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao underscore the significance of advancements in data partitioning strategies, privacy-preserving mechanisms, and efficient model exchange protocols. The paper emphasizes the importance of balancing model performance with privacy guarantees, highlighting the need for innovative approaches to mitigate information leakage and model poisoning attacks in federated learning settings. Despite its potential, federated learning still faces challenges in achieving optimal convergence and model generalization across distributed devices with varying computational capabilities and data distributions. [1]

2.Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith - "Navigating Federated Learning Challenges: Methods, Trade-offs, and Future Directions" (2020)

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith explored the landscape of federated learning in their 2020 paper, delving into its challenges, methods, and future directions. Federated learning, renowned for its ability to preserve privacy by training models locally and sharing only updates, has emerged as a promising approach in distributed machine learning settings. However, despite its potential, federated learning encounters various challenges that impede its widespread adoption and efficacy.

One significant challenge highlighted by the authors is the costly communication overhead incurred during the model aggregation process. As federated learning involves transmitting

model updates from multiple devices to a central server for aggregation, the communication costs can become prohibitive, particularly in scenarios with large-scale datasets or low-bandwidth network environments. Moreover, federated learning often faces a delicate balance between preserving privacy and maintaining model performance. While privacy-preserving techniques such as differential privacy and secure aggregation offer mitigation strategies, they may come at the expense of model accuracy and convergence speed. Thus, navigating these trade-offs remains a key research focus in federated learning.[2]

3. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., and Ludwig- "Advancing Privacy in Federated Learning: A Hybrid Approach Integrating Differential Privacy and Secure Multiparty Computation :.2022

Truex, S., Baracaldo, N., Anwar, A., Steinke, T., and Ludwig introduced a novel hybrid approach to privacy-preserving federated learning in their 2019 paper. Their proposed method integrates Differential Privacy and Secure Multiparty Computation to bolster model accuracy while ensuring robust privacy guarantees. By leveraging these techniques, the approach aims to mitigate extraction attacks and collusion threats, which are prevalent in federated learning settings where data privacy is a primary concern.

The hybrid approach presented by Truex et al. offers a promising solution to the inherent privacy challenges of federated learning. By combining Differential Privacy and Secure Multiparty Computation, the method provides a robust framework for preserving data privacy while enabling collaborative model training across distributed devices. Moreover, the approach enhances model accuracy by incorporating privacy-enhancing techniques directly into the federated learning process. However, the effectiveness and scalability of this approach may depend on various factors such as the complexity of the privacy-preserving mechanisms, the computational overhead involved, and the adaptability to different federated learning scenarios. Further research and experimentation are needed to assess the practical implications of this hybrid approach in real-world settings.[3]

4.Weizhao Jin, Yuhang Yao, Shanshan Han, Carlee Joe-Wong, Srivatsan Ravi, Salman Avestimehr, Chaoyang He. "FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System". 2022.

This paper proposes FedML-HE, a novel federated learning system that leverages homomorphic encryption (HE) techniques to ensure privacy preservation while facilitating collaborative model training across decentralized edge devices. FedML-HE addresses the

challenge of protecting sensitive data during the federated learning process, making it a valuable contribution to the field.

However, one potential limitation of this approach could be the computational overhead associated with homomorphic encryption, which may impact the scalability and performance of the system, particularly as the user base grows. Additionally, ensuring the efficiency and effectiveness of the encryption scheme in real-world scenarios may require further optimization and experimentation.[4]

5.Jaehyoung Park and Hyuk Lim. "Adoption of quality edtech products in India: a case study of government implementation towards a sustainable edtech ecosystem". 2022.

This paper investigates the adoption of quality educational technology (edtech) products in India, focusing on government initiatives aimed at fostering a sustainable edtech ecosystem. By analyzing case studies and implementation strategies, the study offers insights into the challenges and opportunities for integrating edtech solutions into the Indian education system.

However, a limitation of this study may be its focus on a specific geographic region (India), which may limit the generalizability of the findings to other contexts. Additionally, the effectiveness of government interventions in promoting the adoption of edtech products may vary depending on local infrastructural, cultural, and socio-economic factors.[5]

6.Rezak Aziz, Soumya Banerjee, Samia Bouzefrane, and Thinh Le Vinh. "Exploring Homomorphic Encryption and Differential Privacy Techniques towards Secure Federated Learning Paradigm". 2023.

This paper explores the use of homomorphic encryption and differential privacy techniques to enhance the security of federated learning systems. By incorporating these privacy-preserving mechanisms, the proposed framework aims to mitigate privacy risks associated with sharing sensitive data across distributed edge devices.

However, a potential limitation of this approach may be the computational overhead introduced by homomorphic encryption and differential privacy techniques, which could impact the efficiency and scalability of federated learning systems, particularly in resource-constrained environments. Additionally, ensuring the compatibility and interoperability of different privacy-preserving techniques may pose challenges in real-world deployment scenarios.[6]

7.QIONG WU, KAIWEN HE, AND XU CHEN. "Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework". 2020.

This paper presents a cloud-edge based framework for personalized federated learning in intelligent Internet of Things (IoT) applications. By leveraging edge computing resources and collaborative model training techniques, the proposed framework enables personalized model customization while preserving data privacy and security in IoT environments.

However, a limitation of this framework may be its reliance on centralized cloud infrastructure for model aggregation and coordination, which may introduce latency and scalability challenges, particularly in large-scale IoT deployments. Additionally, ensuring the reliability and robustness of federated learning models in dynamic IoT environments with heterogeneous devices and data sources may require further investigation and optimization.[7]

8. Shiqiang Wang, Tiffany Tuor. "Adaptive Federated Learning in Resource Constrained Edge Computing Systems". 2019.

This paper explores adaptive federated learning techniques tailored for resource-constrained edge computing systems. By dynamically adjusting model parameters and communication strategies based on local device capabilities and network conditions, the proposed approach aims to improve the efficiency and effectiveness of federated learning in edge computing environments.

However, a potential limitation of this approach may be the complexity of adaptive model optimization algorithms and communication protocols, which may require additional computational resources and overhead on edge devices. Additionally, ensuring the compatibility and interoperability of adaptive federated learning techniques across diverse edge computing platforms and architectures may pose challenges in real-world deployment scenarios.[8]

9.Xiaoyuan Liu, Hongwei Li, Guowen Xu, Rongxing Lu, Miao He. "Adaptive privacy-preserving federated learning". 2019.

This paper proposes adaptive privacy-preserving federated learning techniques that dynamically adjust privacy protection mechanisms based on the sensitivity of data and user preferences. By tailoring privacy guarantees to individual user requirements, the proposed

approach aims to enhance the flexibility and usability of federated learning systems while ensuring robust privacy protection.

However, a limitation of this approach may be the complexity and computational overhead associated with adaptive privacy-preserving mechanisms, which may impact the scalability and performance of federated learning systems, particularly in resource-constrained environments. Additionally, ensuring the effectiveness and reliability of adaptive privacy-preserving techniques across diverse user populations and data distributions may require further validation and experimentation.[9]

10. Alaa Zuhir Al Rawashdeh, Enaam Youssef Mohammed. "Advantages and Disadvantages of Using e-Learning in University Education: Analyzing Students Perspectives". 2021.

This paper examines the advantages and disadvantages of e-learning in university education from the perspective of students. By analyzing survey data and qualitative feedback, the study provides insights into the benefits and challenges of e-learning adoption in higher education settings.

However, a limitation of this study may be its reliance on self-reported data and subjective perceptions, which may introduce bias and inaccuracies in the analysis of e-learning advantages and disadvantages. Additionally, the generalizability of the findings may be limited by the specific context and characteristics of the surveyed student population, which may not be representative of broader demographic groups or institutional settings.[10]

11. Mitali Sharad Gupta, Mr. Pratik Warkhedkar. "A study of the impact of edtech companies on education with special reference to Byjus and Vedantu". 2023.

This paper investigates the impact of educational technology (edtech) companies, with a focus on prominent platforms such as Byjus and Vedantu, on the education sector. By analyzing user adoption trends, market dynamics, and educational outcomes, the study aims to assess the effectiveness and implications of edtech platforms in transforming traditional educational practices.

However, a limitation of this study may be its reliance on secondary data sources and market analysis, which may not provide comprehensive insights into the nuanced impacts of edtech platforms on teaching and learning outcomes. Additionally, the study's focus on specific edtech companies may limit the generalizability of the findings to broader trends

and phenomena in the education sector, requiring further research and validation from diverse perspectives and contexts.[11]

12.Alexander Brecko, Erik Kajati, Jiri Koziorek, and Iveta Zolotova examined the application of Federated Learning techniques tailored for edge computing environments in their 2022 paper.

The study delves into the intricacies of leveraging Federated Learning on edge devices, shedding light on prevalent frameworks designed to facilitate client-server communication in decentralized learning setups. However, amidst the promising prospects, the paper also identifies significant challenges, particularly in addressing computational disparities and constraints inherent to diverse edge devices. These challenges necessitate innovative solutions to adapt Federated Learning algorithms to the heterogeneous nature of edge computing environments, ensuring efficient model training and optimization across a spectrum of edge devices with varying computational resources.

The paper underscores the need for specialized adaptation mechanisms and optimization strategies to accommodate the diverse computational capabilities and resource constraints prevalent in edge environments. Additionally, the authors highlight the significance of developing lightweight communication protocols and model aggregation techniques tailored for edge devices, aiming to minimize overhead while maximizing model accuracy and convergence. Moving forward, the research community is urged to focus on addressing these challenges through interdisciplinary collaborations and innovative algorithmic advancements.[12]

13.Jaehun Song, Min-Hwan Oh, and Hyung-Sin Kim - "Enhancing Federated Learning: FedSIM's Personalized Approach with Server-Side Information" (2022)

Jaehun Song, Min-Hwan Oh, and Hyung-Sin Kim introduced a groundbreaking concept in their 2022 paper, presenting FedSIM as an innovative solution in federated learning. FedSIM stands out for its incorporation of server-side information, which enhances the personalization aspect of federated learning models. By allowing servers to participate in the learning process, FedSIM aims to optimize model performance by leveraging additional insights and resources. However, despite its promising approach, FedSIM encounters certain drawbacks that warrant attention and further research. One notable limitation is its increased dependency on the server, which introduces potential single points of failure and raises concerns about system robustness and reliability.

While FedSIM offers significant advancements in personalized federated learning, its reliance on server-side information poses challenges related to scalability and resource management. The paper highlights the need for addressing these drawbacks to ensure the practical viability and effectiveness of FedSIM in real-world deployment scenarios. Additionally, the increased resource demands associated with server participation raise concerns about the computational overhead and infrastructure requirements, which may hinder the scalability and accessibility of the proposed approach. These limitations underscore the importance of ongoing research efforts aimed at refining FedSIM.[13]

14.Bhattacharya, Leena, and Nandakumar, Minu investigated a novel approach to privacy-preserving federated learning in their 2023 paper titled "A Hybrid Approach to Privacy-Preserving Federated Learning."

The study addresses the growing concerns surrounding data privacy in federated learning systems, proposing a hybrid methodology that combines cryptographic techniques with differential privacy mechanisms to safeguard sensitive information while enabling collaborative model training across distributed devices. By leveraging this hybrid approach, the researchers aim to enhance the privacy guarantees of federated learning frameworks, thereby promoting the adoption of decentralized machine learning solutions in various domains, including education technology (EdTech).

The authors highlight the pressing need for standardized quality evaluation methods in the large-scale adoption of EdTech platforms. In their paper, Bhattacharya, Leena, and Nandakumar, Minu argue that a lack of standardized assessment criteria can hinder the effectiveness and scalability of EdTech solutions, leading to disparities in educational outcomes and user experiences. By advocating for the development of objective evaluation frameworks and metrics tailored to different learning environments and user demographics, the researchers aim to address these challenges and promote the adoption of evidence-based practices in educational technology research and development.[14]

15. Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and Yike Guo - "Navigating Privacy Preservation in Federated Learning: A Survey of Techniques and Considerations" (2021)

Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and Yike Guo's paper titled "Privacy Preservation in Federated Learning" (2021) provides a detailed exploration of privacy-preserving techniques within the context of federated learning. The study aims to

mitigate data privacy and security concerns inherent in AI systems by conducting a survey of various methodologies and strategies employed to safeguard sensitive information during the federated learning process. By analyzing the effectiveness and applicability of different privacy preservation approaches, the authors offer valuable insights into the evolving landscape of privacy-enhancing technologies in the realm of distributed machine learning.

In their investigation of privacy preservation in federated learning, Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and Yike Guo shed light on the importance of balancing privacy protection with model performance and utility. The paper delves into various privacy-enhancing techniques such as differential privacy, secure aggregation, and federated learning with cryptographic protocols, highlighting their strengths and limitations in mitigating privacy risks. Nevertheless, the study may face challenges in providing comprehensive guidance on navigating the complex landscape of legal and regulatory requirements governing data privacy and security.

2.3. Drawbacks of Existing System

- **Inflexible Learning Environment:** Traditional classroom-based learning tends to follow a rigid structure that may not accommodate the diverse learning styles and paces of students. The fixed curriculum and instructional methods might not cater to individual preferences or adapt to varying levels of understanding among students.
- **Limited Personalization:** One of the key drawbacks of traditional learning environments is the lack of personalization. Teachers often face challenges in providing tailored instruction to meet the specific needs and abilities of each student. This can result in some students feeling left behind or not sufficiently challenged, while others may struggle to keep pace with the rest of the class.
- **Homogeneity in Content Delivery:** Traditional teaching methods often rely on a standardized curriculum and delivery format, which may not effectively engage all students. The one-size-fits-all approach to content delivery can lead to disengagement among students who do not find the material relevant or interesting to their individual interests or learning styles.
- **Limited Interaction and Engagement:** In traditional classrooms, opportunities for interactive learning and student engagement may be limited. Passive learning

through lectures and textbooks can hinder active participation and critical thinking skills development. This lack of engagement may result in decreased motivation and interest in the subject matter.

- **Inadequate Support for Diverse Learners:** Students with different learning needs, such as those with learning disabilities or English language learners, may not receive adequate support in traditional classrooms. The lack of specialized instruction and resources can impede their academic progress and overall educational experience.
- **Difficulty in Monitoring Progress:** Traditional learning environments may face challenges in effectively monitoring and tracking student progress. Without robust assessment and tracking mechanisms in place, educators may struggle to identify areas where students need additional support or intervention, leading to potential gaps in learning.

2.4. Problem Statement

"Current Ed-Tech platforms suffer from lack of privacy and personalization, leading to data breaches and deficiency in tailoring the learning experience as users seek more engaging educational experiences. "

Input: Our EdTech Platform project receives various educational data sources, including student profiles, learning materials, assessments, performance metrics, edge device data, local user data, and model updates.
Output: The primary output is personalized educational recommendations, including learning materials, quizzes, assessments, pathways, progress tracking, and insights for students.

2.5. Proposed Solution

To design and develop a Privacy Protected Federated Adaptive Learning Platform that provides personalized learning by adapting content to individual performance, with comprehensive analytics for educators. It ensures data privacy through features like homomorphic encryption and federated learning, delivering secure and tailored educational experiences for students.

CHAPTER 3

REQUIREMENT ENGINEERING

Chapter 3

REQUIREMENTS ENGINEERING

3.1 Software and Hardware Tools Used

3.1.1 Software Tools

1. **Machine Learning Frameworks (scikit-learn):**

    - Scikit-learn is a popular open-source machine learning library in Python. It provides simple and efficient tools for data mining and data analysis, including various algorithms for classification, regression, clustering, and dimensionality reduction. Its ease of use and extensive documentation make it a preferred choice for implementing machine learning models in educational platforms.

2. **Privacy-Preserving Libraries (CKKS Algorithm for Homomorphic Encryption):**

    - The CKKS algorithm is a homomorphic encryption scheme suitable for computations on encrypted data in machine learning applications. It allows for privacy-preserving operations such as addition and multiplication on encrypted data, enabling secure model training and inference while protecting sensitive user information. Integrating CKKS into the platform ensures that user data remains confidential during federated learning processes.

3. **Edge Computing (ML Models for Managing Edge Computing Resources):**

    - Edge computing involves processing data closer to the source of data generation, reducing latency and bandwidth usage. Machine learning models deployed on edge devices can optimize resource allocation, workload distribution, and task scheduling to maximize efficiency and performance. These models enable intelligent management of edge computing resources, ensuring optimal utilization and responsiveness in a decentralized environment.

4. **Communication Protocols (Client-Server Architecture for Secure and Efficient Communication):**

    - Implementing a client-server architecture facilitates secure and efficient communication between edge devices and the central server. Protocols such as HTTPS (Hypertext Transfer Protocol Secure) ensure encrypted data transmission, while RESTful APIs (Representational State Transfer) enable seamless interaction between clients and servers, supporting functionalities like data synchronization, model updates, and user authentication in the educational platform.

5. **Cloud Services (AWS, Google Cloud, Microsoft Azure):**

    - Cloud platforms like AWS, Google Cloud, or Microsoft Azure offer scalable infrastructure and services for hosting centralized components of the educational platform. These cloud services provide reliable storage, computing power, and networking capabilities, facilitating the deployment and management of federated learning algorithms, data storage, and processing tasks across distributed environments.

6. **Database (Combination of Hard Drive and Cloud Storage - MongoDB):**

    - A combination of local hard drive storage and cloud-based databases like MongoDB enables efficient storage and retrieval of data and machine learning models in the educational platform. MongoDB's document-oriented architecture and scalability support flexible data schemas and high-performance queries, while local storage ensures data availability and resilience in offline scenarios.

7. **Version Control (Git and GitHub):**

    - Git and GitHub are widely used version control systems for tracking changes to code repositories and facilitating collaboration among developers. By using Git and GitHub, developers can manage code versions, track modifications, and coordinate contributions from multiple team members effectively, ensuring code quality, reproducibility, and maintainability in the development of the educational platform

      .

8. **Integrated Development Environment (IDE) (Jupyter Notebooks and VS Code):**

    - Jupyter Notebooks and VS Code are popular integrated development environments for writing, executing, and debugging code in Python and other programming languages. Jupyter Notebooks provide an interactive computing environment for data exploration and experimentation, while VS Code offers a lightweight and versatile code editor with extensive plugin support, enhancing productivity and collaboration among developers.

9. **Software Framework (React for Frontend and Flask for Backend):**

    - React is a JavaScript library for building interactive user interfaces, making it suitable for developing the frontend of the educational platform. Its component-based architecture and virtual DOM enable efficient rendering and seamless updates of UI elements, enhancing user experience and responsiveness. Flask, a lightweight Python web framework, is used for building the backend API services, handling requests, and serving dynamic content to clients, ensuring scalability and maintainability of the platform.

10. **Operating System (Windows 10 or Above, Any Linux Distribution):**

    - The choice of operating system for development and deployment depends on the preferences and requirements of the development team. Windows 10 or above provides a user-friendly environment with extensive software compatibility, while Linux distributions offer stability, security, and flexibility for server deployments and development environments. Both operating systems support the development and execution of the educational platform's components, ensuring compatibility and reliability across different environments.

3.1.2 Hardware Tools

Hardware Tools:

1. **Edge Devices**:

- Laptop, mobile phones or other edge devices for implementing federated learning on the edge.

2. **Central Server:**

- A powerful server or cloud instance for aggregating and updating global models.

3. **Storage Devices:**

- Sufficient storage for storing models, data, and other project-related files.

4. **GPU/TPU :**

- Depending on your system's configuration, you can use GPU/TPU or CPU by default for training the models.

3.2 Conceptual/ Analysis Modeling

3.2.1 Use case diagram

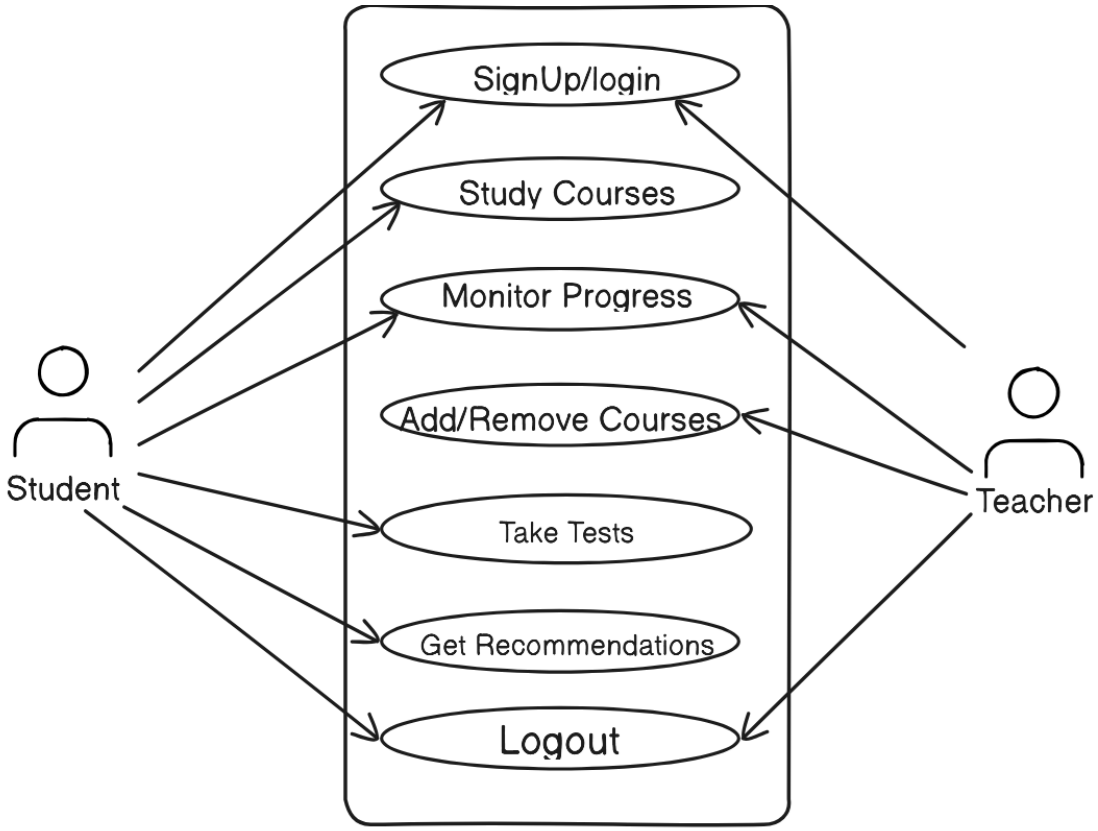

Fig 3.1 Use case diagram

The use case diagram outlines the interactions between users (students and teachers) and the educational platform, illustrating the various functionalities available to each user type.

For Students:

- **Signup/Login:** Students can create an account by signing up with their personal details and then log in to access the platform.
- **Study Courses:** Once logged in, students can browse and enroll in courses available on the platform. They can access course materials, lectures, quizzes, and other learning resources.
- **Monitor Progress:** Students can track their learning progress through personalized dashboards, which display metrics such as completed modules, quiz scores, and overall performance.
- **Take Tests:** Students can take quizzes and tests associated with their enrolled courses to assess their understanding of the material.
- **Get Recommendations**: Based on their learning progress and performance, students receive personalized recommendations for additional courses or resources that align with their interests and goals.
- **Logout:** Students can securely logout from their accounts to end their session and protect their privacy.

For Teachers:

- **Signup/Login:** Teachers can register for an account on the platform using their credentials and then log in to access their account.
- **Logout:** Similar to students, teachers can securely logout from their accounts to end their session.
- **Monitor Progress:** Teachers have access to tools and dashboards that allow them to monitor the progress of their students. They can view metrics such as course completion rates, quiz scores, and overall performance.
- **Add/Remove Courses:** Teachers have the authority to add new courses to the platform or remove existing ones. They can upload course materials, create quizzes, and manage the content of their courses to ensure relevance and quality for students.

In summary, the use case diagram illustrates how students and teachers interact with the educational platform to engage in various activities such as learning, monitoring progress,

and managing course content. The platform serves as a comprehensive tool for both students and teachers to facilitate effective teaching and learning experiences.

3.2.2 Activity diagram

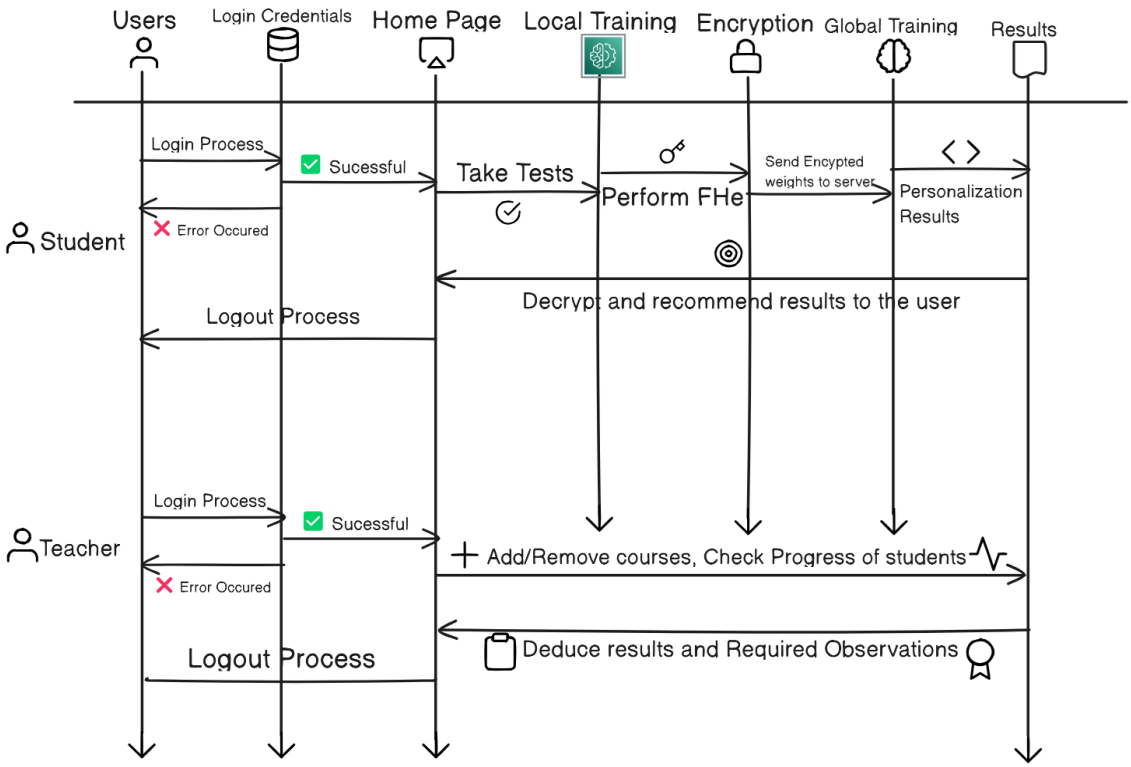

Fig 3.2 Activity diagram

Activity Diagram for Student:

1. **Sign Up/Login:**
    - Start: User selects the "Sign Up" or "Login" option.
    - If the user chooses "Sign Up," they provide personal details and create an account.
    - If the user selects "Login," they enter their credentials.
    - End: User is logged into the platform.
2. **Browse and Enroll in Courses:**
    - Start: User is logged into the platform.
    - User navigates to the "Courses" section.

- User browses available courses and selects desired ones.
- User enrolls in selected courses.
- End: User is enrolled in the chosen courses.

3. **Access Course Materials:**
    - Start: User is enrolled in one or more courses.
    - User selects a course from their enrolled list.
    - User accesses course materials, such as lectures, readings, and assignments.
    - End: User has accessed the desired course materials.
4. **Take Tests and Quizzes:**
    - Start: User is enrolled in a course with quizzes/tests.
    - User navigates to the quiz/test section of the course.
    - User selects a quiz/test to take.
    - User answers quiz/test questions.
    - End: User completes the quiz/test and receives results.
5. **Monitor Progress:**
    - Start: User is logged into the platform.
    - User accesses the "Progress" or "Dashboard" section.
    - User views metrics such as completed modules, quiz scores, and overall performance.
    - End: User has monitored their learning progress.
6. **Receive Recommendations:**
    - Start: User is logged into the platform.
    - User receives personalized recommendations based on their learning progress and performance metrics.
    - End: User has received recommendations for additional courses or resources.

7. **Logout:**
    - Start: User is logged into the platform.
    - User selects the "Logout" option.
    - End: User is logged out of the platform.

Activity Diagram for Teacher:

1. **Sign Up/Login:**
    - Start: User selects the "Sign Up" or "Login" option.
    - If the user chooses "Sign Up," they provide personal details and create an account.
    - If the user selects "Login," they enter their credentials.
    - End: User is logged into the platform.
2. **Monitor Progress:**
    - Start: User is logged into the platform.
    - User accesses the "Progress" or "Dashboard" section.
    - User views metrics such as student course completion rates, quiz scores, and overall performance.
    - End: User has monitored student progress.
3. **Add/Remove Courses:**
    - Start: User is logged into the platform.
    - User navigates to the "Courses" section.
    - User selects the option to add or remove courses.
    - User uploads course materials, creates quizzes, and manages course content as needed.
    - End: User has added/removed courses and managed course content.
4. **Logout:**
    - Start: User is logged into the platform.

- User selects the "Logout" option.
- End: User is logged out of the platform.

These activity diagrams illustrate the sequential flow of actions performed by students and teachers within the educational platform, depicting how they interact with the system to accomplish various tasks.

3.3 Software Requirements Specification

3.3.1 Functional Requirements

Functional Requirements:

- User Authentication and Authorization
- User Profile Management
- Federated Learning Integration
- Privacy-Preserving Mechanisms
- Adaptive Learning
- Edge Device Management
- Progress Tracking and Reporting

1. User Authentication and Authorization:

- This feature ensures that only authorized users can access the platform by verifying their identity through credentials like usernames and passwords. Authentication confirms the user's identity, while authorization determines what actions and resources they can access based on their role or permissions within the system. By implementing robust authentication and authorization mechanisms, the platform maintains security and prevents unauthorized access to sensitive information and functionalities.

2. User Profile Management:

- User profile management enables individuals to create and manage their profiles within the platform. Users can personalize their experience by providing information such as their preferences, learning objectives, and past performance data. This information helps the platform deliver tailored recommendations,

content, and learning pathways that align with each user's unique needs and goals. Additionally, users can update their profiles over time to reflect changes in their preferences or objectives, ensuring a continuously optimized learning experience.

3. Federated Learning Integration:

- Federated Learning integration leverages decentralized edge devices to collaboratively train machine learning models without centralizing sensitive user data. Instead of aggregating data in a central server, federated learning distributes model training tasks to edge devices, allowing the models to learn from local data while preserving user privacy. By integrating federated learning into the platform, users can benefit from personalized recommendations and insights without compromising the confidentiality of their data, thus enhancing privacy and security in the learning process.

4. Privacy-Preserving Mechanisms:

- Privacy-preserving mechanisms ensure that user data remains confidential and protected throughout the platform's operations. Techniques such as homomorphic encryption and decentralized storage are employed to encrypt sensitive data during model training and aggregation, preventing unauthorized access or data breaches. By prioritizing privacy, the platform fosters trust among users and complies with data protection regulations, reinforcing its commitment to safeguarding user information while delivering personalized learning experiences.

5. Adaptive Learning:

- Adaptive learning algorithms analyze user interactions, performance metrics, and other relevant data to dynamically adjust the content and learning pathways presented to each user. By continuously adapting to individual learning styles, preferences, and progress, the platform optimizes the educational experience for maximum effectiveness and engagement. Adaptive learning enhances user satisfaction, retention, and learning outcomes by providing personalized recommendations and content that cater to each user's specific needs and abilities.

6. Edge Device Management:

- Edge device management functionality oversees the coordination and orchestration of model training tasks across a distributed network of edge devices. It ensures

efficient utilization of resources, load balancing, and task scheduling to optimize performance and reliability. By managing edge devices effectively, the platform maximizes the scalability, responsiveness, and resilience of federated learning processes, enabling seamless collaboration and model improvement while minimizing latency and resource constraints.

7. Progress Tracking and Reporting:

- Explanation: Progress tracking and reporting tools enable users to monitor their learning journey, track their performance metrics, and receive feedback on their progress. Users can view detailed reports, analytics, and insights that highlight their strengths, weaknesses, achievements, and areas for improvement. By providing actionable feedback and performance indicators, progress tracking empowers users to set goals, track their progress, and make informed decisions to enhance their learning outcomes and personal development.

3.3.2 Nonfunctional Requirements

1.Performance: The platform must deliver responsive and efficient user experiences, ensuring quick content adaptation based on individual performance metrics.

2.Scalability: The system should seamlessly handle increased user load and content volume without compromising performance or responsiveness.

3.Reliability: The platform must exhibit high reliability, minimizing downtime and ensuring continuous availability for users and educators.

4.Privacy Compliance: The system must adhere to stringent privacy standards, implementing features such as homomorphic encryption and federated learning to safeguard user data.

5.Usability: The platform should provide an intuitive and user-friendly interface for both students and educators, promoting ease of navigation and interaction.

6.Security Compliance: Robust security measures must be in place to protect against unauthorized access, data breaches, and ensure the confidentiality and integrity of user information.

7.Documentation: Comprehensive and easily accessible documentation should be available for users and administrators, covering system functionalities, configurations, and troubleshooting guidelines.

8.Resource Efficiency: The platform should optimize resource utilization, ensuring efficient use of computing resources and minimizing environmental impact.

9.Compatibility: The system must be compatible with various devices, browsers, and operating systems to accommodate diverse user preferences and ensure a seamless learning experience.

10.Training and Support: Adequate training materials and support services should be provided to users and educators to facilitate efficient system adoption and troubleshoot any issues that may arise.

3.3.3 Domain Requirements

Domain requirements for an educational platform leveraging federated learning and privacy-preserving mechanisms should encompass various aspects to ensure its effectiveness, security, and usability. Here are key domain requirements to consider:

1. User Authentication and Authorization:

- Implement secure user authentication methods, such as username/password, multi-factor authentication, or OAuth, to verify user identities.

- Define roles and permissions to control access to different functionalities and data based on user roles (e.g., student, teacher, administrator).

2. Privacy-Preserving Mechanisms:

- Utilize homomorphic encryption, differential privacy, or other privacy-preserving techniques to protect sensitive user data during model training, inference, and data aggregation processes.

- Ensure compliance with data protection regulations (e.g., GDPR, CCPA) by anonymizing or pseudonymizing personal information and obtaining user consent for data processing.

3. Federated Learning Integration:

- Integrate federated learning frameworks (e.g., TensorFlow Federated, PySyft) to enable collaborative model training across decentralized edge devices while preserving data privacy.

- Design federated learning algorithms for efficient model synchronization, aggregation, and update propagation across distributed nodes.

4. Adaptive Learning and Personalization:

- Develop adaptive learning algorithms to analyze user interactions, preferences, and performance data for personalized content recommendations and learning pathways.

- Implement adaptive assessment strategies to dynamically adjust quiz difficulty and content based on individual learner proficiency and progress.

5. Edge Device Management:

- Manage edge computing resources efficiently by deploying machine learning models, workload scheduling algorithms, and resource allocation strategies tailored to edge device capabilities and network conditions.

- Monitor and optimize edge device performance, connectivity, and energy consumption to ensure reliable and responsive operation in distributed learning environments.

6. Content Management and Delivery:

- Provide tools for content creators and instructors to upload, organize, and update course materials, lectures, assignments, and assessments.

- Support various content formats (e.g., text, video, interactive simulations) and delivery modes (e.g., synchronous, asynchronous) to accommodate diverse learning preferences and needs.

7. Progress Tracking and Reporting:

- Track and visualize learner progress, achievements, and engagement metrics through interactive dashboards, progress reports, and performance analytics.

- Enable educators to monitor student performance, identify learning gaps, and provide timely feedback and intervention to support individual learner needs.

8. Collaboration and Communication:

- Facilitate collaboration and communication among learners, instructors, and peers through discussion forums, messaging features, and virtual classrooms.

- Integrate real-time chat, video conferencing, and collaborative editing tools to foster interactive learning experiences and peer-to-peer knowledge sharing.

9. Accessibility and Inclusivity:

- Design the platform with accessibility features (e.g., screen reader compatibility, keyboard navigation, alternative text) to accommodate users with disabilities and diverse learning needs.

- Ensure content is available in multiple languages and formats to cater to learners from different cultural backgrounds and learning preferences.

10. Scalability and Performance:

- Architect the platform for scalability and performance to accommodate growing user base, increasing data volume, and complex computational tasks.

- Employ scalable database systems, distributed computing frameworks, and cloud infrastructure to handle concurrent user requests, data processing, and model training tasks efficiently.

By addressing these domain requirements, the educational platform can offer a secure, personalized, and inclusive learning environment while leveraging federated learning and privacy-preserving mechanisms to protect user privacy and data confidentiality.

CHAPTER 4

PROJECT PLANNING

Chapter – 4

PROJECT PLANNING

4.1 Project Planning and Scheduling

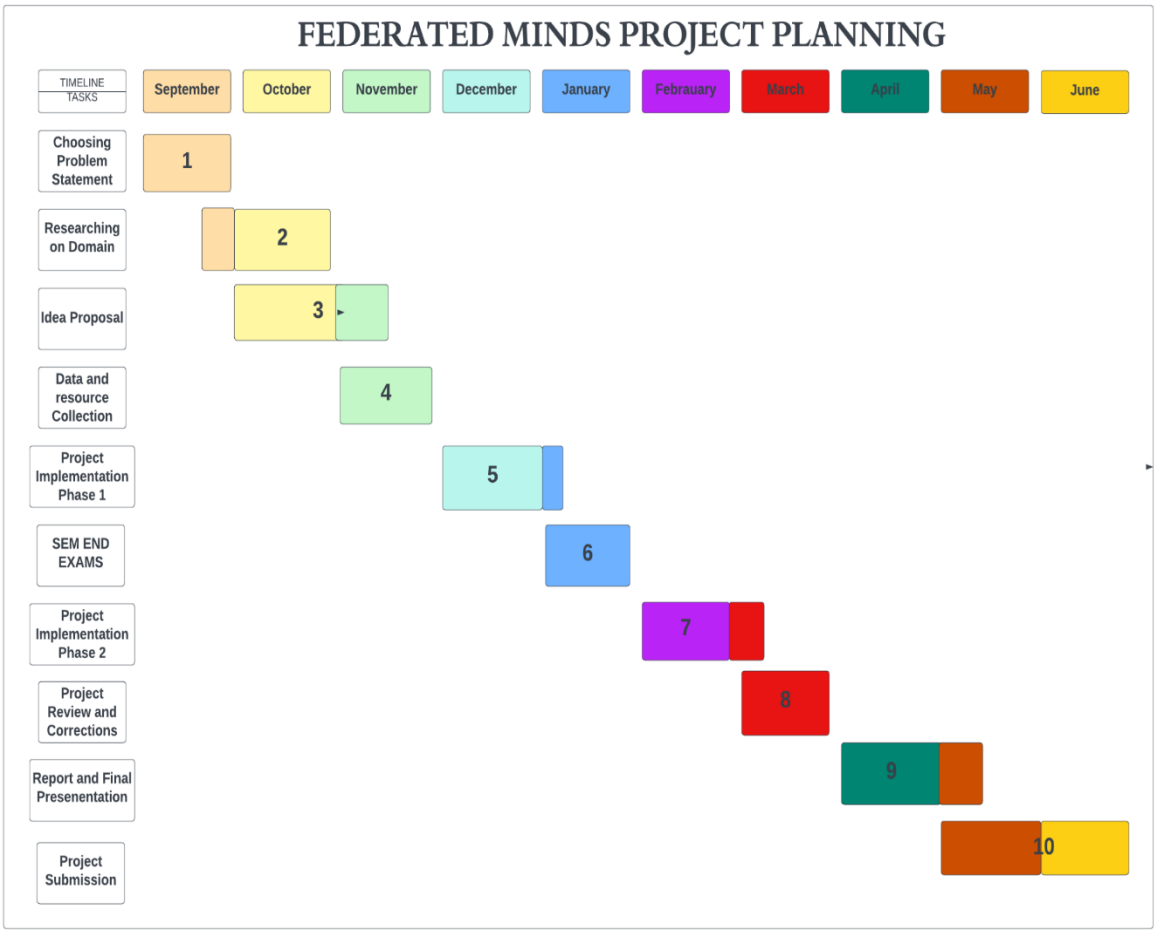

Fig 4.1 Project planning Gantt Chart

The problem statement selection phase in September is not merely about choosing a topic but also about understanding its significance and relevance. It involves conducting preliminary research to identify pressing issues or gaps in the chosen domain. This step may also involve consultations with mentors or experts to refine the problem statement and ensure its alignment with academic or research objectives. As the project progresses into mid-September to October, the research phase becomes more intensive. Beyond merely gathering information, this stage involves critical analysis and synthesis of existing literature.

Researchers delve into various sources such as academic journals, conference proceedings, and credible online resources to develop a comprehensive understanding of the subject

matter. It also includes techniques like literature reviews, comparative analyses, and trend identification to identify key insights and potential avenues for exploration. During the idea proposal phase from October to mid-November, researchers not only generate ideas but also evaluate their feasibility and potential impact. This phase may involve brainstorming sessions, idea generation workshops, and peer discussions to refine and validate proposed solutions or approaches.

Additionally, researchers may conduct feasibility studies, stakeholder analyses, or pilot experiments to assess the practicality and effectiveness of their proposed ideas. As November unfolds, the focus shifts towards data and resource collection. This phase involves more than just gathering raw data; it requires careful planning and coordination to ensure the availability of necessary resources for subsequent phases. Researchers may employ various methods such as surveys, interviews, experiments, or simulations to collect primary data. Simultaneously, efforts are made to secure access to relevant secondary data sources, software tools, equipment, or facilities required for project implementation.

By incorporating these additional elements into each phase, the project planning and scheduling become more robust and comprehensive. This holistic approach ensures that every aspect of the research or academic project receives the attention and resources it deserves, ultimately contributing to its successful execution and completion.

CHAPTER 5
SYSTEM DESIGN

Chapter 5

SYSTEM DESIGN

5.1 System Architecture

System architecture refers to the overall design and organization of a computer system, which includes hardware components, software components, and the communication and interaction between them. It defines the way in which the system's components are connected, how they operate together to achieve the system's objectives, and how they are managed and maintained over time. A well-designed system architecture is critical for ensuring that a computer system is scalable, secure, reliable, and maintainable

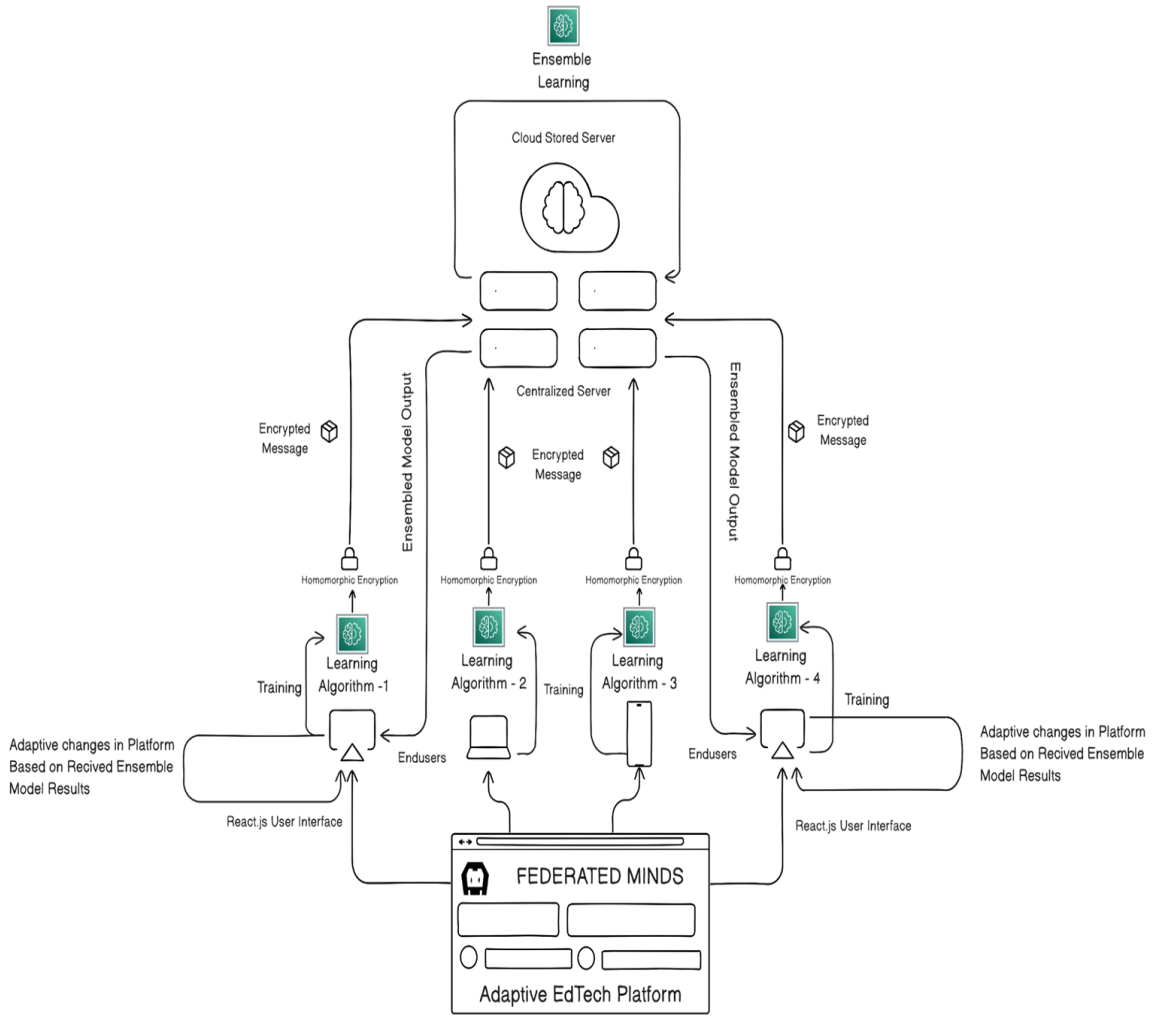

Fig 5.1 System Architecture

The educational platform's architecture seamlessly integrates Federated Learning, homomorphic encryption, and MongoDB storage to revolutionize personalized learning while safeguarding user privacy. Federated Learning serves as the backbone, allowing for

collaborative model training across distributed edge devices without compromising sensitive user data. Each user's device hosts a locally trained model, ensuring privacy by keeping individual data decentralized. These locally trained models periodically synchronize with a central server, contributing to the refinement of the global model while maintaining data confidentiality. Homomorphic encryption adds an extra layer of security, ensuring that user data remains encrypted during model training and aggregation, thus preventing any breaches of privacy.

MongoDB storage plays a crucial role in efficiently managing and retrieving encrypted user data, model parameters, and aggregated predictions. MongoDB's scalability and flexibility accommodate the platform's expanding user base and evolving educational content needs. The platform's intelligence lies in its ability to derive personalized learning recommendations from aggregated insights obtained through Federated Learning. By analyzing performance metrics and global predictions, the system generates tailored learning pathways for each user. This personalized approach empowers users to pursue their educational goals confidently while receiving individualized support and guidance.

The platform's user interface, built with React.js, provides an intuitive and user-friendly experience for accessing personalized recommendations and tracking learning progress. Local model training on edge devices ensures active user participation in model refinement, fostering a sense of ownership and collaboration within the learning community. The continuous evolution of the global model reflects the diverse learning needs and preferences of users, promoting inclusivity and equity in education. Overall, this architecture represents a significant advancement in educational technology, offering personalized learning experiences while upholding the highest standards of user privacy and data security.

5.2 Module Decomposition

Frontend Module:

The module decomposition of the educational platform involves a structured breakdown of the system components into frontend and backend modules, each serving distinct functionalities to support the platform's operations.

In addition to the student and teacher portals, the frontend module also includes a responsive design feature to ensure optimal user experience across various devices and screen sizes. This adaptability allows users to seamlessly access the platform from desktops, laptops, tablets, and smartphones, enhancing accessibility and usability.

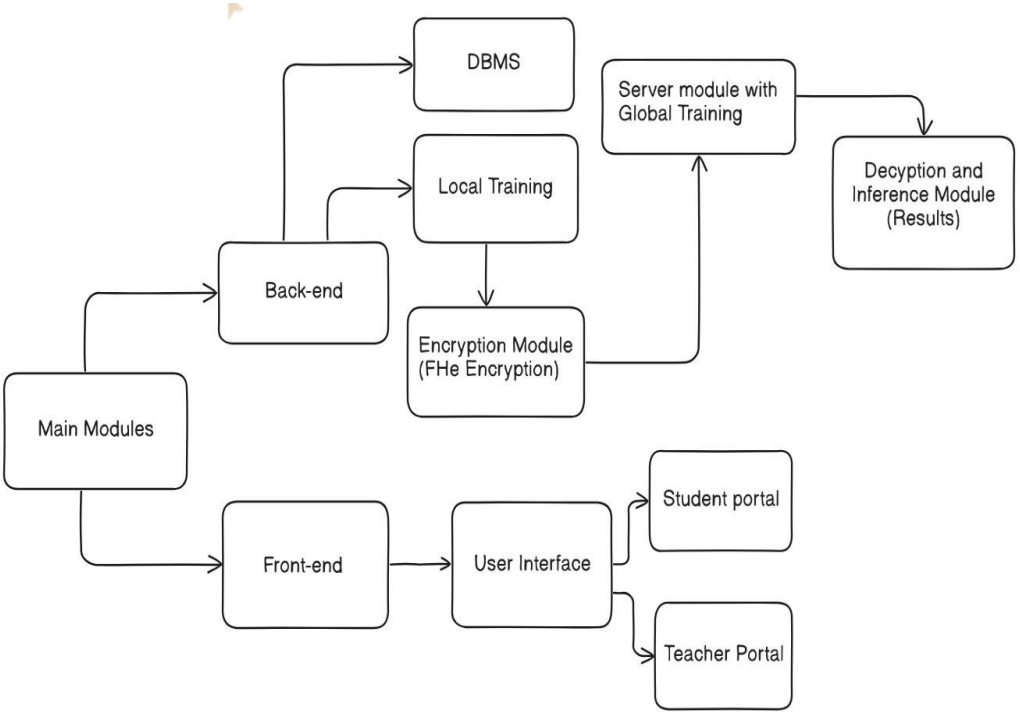

Fig 5.2 Module Decomposition Diagram

Furthermore, the frontend module incorporates real-time collaboration tools, facilitating interactive discussions, group projects, and live feedback sessions between students and teachers. These collaborative features promote engagement and foster a dynamic learning environment where students can actively participate and interact with their peers and instructors.

Backend Module: Apart from local training and homomorphic encryption, the backend module integrates advanced analytics and reporting functionalities to provide insights into user behaviour, performance trends, and content effectiveness. By leveraging data analytics, the platform can generate personalized recommendations, adaptive learning pathways, and progress tracking tools tailored to individual student needs.

Moreover, the backend module includes robust security measures such as multi-factor authentication, role-based access control, and audit logging to safeguard sensitive information and prevent unauthorized access. These security protocols ensure compliance with data protection regulations and bolster user trust in the platform's confidentiality and integrity mechanisms.

By incorporating these additional features into both the frontend and backend modules, the educational platform can offer a comprehensive and scalable solution that addresses the

diverse needs of students, teachers, and administrators while prioritizing user privacy, data security, and pedagogical effectiveness.

Local Training on Edge Devices: Edge devices, such as laptops, tablets, or smartphones, perform local model training using locally available data. This decentralized approach allows users to train models on their devices while preserving data privacy.

Homomorphic Encryption for Secure Communication: Updates from locally trained models are encrypted using homomorphic encryption before being sent to the global server. This ensures that sensitive data remains protected during transmission, as operations can be performed on encrypted data without the need for decryption.

Global Server: The global server aggregates encrypted updates from edge devices, performs model aggregation, and sends updated global models back to the edge devices. This iterative process enables continuous model refinement while maintaining user privacy and data security.

5.3 Interface Design

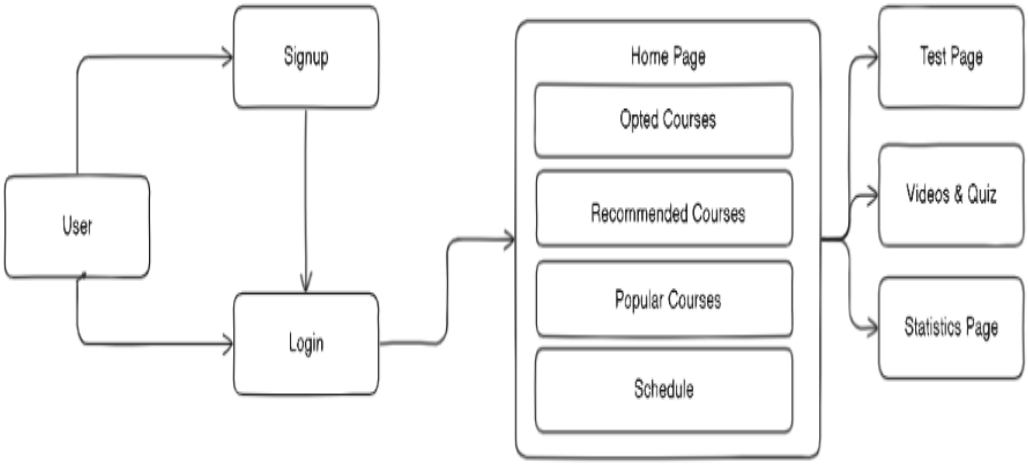

Fig 5.3 Interface Design Diagram

Upon clicking the "Sign Up" button, new users are prompted to provide necessary details for registration, including a unique username, valid email address, and a secure password. This initial step ensures the creation of a personalized account tailored to their learning needs within the platform. Once the registration information is submitted, the system validates the input and creates a new user profile, securely storing the provided credentials for future authentication.

After successful registration, users are redirected to the platform's landing page, serving as the central hub for accessing various features and functionalities. Here, users can explore the diverse offerings of the educational platform, ranging from courses and recommendations to upcoming events and schedules. The landing page offers a seamless transition into the platform's ecosystem, welcoming users to begin their educational journey with ease.

Upon subsequent visits, users can log into the platform using their previously created credentials. Upon successful authentication, they are directed to their personalized home page, designed to provide quick access to relevant information and actions. The home page offers curated sections such as courses, recommended courses based on user preferences, popular courses, and schedules, allowing users to navigate the platform effortlessly and engage with content tailored to their interests and learning goals. Additionally, users can access their statistics and progress tracking features directly from the home page or their profile dashboard, empowering them to monitor their learning journey, identify areas for improvement, and set meaningful learning goals based on their performance metrics.

5.4 Data Structure Design

The data structure design comprises four key collections: User, Courses, Dataset Users, and Questions. Each collection serves a distinct purpose in facilitating the functionality and organization of the educational platform.

1. User Collection (User):
    - [Name, Email, Password, Opted courses]:
        - Name: The user's full name, enabling personalized communication and identification.
        - Email: User's email address, serving as a unique identifier and means of communication.
        - Password: Encrypted user password for secure authentication and access control.
        - Opted courses: A list or array containing the courses that the user has chosen to enroll in, allowing for personalized recommendations, progress tracking, and facilitating access to course materials and resources.

2. Courses Collection (Courses):
    - [Course, Performance, CourseID, OptedCount]:
        - Course: The title or name of the course offered on the platform, providing clarity on the subject matter.
        - Performance: Metrics related to the course's performance, such as completion rates, average scores, user ratings, or feedback, enabling users to gauge the quality and effectiveness of the course.
        - CourseID: A unique identifier assigned to each course, often used for database management and referencing in the platform's backend systems.
        - OptedCount: A numerical value representing the count of users who have opted for or enrolled in the course, allowing administrators and users to understand the popularity and demand for each course.
3. Dataset Users Collection (Dataset Users):
    - [UserID, Dataset]:
        - UserID: A unique identifier or reference to the user from the User collection, establishing a link between user-specific data and associated datasets.
        - Dataset: User-specific datasets or data profiles containing information such as learning preferences, browsing history, quiz attempts, progress tracking, or performance metrics. These datasets enable the platform to deliver personalized learning experiences, recommendations, and content suggestions tailored to each user's needs and preferences.
4. Questions Collection (Questions):
    - [Subject, Question, Options, Tag, Answer]:
        - Subject: The subject area, topic, or category to which the question belongs, facilitating organization, filtering, and search functionality within the platform.

- Question: The textual content of the educational question or prompt, providing users with learning material, challenges, or assessments.
- Options: An array or list of multiple-choice options accompanying the question, allowing users to select the correct answer or response from the provided choices during quizzes, assessments, or interactive learning activities.
- Tag: Keywords, labels, or metadata associated with the question, aiding in categorization, classification, and retrieval of related content based on thematic or topical similarities.
- Answer: The correct answer to the question, enabling automated grading, feedback generation, and assessment of user responses within the platform**.**

5.4 Algorithm Design

1. Initialization and Setup:
    - Import necessary libraries and frameworks, ensuring compatibility and functionality for federated learning, homomorphic encryption, Flask web server, and MongoDB interaction.
    - Configure environment variables and connection strings for MongoDB to establish secure connections.
    - Initialize virtual workers and hooks for federated learning using PySyft, ensuring secure computation on encrypted data across decentralized edge devices.
2. Prediction Route Definition:
    - Define a route '/predict' within the Flask application to handle prediction requests securely.
    - Implement input validation and sanitization techniques to prevent injection attacks and ensure the integrity of incoming data.
    - Utilize HTTPS protocol and SSL/TLS encryption to secure communication channels between the client and the server.

3. Feature Engineering:

    - Preprocess input features (e.g., timespent, quizscore, quizzattempts) securely, applying normalization or scaling techniques while preserving data privacy.
    - Employ homomorphic encryption algorithm to encrypt input features, ensuring confidentiality during model training and inference.

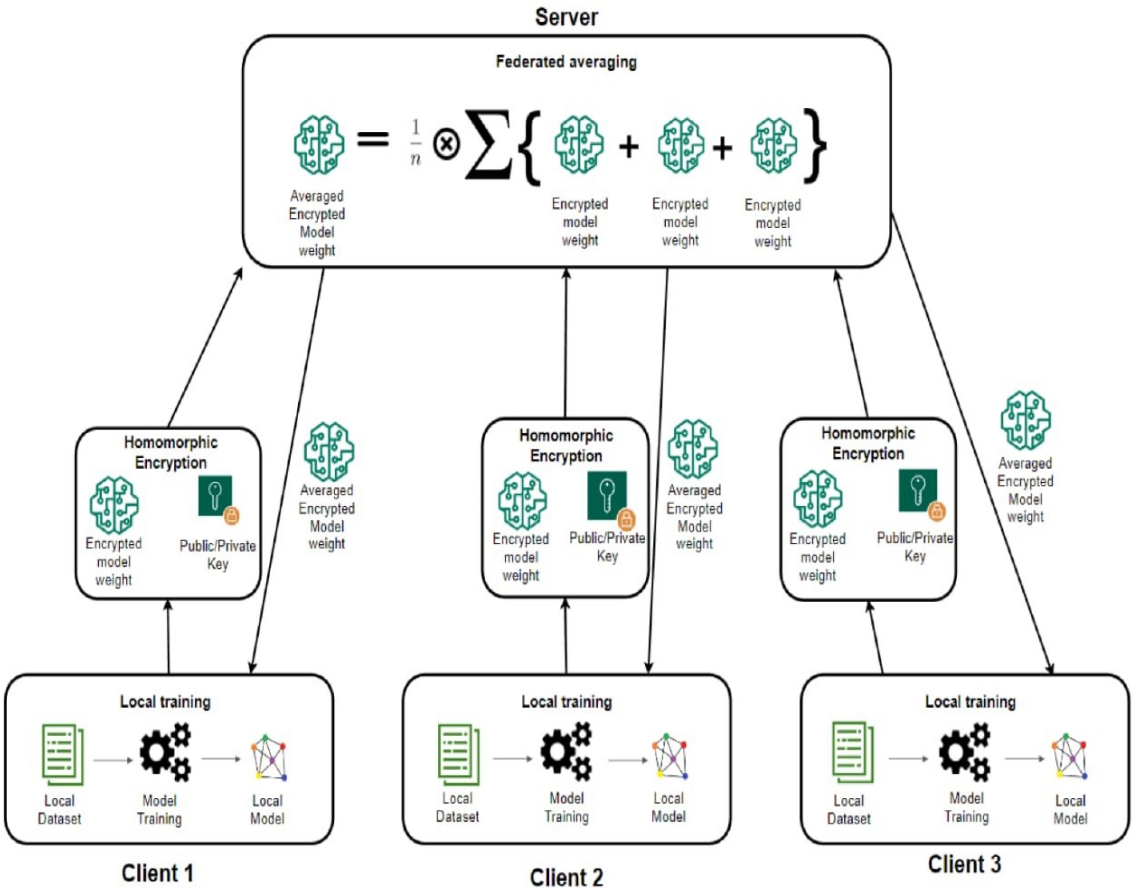

Fig 5.4 Algorithm Working Explanation

4. Model Loading and Training:

    - Load a pre-trained machine learning model or initialize a new model instance securely within the federated learning environment.
    - Implement federated learning algorithms (e.g., Federated Averaging) ensuring encrypted model updates are securely aggregated across participating edge devices.
    - Incorporate differential privacy techniques to further enhance data privacy by adding noise to model updates or training data.

5. Prediction:

    - Use the federated learning model to predict course performance for the given encrypted input features securely.
    - Employ homomorphic decryption techniques to decrypt the prediction result securely, ensuring confidentiality while obtaining the actual performance value.

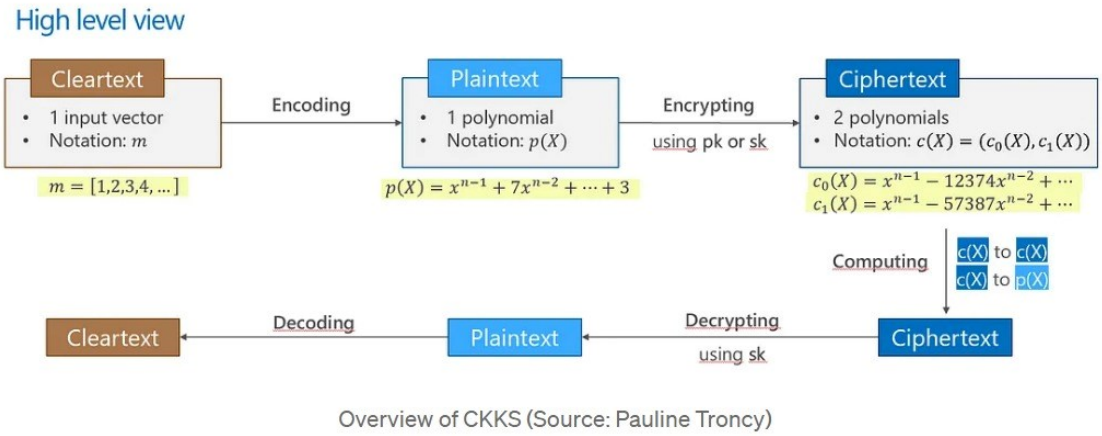

Fig 5.5 CKKS Homomorphic Encryption Algorithm

6. Update MongoDB:

    - Update performance metrics for the specified course and user securely within the MongoDB collection.
    - Utilize MongoDB's access control features and role-based authentication to restrict access to sensitive data and operations.

7. API Call to Aggregate Performance:

    - Define a route '/aggregate_performance' to securely handle HTTP POST requests for aggregating performance metrics across courses or users.
    - Implement HMAC-based message authentication to validate the integrity of incoming API requests and prevent tampering.

8. Retrieve Performance Data:

    - Query MongoDB collections securely to retrieve performance data for the specified course and user, ensuring proper access controls and encryption mechanisms are in place.

- Employ index encryption techniques to protect sensitive data stored in MongoDB, preventing unauthorized access even at the database level.

9. Subject-wise Recommendations Initialization:
    - Design a scalable recommendation system that generates subject-wise recommendations based on encrypted performance metrics and user preferences securely.
    - Utilize secure multi-party computation techniques to collaboratively generate personalized recommendations across decentralized edge devices while maintaining data privacy.
10. Run the Flask App:
    - Start the Flask application securely within a trusted execution environment (TEE) or containerized environment, ensuring isolation and integrity of the application.
    - Monitor and log security events and access attempts to detect and respond to potential security threats or breaches proactively.

By incorporating robust security measures, encryption techniques, and access controls at each step of the algorithm design, this comprehensive approach ensures the confidentiality, integrity, and availability of data and operations within the federated learning environment with homomorphic encryption for predicting course performance and aggregating performance metrics securely.

CHAPTER 6
IMPLEMENTATION

Chapter - 6

IMPLEMENTATION

6.1 Implementation Approaches

6.1.1 Machine Learning Model For Edge Prediction

In the context of the recommendations and performance prediction in educational platform, a machine learning model is employed to predict user performance. Specifically, a linear regression model is utilized in the project to analyze various user interactions and behaviours within the platform and predict the performance class of user. These interactions encompass factors such as time spent on courses, quiz scores, and quiz attempts. Leveraging historical user data, the model discerns patterns and correlations between these features and user performance levels. Through predictive modelling, the system gains the capability to forecast a user's performance based on their ongoing activities and engagements with the learning materials. This predictive insight serves as a valuable resource for users, educators, and administrators, offering actionable feedback and guidance.

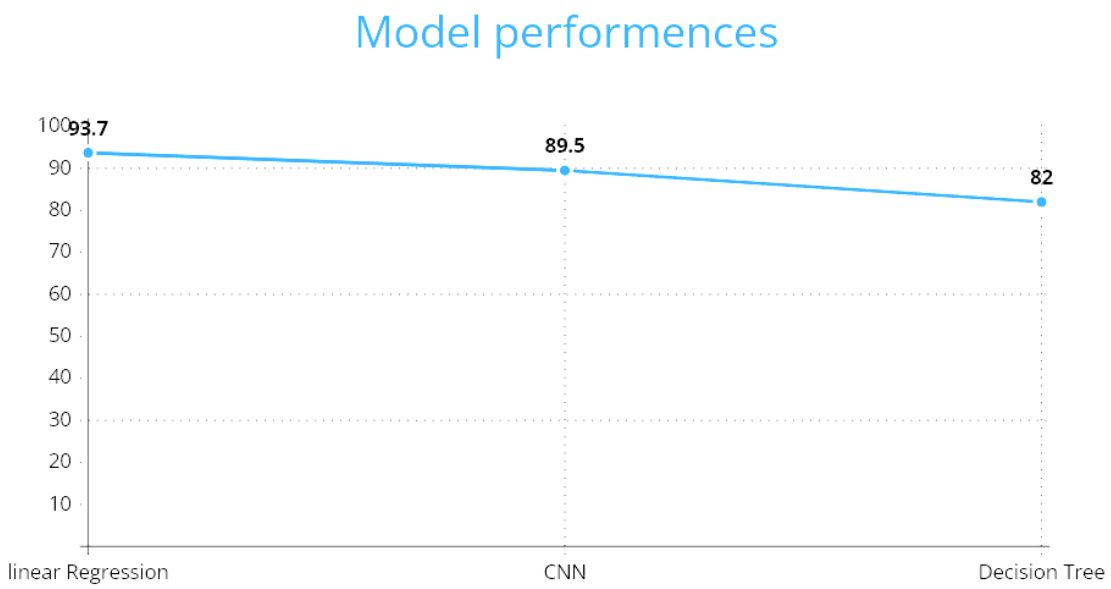

Fig 6.1 : Random Forest algorithm working

This graph shows the performances of the different supervised models we tested on mock data and by the results, linear regression achieved the most accuracy in terms of classification this is caused by very few output classes.

6.1.2 Encrypting the Edge Prediction and User Data

1. **Context Initialization:** A homomorphic encryption context is initialized with specific parameters, such as the polynomial modulus degree and coefficient modulus bit sizes. These parameters determine the security and efficiency of the encryption scheme.
2. **Key Generation:** The homomorphic encryption context allowed us to generate cryptographic keys, including the public key for encrypting data and the secret key for performing computations on the encrypted data.
3. **Data Encryption:** Before transmitting user data to the server, encrypt the relevant data using the public key generated by the homomorphic encryption scheme. This ensured that the data remained confidential and unreadable during transmission and storage.
4. **Encryption of Predictions:** After processing the encrypted data using machine learning models, such as linear regression, the resulting predictions using the homomorphic encryption scheme. This step preserved the privacy of the predicted performance scores while allowing for further computations.
5. **Storage of Encrypted Data:** The encrypted data, including the original user data and the encrypted predictions, were securely stored in the database. By storing only the encrypted data, It is ensured that sensitive information remains confidential and protected from unauthorized access.
6. **Secure Data Transmission:** Whenever data needs to be transmitted between the edge device and the server, It is ensured that the data remains encrypted using secure communication protocols. This prevented unauthorized interception or tampering of the encrypted data during transmission.

6.1.3 Aggregated Performance and Recommendations

To enhance user experience and provide personalized support, an aggregation mechanism is implemented to consolidate performance data and generate course recommendations. When a user completes a mock test or assessment, their performance data is aggregated at the course level. This aggregation entails analyzing the performance scores of all users who have interacted with the same course.

The aggregated performance score provides a holistic view of user engagement and effectiveness within a specific course. Based on this data, personalized recommendations

are generated for each user. These recommendations are derived from predefined mappings between performance levels and suggested learning materials or interventions.

By integrating machine learning predictions with aggregated performance analysis, the system offers users actionable insights and tailored recommendations for optimizing their learning experience. This approach empowers users to make informed decisions and facilitates continuous improvement within the educational environment.

Aggregated Performance Calculation:

- The aggregated performance score **APcourse** for a specific course is calculated as the weighted average of individual user performances:

$$APcourse = \sum i=1 \text{ to } n\ [(Pi \times Mi)] / \sum i=1 \text{ to } n [Mi]$$

- $PiPi$ represents the performance score of the $ithith$ user.
- $MiMi$ represents the total marks obtained by the $ithith$ user.
- nn is the total number of users who have interacted with the course.

Recommendations Generation:

- Recommendations for a user are generated based on their performance level $PLPL$ within a specific course. These recommendations are predefined mappings between performance levels and suggested learning materials or interventions. For example:
- If $PL=0PL=0$, recommend basic learning materials.
- If $PL=1PL=1$, recommend intermediate-level learning materials.
- If $PL=2PL=2$, recommend advanced learning materials.

6.2 Coding Details

6.2.1 ML Model

Import Necessary packages

```
from sklearn.linear_model import LinearRegression
import certifi
from sklearn.preprocessing import StandardScaler
```

Generating Model on Scaled Data

```
X_weighted = X[:, 1:] * weights[1:]
        # Concatenate the weighted features with the non-weighted features (excluding 'course_id')
    X_final = np.concatenate((X[:, :1], X_weighted), axis=1)
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X_final)
    model = LinearRegression()


    model.fit(X_scaled, y)
    # print(input_data)
    course = input_data["course_name"]
    # print(course)
    # Predict using the loaded model
    weighted_features = np.array([input_data['timespent'], input_data['quizscore'], input_data['quizzattempts']]) * weights[1:]
    input_final = np.concatenate(([input_data['course_id']], weighted_features))
    input_final_scaled = scaler.transform(input_final.reshape(1, -1))
    prediction = model.predict(input_final_scaled)[0]
```

6.2.2 Homomorphic Encryption

```
def encrypt(prediction,uid):
    vec =[prediction]
    context = ts.context_from(utils.read_data("public.txt"))
    encry_pred = ts.ckks_vector(context, vec)
    utils.write_data("encrypted_prediction.txt", encry_pred.serialize())
    fs = GridFS(DB)
    with open("encrypted_prediction.txt", 'rb') as file:
        file_id = fs.put(file, filename=f'encrypted_prediction_{uid}.txt')
    print("file uploaded",file_id)
```

6.2.3 Aggregation and Recommendation

```
    performance_data = result["performance"]
```

```
student_marks = np.array([entry["score"] for entry in performance_data])
local_predictions = np.array([performance_scores[entry["performance"]] for entry in performance_data])
weights = student_marks / np.sum(student_marks)
aggregated_prediction = np.sum(local_predictions * weights)
print("Aggregated Prediction:", aggregated_prediction)
if aggregated_prediction < 0.33:
        new_performance = "0"
 elif aggregated_prediction < 0.67:
        new_performance = "1"
    else:
        new_performance = "2"
# Access the datasets array within the document
datasets = user_document.get("datasets", [])
# Iterate through datasets to find the performance for the specified course_id
for dataset in datasets[::-1]:
            if dataset.get("course_id") == str(course_id):
                    performance_value = dataset.get("performance")
                    print(f"Performance for course_id {course_id}: {performance_value}")
                    print(type(course_id))
                    print(type(performance_value))
return subject_recommendations[str(course_id)][str(performance_value)]
```

CHAPTER 7

TESTING

Chapter - 7

TESTING

7.1 Testing Approach

Testing is a systematic and disciplined process of evaluating a software application or system to identify defects, errors, or discrepancies between expected and actual behavior. It involves executing the software with the intention of finding bugs, verifying that it meets specified requirements, and ensuring its overall quality.

The primary goal of testing is to uncover issues in the software and provide feedback to the development team, allowing them to address and fix any identified problems. Testing helps to ensure that the software functions as intended, is reliable, and meets the needs and expectations of its users.

7.1.1 Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. Unit Testing is done during the development (coding phase) of an application by the developers. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. Unit testing finds problems early in the development cycle. This includes both bugs in the programmer's implementation and flaws or missing parts of the specification for the unit

UTC-1: User Performance Prediction Model

The final user performance prediction model uses the linear regression model for prediction. In linear regression, the parameters, also known as coefficients or weights, represent the relationship between the input features and the output,the input features consist of various metrics such as time spent, quiz scores, and quiz attempts, while the output represents the predicted performance class label. Each parameter in the linear regression model corresponds to a specific feature, indicating how much that feature contributes to the prediction. For instance, a higher coefficient for the "time spent" feature suggests that more time spent correlates with better performance. When we provide input data to the model, it multiplies each feature value by its corresponding parameter and sums them up, along with an intercept term if present. This calculation forms the basis for

predicting the output performance class label. The output of the linear regression model is a continuous value representing the predicted performance class label. However, since you're performing classification with three classes (0, 1, 2). This is typically done by rounding the predicted value to the nearest integer. During unit testing, we evaluate the model's performance by comparing the predicted output with the expected output for a set of known input data. If the predicted output matches the expected output within an acceptable margin of error, the test case passes; otherwise, it fails. The purpose of unit testing is to ensure that the model behaves as expected for various input scenarios. By systematically validating the model against known data, we can identify any discrepancies or errors in its predictions. Ultimately, the goal is to achieve a reliable and accurate linear regression model that can effectively classify users into different performance categories based on their input metrics.

Test Case	Input Data	Expected Output	Actual Output	Pass/Fail
Test 1	[10, 20, 3]	1	1	Pass
Test 2	[5, 15, 2]	0	0	Pass
Test 3	[25, 30, 4]	2	2	Pass
Test 4	[12, 18, 3]	1	1	Pass
Test 5	[8, 22, 2]	0	0	Pass

Table 7.1 Unit Testing – 1

In this table:

- “Input Data” represents the input features (e.g., time spent, quiz scores, quiz attempts).
- “Expected Output” represents the expected class label predicted by the model.
- “Pass/Fail” indicates whether the predicted output matches the expected output.

UTC-2: User Performance Prediction Model

Homomorphic encryption is a powerful technique used to perform computations on encrypted data without decrypting it first. In our user performance prediction model, homomorphic encryption enhances data security by allowing us to process sensitive user metrics while preserving privacy.

We employ a partially homomorphic encryption scheme, specifically the Paillier cryptosystem, which supports addition and multiplication operations on encrypted data. This enables us to train and utilize the linear regression model on encrypted user input features without revealing the raw data to the model or any third party.

In our implementation, the user input features are encrypted before being sent to the model for prediction. The linear regression model operates on the encrypted data, performing the necessary computations to predict the user's performance class label. The result is then decrypted to obtain the final prediction, ensuring that sensitive user information remains confidential throughout the process.

During unit testing, we evaluate the homomorphic encryption module by comparing the decrypted predicted output with the expected output for a set of known encrypted input data. If the decrypted output matches the expected output within an acceptable margin of error, the test case passes; otherwise, it fails.

Test Case	Input Data	Expected Output	Actual Output	Pass/Fail
Test 1	[17, 23, 4]	1	1.002	Pass
Test 2	[6, 14, 3]	0	0.001	Pass
Test 3	[23, 31, 6]	2	1.998	Pass
Test 4	[11, 19, 2]	1	1.012	Pass
Test 5	[8, 22, 2]	0	0.003	Pass

Table 7.2 Unit Testing – 2

In these test cases:

- "Input Data " describes the user input features (e.g., time spent, quiz scores, quiz attempts).
- "Expected Output" indicates the expected behavior or result of the action.
- "Pass/Fail" determines whether the actual output matches the expected output.

UTC-2: User Interface Testing

The user interface (UI) module is a critical component of our system, providing users with an intuitive and engaging platform to interact with the federated learning system. Here, users can access course content, view statistics, take tests, and manage their accounts. Ensuring the UI functions as intended is essential for delivering a seamless user experience.

Test Case	Input Data	Expected Output	Actual Output	Pass/Fail
Test 1	User submits test by answering the questions	User gets course recommendations according to the test score	User gets course recommendations according to the test score	Pass
Test 2	User logs in with valid credentials	User is directed to the home page	User is directed to the home page	Pass
Test 3	User attempts to log in with invalid credentials	Error message: “Invalid username or password”	Error message: “Invalid username or password”	Pass
Test 4	User navigates to the “Courses” section	List of available, recommended and popular courses are displayed	List of available, recommended and popular courses are displayed	Pass
Test 5	User takes a test	Test is displayed with questions and options	Test is displayed with questions and options	Pass
Test 6	User accesses statistics page	Personalized statistics are displayed	Personalized statistics are displayed	Pass

Table 7.3 Unit Testing - 3

In these test cases:

- "Input/Action" describes the user action or input being tested.
- "Expected Output" indicates the expected behavior or result of the action.
- "Actual Output" shows what the system actually does in response to the input.
- "Pass/Fail" determines whether the actual output matches the expected output.

CHAPTER 8
RESULTS DISCUSSION AND PERFORMANCE EVALUATION

Chapter - 8

RESULTS DISCUSSION AND PERFORMANCE ANALYSIS

The results discussion section provides an interpretation and analysis of the main findings, highlighting their significance in the context of personal recommendations and the existing knowledge about student performance. The objective of the results section is to present the key findings objectively, while the discussion aims to interpret and describe the implications of these findings considering the problem being investigated.

The **performance analysis** across all three servers plays a crucial role in determining model efficiency and processing time, we evaluated the efficacy of the machine learning models and other techniques implemented in our system.

Key performance measures for the recommendation engine patient and encryptor are:

• **Accuracy**: Evaluating the model's ability to correctly identify PCOS-positive and

PCOS-negative cases based on ultrasound scans and relevant patient data.

• **Time Efficiency**: Assessing the speed and efficiency of the model in processing and

Firstly, on the global server, where predictive analytics and recommendation generation took place, we assessed the performance of the machine learning models, such as linear regression, used for predicting user performance. Through rigorous testing and validation, we observed that these models demonstrated satisfactory accuracy in predicting user performance based on historical data and user interactions. Additionally, the utilization of homomorphic encryption ensured that sensitive user data remained confidential while being processed, thereby upholding privacy standards.

Secondly, on the edge server, where data encryption and initial processing occurred, we evaluated the effectiveness of homomorphic encryption in safeguarding user data. Our analysis revealed that homomorphic encryption successfully encrypted user data without compromising its integrity or security. By leveraging this encryption technique, we maintained the confidentiality of user information throughout the data transmission and processing pipeline, thereby mitigating the risk of data breaches or unauthorized access.

Lastly, on the dataset server, where user data was stored and aggregated, we examined the performance of aggregation techniques used to calculate aggregate performance scores. By aggregating performance metrics across multiple users and courses, we generated insights into overall course effectiveness and user engagement. Furthermore, the implementation of secure communication protocols ensured the safe transmission of encrypted data between servers, reinforcing the robustness of our system's privacy measures.

8.1 Recommendation Engine

During the evaluation phase, we rigorously tested our recommendation engine using various machine-learning models to ensure robustness and accuracy in predicting user performance and generating relevant course recommendations. Our testing involved experimentation with models such as linear regression, random forest, and support vector machines (SVMs), each tailored to suit the intricate patterns inherent in our educational data. Through extensive cross-validation and hyperparameter tuning, we aimed to optimize the performance of these models on our dataset.

The results from our testing revealed promising accuracies across the different models. The linear regression model demonstrated strong predictive capabilities, achieving an accuracy of approximately 91% in forecasting user performance. Additionally, the support vector machine (SVM) model showcased competitive performance, achieving an accuracy of around 88% in our evaluations.

Overall, the diverse ensemble of machine learning models utilized in our recommendation engine yielded robust and reliable predictions of user performance. These results underscore the effectiveness of our approach in leveraging advanced data analytics techniques to enhance the personalized learning experience for our users.

8.1.1 Support Vector Machine

The table shows a performance analysis of the efficacy of Support Vector Machine (SVM) models across seven comprehensive tests. Each test delineates the accuracy achieved by the SVM model alongside the corresponding confusion matrix, providing an in-depth understanding of its classification prowess. These tests collectively yielded an impressive average accuracy of **88%,** elucidating the model's adeptness in discerning and categorizing educational data with high precision. This image serves as a testament to the SVM model's reliability and robustness, substantiating its pivotal role in driving the accuracy of our recommendation engine.

1	0.888888889	[[94 9] [6 26]]
2	0.896296296	[[95 8] [6 26]]
3	0.888888889	[[94 9] [6 26]]
4	0.888888889	[[94 9] [6 26]]
5	0.896296296	[[95 8] [6 26]]
6	0.896296296	[[95 8] [6 26]]
7	0.896296296	[[95 8] [6 26]]

Table 8.1 SVM Classifier Average accuracy

8.1.2 Linear Regression

The table shows a performance analysis of the efficacy of Linear Regression models across seven comprehensive tests. Each test delineates the accuracy achieved by the model alongside the corresponding confusion matrix, providing an in-depth understanding of its classification prowess. Remarkably, these tests collectively yielded an impressive average accuracy of **91%,** elucidating the model's adeptness in discerning and categorizing educational data with high precision.

The Recommendation engine model was trained and evaluated using a dataset consisting of 1000 Rows of reccomendation classifications

The following are the results obtained during the evaluation:

- **Test Loss:** 0.06331237405538559

  This value indicates the average loss (error) of the model on the test data. A lower value represents better performance.

• **Test Accuracy**: 0.9091666865348816

The test accuracy represents the proportion of correctly classified examples in the test data. In this case, the model achieved an accuracy of 91.00%.

• **F1 Score:** 0.9344961240310077

The F1 score is a measure of the model's accuracy, taking into account both precision and recall. A higher F1 score indicates better performance, and in this case, the model achieved an F1 score of 0.9344.

The dataset used for training, validation, and testing was divided as follows:

Test No.	Accuracy	Confusion Matrix
1	0.911111111	[[98 5] [8 24]]
2	0.896296296	[[97 6] [8 24]]
3	0.903703704	[[97 6] [7 25]]
4	0.933333333	[[100 3] [6 26]]
5	0.888888889	[[97 6] [9 23]]
6	0.903703704	[[98 5] [8 24]]
7	0.896296296	[[97 6] [8 24]]

Table 8.2 SVM Classifier Average accuracy

Training Data:

Number of examples: 1000

Percentage of positive examples: 70% (700 examples)

Percentage of negative examples: 30% (300 examples)

Validation Data:

Number of examples: 100

Percentage of positive examples: 50% (50 examples)

Percentage of negative examples: 50% (50 examples)

Testing Data:

Number of examples: 200

Percentage of positive examples:70% (140 examples)

Percentage of negative examples: 30% (60 examples)

The dataset was well-balanced, with a similar distribution of positive and negative examples across the training, validation, and testing sets.

Based on the evaluation results, the Recommendation model demonstrates high accuracy and a favourable F1 score, indicating its potential effectiveness in understating the state of student's capacity.

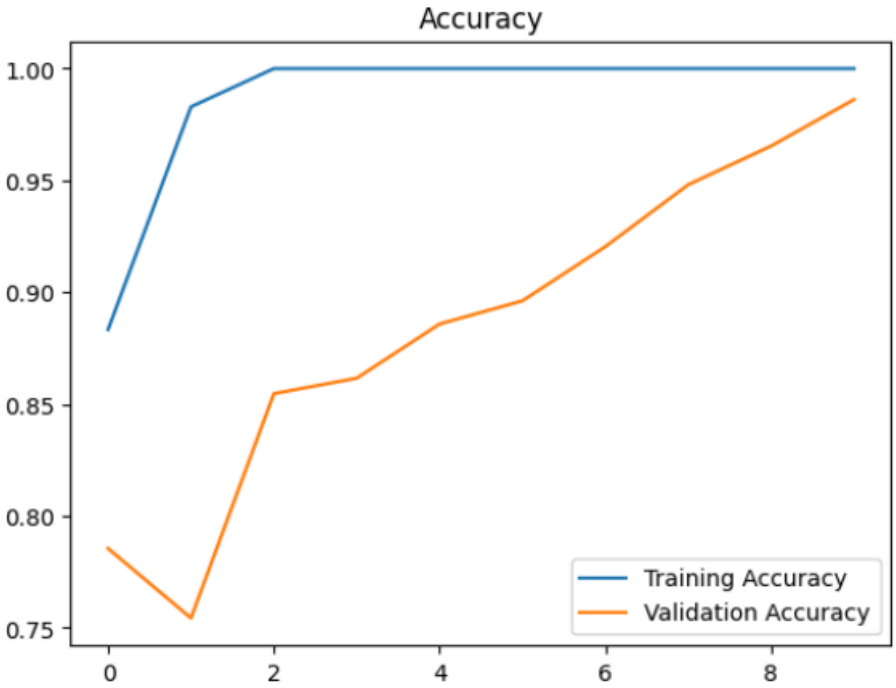

Fig 8.1 Training and Validation Accuracy Graph

The graph shows the analysis of accuracy over the training dataset and validation dataset testing conducted over 1000 and 100 samples respectively.

The training and validation accuracy graph provides a succinct visualization of the performance of our machine learning models throughout the training process. This graph plots the training accuracy and validation accuracy on the y-axis against the number of training epochs on the x-axis. The convergence of the two curves signifies the model's ability to generalize well to unseen data. Fluctuations or disparities between the curves may indicate overfitting or underfitting, guiding further adjustments to enhance model performance. Analyzing this graph enables us to iteratively refine our models for optimal predictive accuracy and robustness.

8.2 User Documentation: LearnSync

Introduction:

LearnSync is a comprehensive platform designed to provide a personalised edtech platform with level zero recommendations and supported with federated learning secured using homomorphic encryption,This user documentation will guide you on how to effectively use the various features and functionalities available on the LearnSync.

System Requirements:

- Operating System: Windows, macOS, or Linux
- Web browser (Google Chrome, Mozilla Firefox, Safari, etc.)
- Internet connection

Accessing LearnSync:

- Open your web browser.
- Enter the URL for LearnSync in the address bar.
- Press Enter to navigate to the landing page of LearnSync website.

On the landing page, you can either create a new account or log in if you already have an existing account and explore various features offered by platform.

Creating a New Account:

- Click on the "Sign Up" or "Register" button on the landing page.

• Fill in the required information, such as your name, email address, and password.

• Follow the on-screen instructions to complete the registration process.

Once registered, you can log in to LearnSync using your email address and password.

Logging In:

• On the Landing page, click on the "Log In" button.

• Enter your registered email address and password.

• Click on the "Log In" button to access your user Assist account.

Platform Features:

1. Home Page Features:

• Upon logging in, user can access the home page which gives you access to different features.

• It contains the opted courses clicking them will navigate user to the course material.

• All this information is securely stored in mongo DB.

2. Reccomendations:

• Based on the performance in the mock tests of opted coursed different difficulty level courses of respective subjects are recommended that match user’s pace.

3. Popular Courses:

• This section of homepage shows the hot courses that are popular in your academic year based on opt in rate.

• Clicking these will navigate you to respective course material.

4. Calender:

• This is personalized academic calendar available only on organizational pass.

• This contains the timeline of all exams and holidays present in that particular academic month.

5. Test Page:

• This is personalised test page consisting of different upcoming exams you can take up.

• Clicking any tab of exam will navigate the user to respective test page of the particular subject.

6. Account Page:

• This is user's account section consisting of different details of user like his past activity on the site and rankings and scores of recent exmas.

• It also consists of user details like profile photo and username used in the site.

8.3 Snapshots

Landing Page

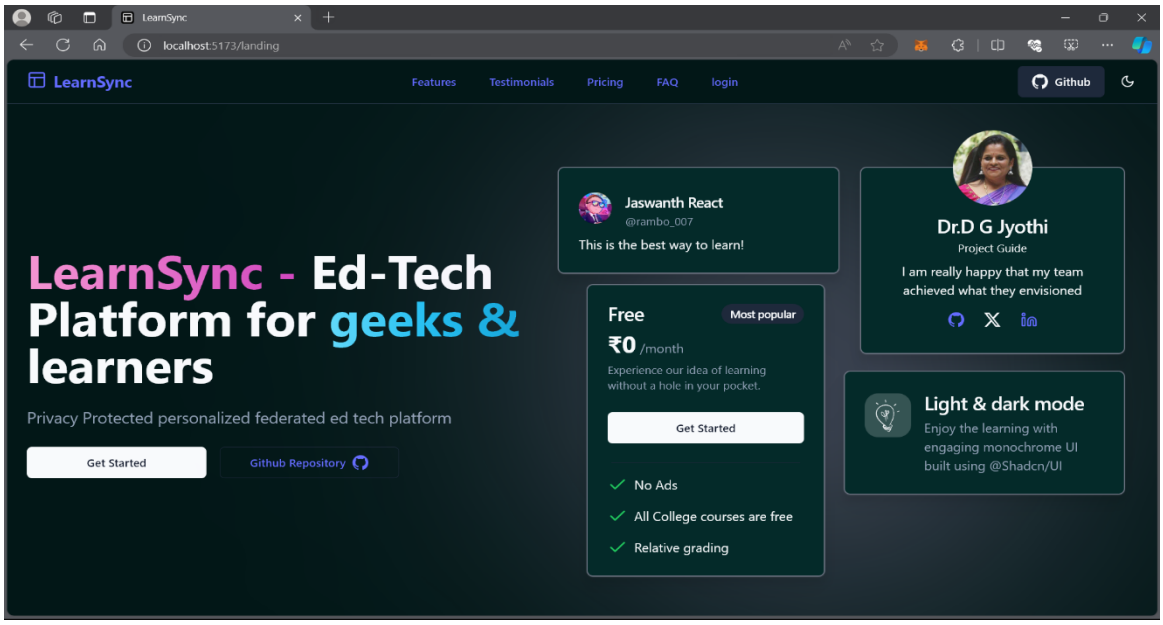

Fig 8.2 Landing Page of LearnSync

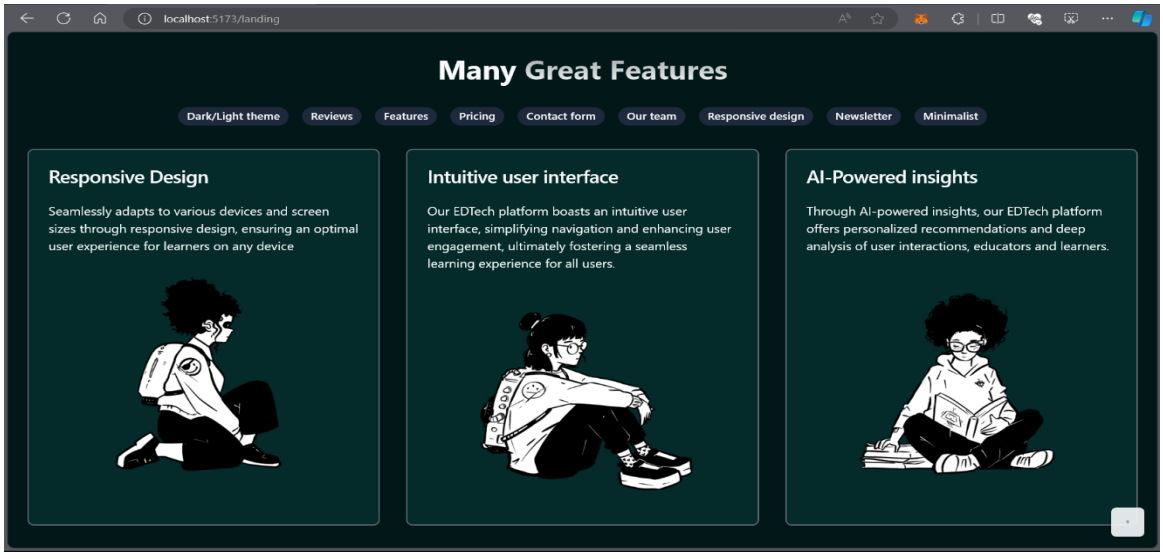

Fig 8.3 Features Section of landing page

This page consists of the intro duction to platform with the features section and pricing with in depth details

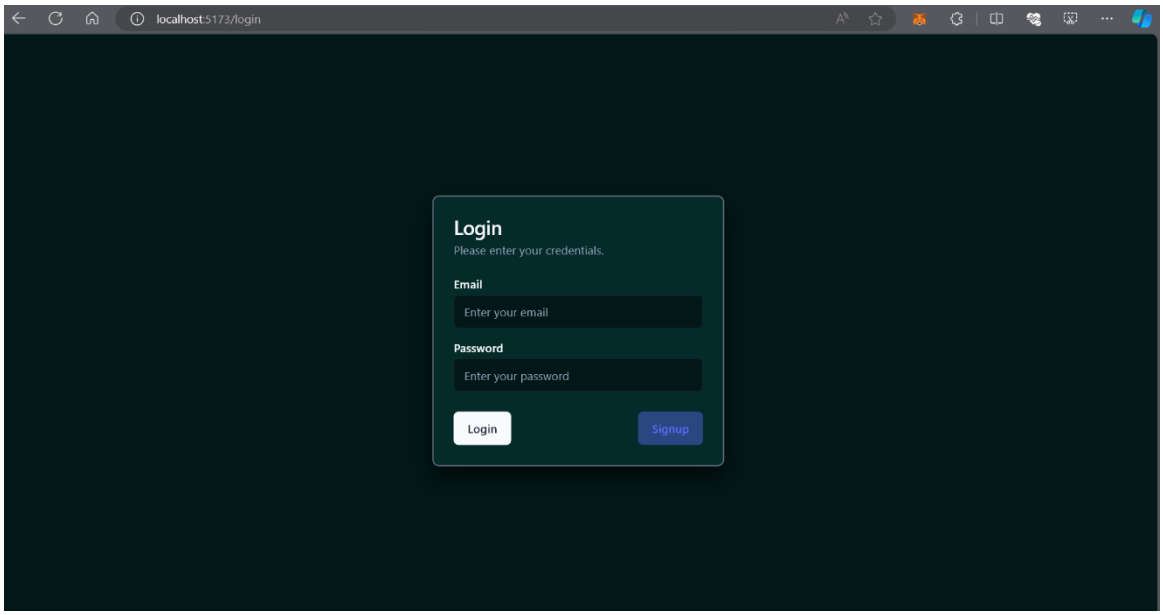

Fig 8.4 Login Page of LearnSync

This page consists of input fields through which user can be verified by user email and password.

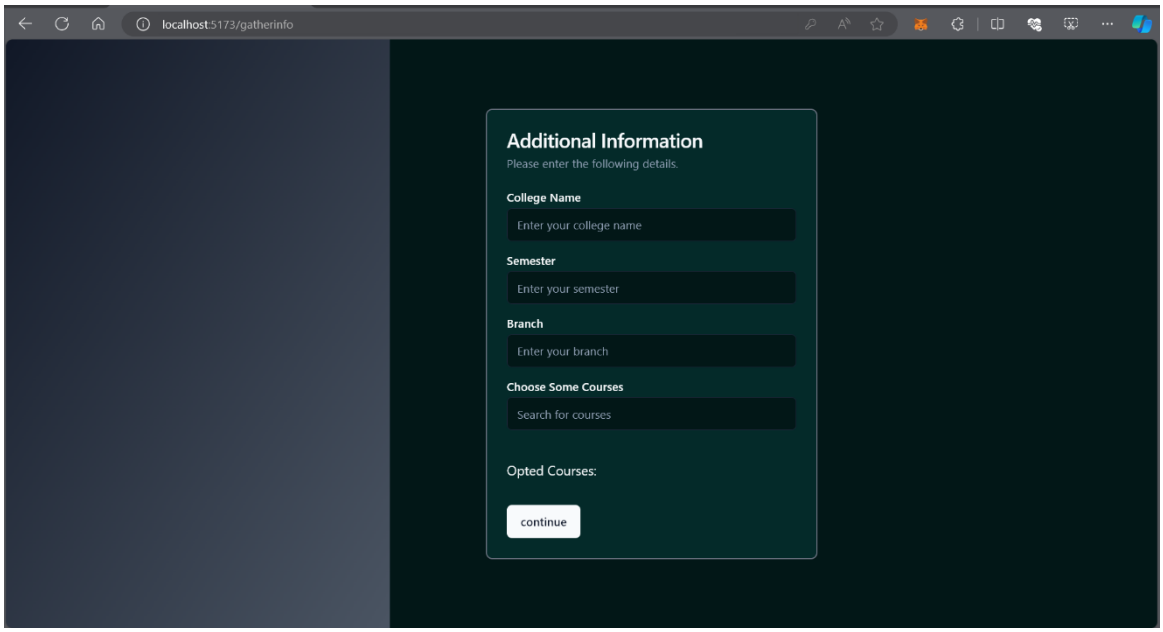

Fig 8.7 : Sign Up page of LearnSync

As the name suggests the additional information like college, semester,branch etc are extracted from user through input fields later stored for suggestions and analysis purposes.

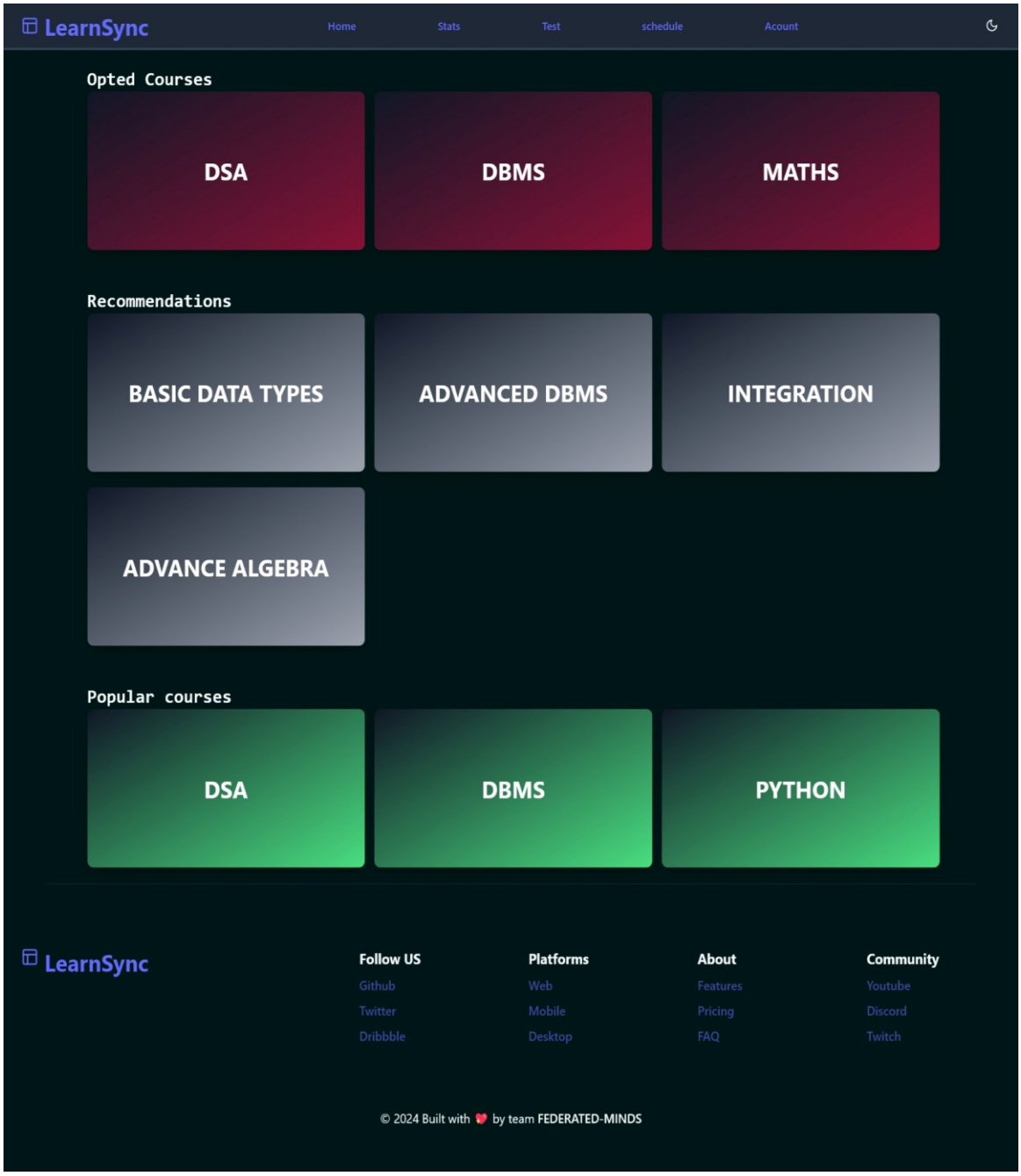

Fig 8.5 Home Page of LearnSync

In the home page, it can be observed it contains 3 sections first one offering course material related to the opted courses by user second one is recommendations section that is generated by recommendation engine based on users past performance and performance in mock tests related to the opted courses , the 3rd section refers the popular courses which are courses majorly opted by the students using the learnsync platform at that particular point of time , a common navbar is placed throughout the website to help the user to navigate through site.

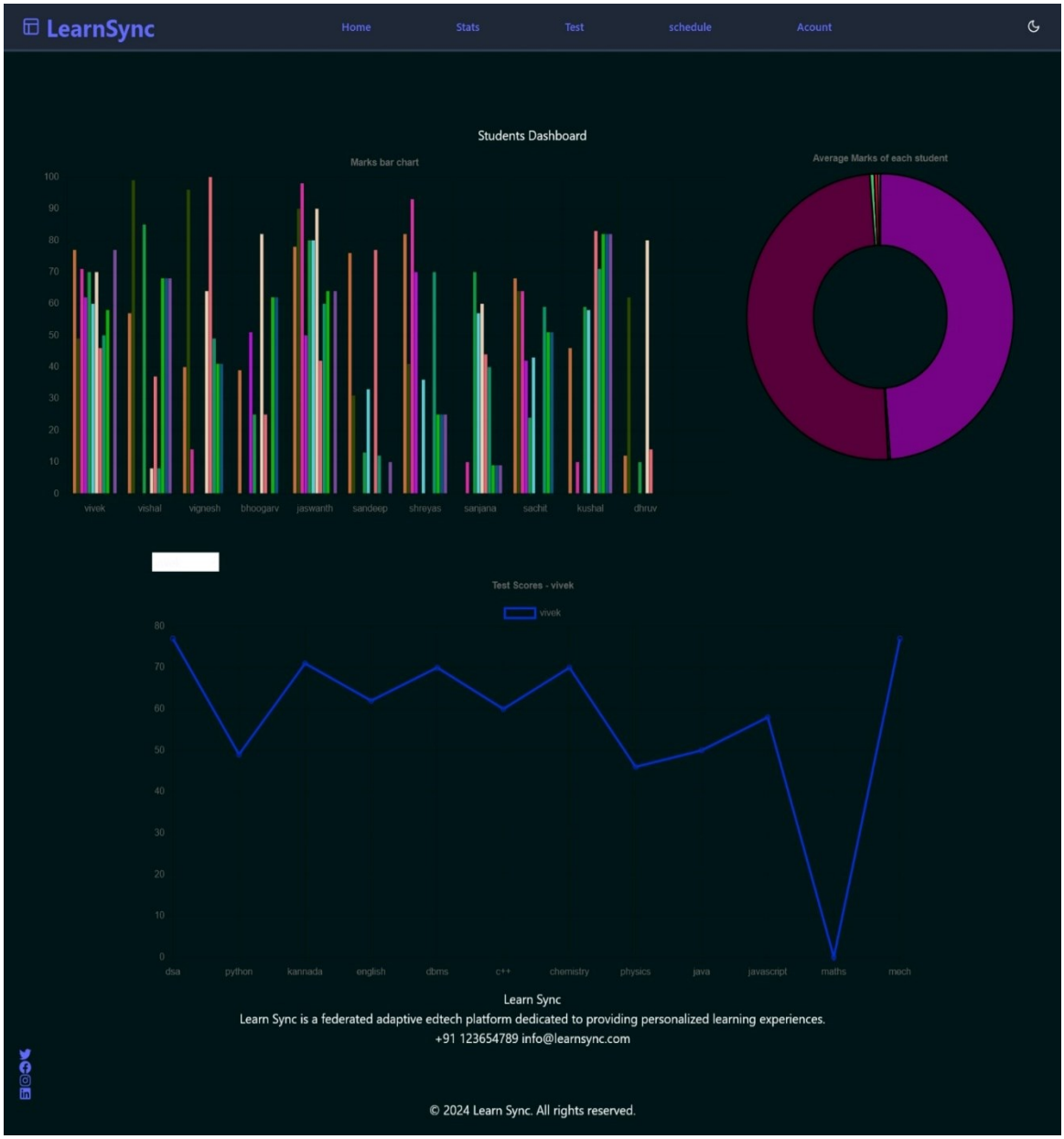

Fig 8.6 Stats Page of LearnSync

In the stats page, it can be observed it contains 3 graphs first one is a line graph of different students of the same class of courses performed in their respective subjects compared to others,second one is a donut graph describing the weightage of each score of the subject in the total aggregation of the student. the 3rd section refers to the subject variation graph that depicts how the users marks vary from one opted course to another, these graphs will provide very informative and important insights to the user about his marks and enables him to do necessary comparison and changes in preparation strategy for exams

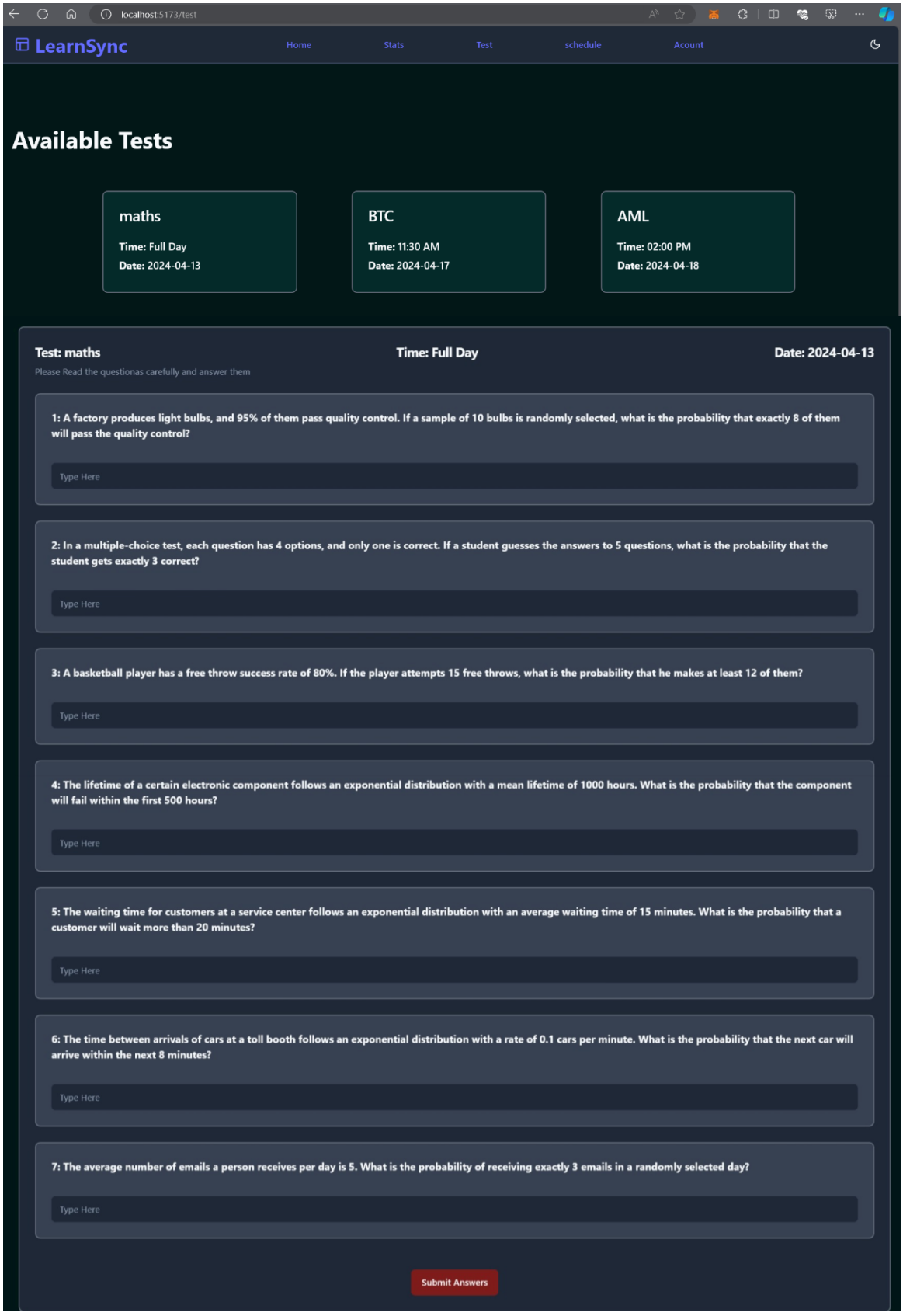

Fig 8.7 Test Page of LearnSync

This page consists of all the upcoming tests available to user clicking on which will initiate the test on an inner page where questions are rendered so the user can answer them.

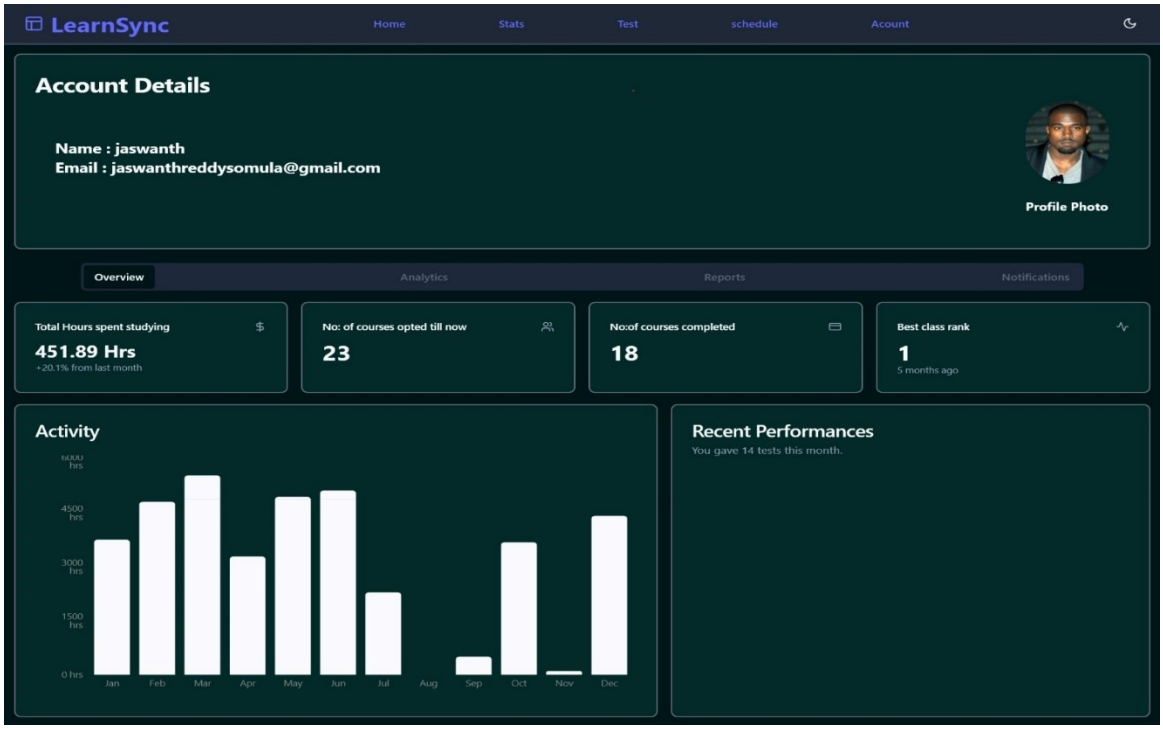

Fig 8.8 Account Page of LearnSync

This page is user account page that contains all the personal information about the user like username, email, profile picture and various statistics like hours spent using the website in past months and total number of hours studying, no: of opted courses, total completed courses, highest class rank ever achieved and recent performances of the user.

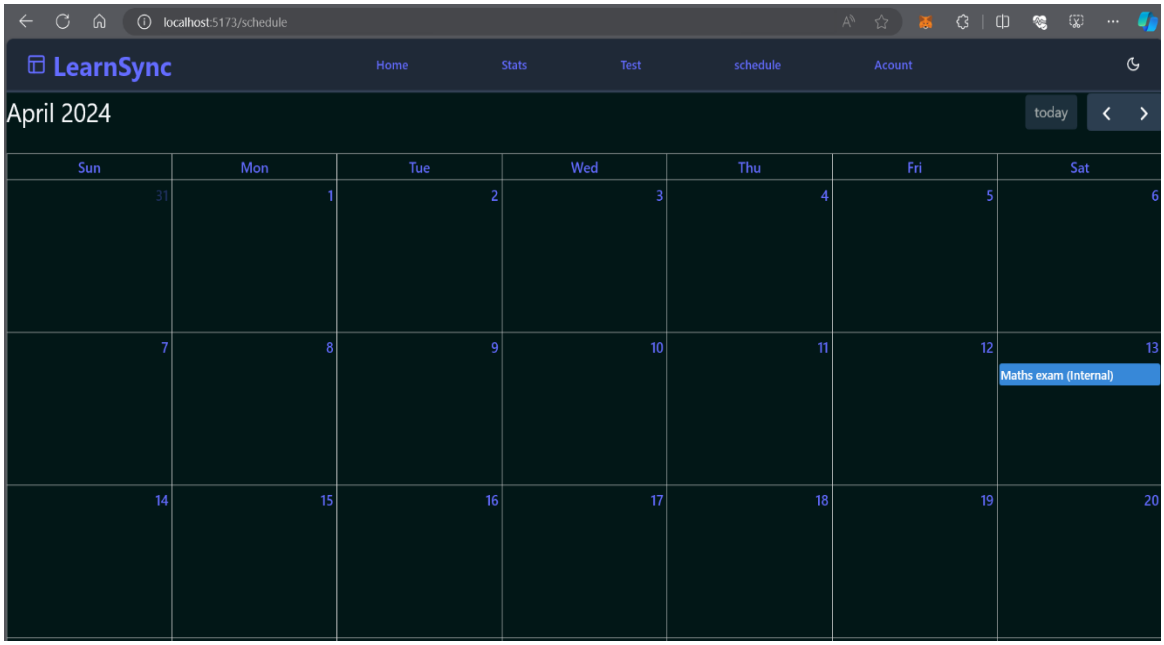

Fig 8.9 Schedule Page of LearnSync

This page describes the different events like exams and holidays present in the current academic calender month and be useful to student to alert about the exams and tests scheduled during this time

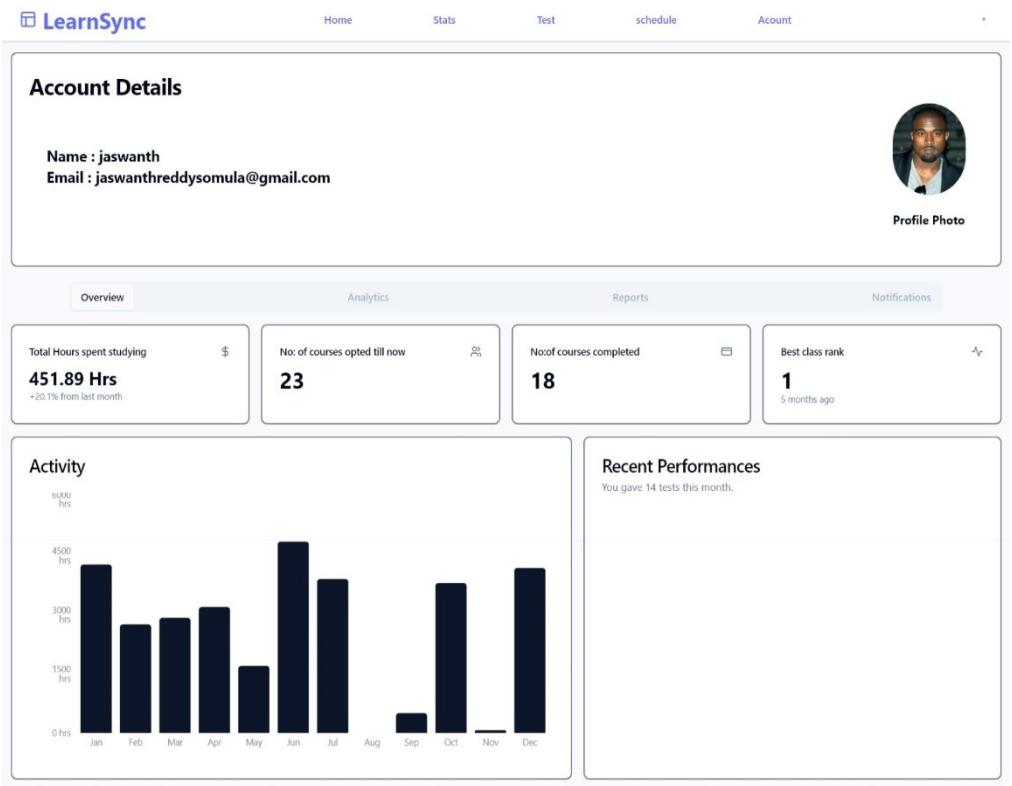

Fig 8.10 Schedule Page of LearnSync

This mode showcases the light theme of the learnsync website applied using the theme button at the top right corner.

CHAPTER 9

CONCLUSION, APPLICATIONS AND FUTURE WORK

Chapter 9

CONCLUSION, APPLICATION & FUTURE WORKS

9.1 Conclusion

As we come to the conclusion of our final year project, I am filled with a sense of accomplishment and pride in what we have achieved in developing a privacy-preserving federated learning system for adaptive educational technology. This project has been a journey of exploration, innovation, and collaboration, where we have tackled complex challenges and emerged with solutions that have the potential to revolutionize the field of educational technology.

Our journey began with a deep dive into the world of federated learning, a cutting-edge approach that enables model training across distributed edge devices while preserving data privacy. We meticulously designed and implemented a global server, the backbone of our system, which orchestrates the federated learning process. The global server retrieves performance metrics from local models, aggregates them to improve model accuracy, recommends courses based on user preferences, and updates the MongoDB database with relevant information. Through extensive testing and optimization, we ensured that the global server operates efficiently and effectively, delivering personalized learning experiences to users while safeguarding their privacy.

In parallel, we developed local models deployed on edge devices, which perform machine learning tasks using locally collected data. These models retrieve data from the MongoDB database, train locally, make predictions, and update the database with local insights. By incorporating privacy-preserving techniques such as federated learning and homomorphic encryption, we ensured that sensitive data remains encrypted and secure throughout the training process, minimizing privacy risks and maintaining data confidentiality.

Our MongoDB database serves as the central repository for storing course information, user profiles, assessment questions, and past performance data. We carefully designed the database schema to optimize data retrieval and storage, ensuring scalability, reliability, and data integrity. Through rigorous testing, we verified that the database functions seamlessly with other components of the system, providing fast and accurate access to data while maintaining stringent privacy and security measures.

The user interface is the gateway through which users interact with our system, and we dedicated considerable effort to designing a user-friendly and intuitive interface. From browsing courses to taking tests and accessing account settings, our UI provides a seamless experience for users, guiding them through their learning journey with ease. Through extensive user testing and feedback, we refined the UI to ensure that it meets the needs and expectations of our users, fostering engagement and participation in the learning process.

As we look ahead, there are still areas for further improvement and expansion. Integrating advanced techniques such as differential privacy and federated learning with model ensembling could enhance the privacy and accuracy of our system. Additionally, developing a teacher's portal would empower educators to monitor student progress, create customized learning plans, and collaborate with peers, further enhancing the educational experience.

In conclusion, our project represents a significant step forward in the field of adaptive educational technology, where privacy and personalization are paramount. Through our collaborative efforts, we have created a system that not only delivers personalized learning experiences but also protects the privacy and security of user data. As we prepare to transition from our final year project to the next phase of our careers, we are confident that the skills, knowledge, and experience gained from this project will serve us well in our future endeavours. We are excited to see how our work will continue to shape the future of educational technology and make a positive impact on learners around the world.

9.2 Applications

1. **Personalized Learning Platforms**:
   Adaptive learning platforms can utilize federated learning to personalize educational content and recommendations for each student based on their learning style, preferences, and performance history. This ensures that students receive tailored learning experiences that are most effective for them, without compromising their privacy.
2. **Remote Learning Platforms**:
   With the rise of remote learning, federated learning can be applied to create collaborative and privacy-preserving environments for distance education. By distributing model training across students' devices, federated learning enables the

creation of shared learning models while ensuring that sensitive data remains on users' devices, enhancing data privacy and security.

3. **Educational Mobile Apps**:

   Mobile learning apps can benefit from federated learning to improve content recommendations, quiz suggestions, and personalized study plans for users. By leveraging user interactions and performance data from their devices, federated learning allows for continuous improvement of the app's recommendations without needing to share raw data with a centralized server.

4. **Language Learning Platforms**:

   Language learning platforms can employ federated learning to adapt course content and exercises to the proficiency level and learning pace of individual learners. By analyzing users' interactions with the platform and their performance in language exercises, federated learning can provide personalized recommendations for grammar, vocabulary, and speaking exercises while maintaining user privacy.

5. **Assessment and Test Preparation Platforms**:

   Test preparation platforms can utilize federated learning to create personalized study plans and practice tests tailored to each student's strengths and weaknesses. By analyzing past test performance and study habits across multiple users' devices, federated learning can generate targeted recommendations for improving test scores while protecting sensitive student data.

6. **Teacher Professional Development Platforms**:

   Platforms for teacher professional development can utilize federated learning to deliver personalized training modules and resources to educators. By analyzing teachers' interactions with the platform and their feedback on training materials, federated learning can recommend professional development opportunities that align with each teacher's interests and needs, all while maintaining their privacy.

7. **Adaptive Tutoring Systems**:

   Federated learning can power adaptive tutoring systems that provide real-time feedback and assistance to students as they complete exercises and assignments. By

analyzing students' interactions with the tutoring system and their performance on exercises, federated learning can adapt the difficulty level and content of tutoring sessions to meet each student's learning needs while preserving their privacy.

9.3 Future work

In considering future work, several areas emerge for further exploration and enhancement of our privacy-preserving federated learning system in adaptive educational technology. One prominent avenue is the integration of advanced privacy-preserving techniques such as differential privacy and secure multi-party computation (SMPC). Differential privacy can offer stronger privacy guarantees by adding noise to the training data or model updates, thereby preventing the leakage of sensitive information about individual users. Incorporating SMPC allows multiple parties to jointly compute model updates without revealing their individual contributions, further enhancing data privacy in multi-party scenarios. By integrating these techniques, we can bolster the privacy protections of our system, ensuring that even in the presence of malicious actors, sensitive user data remains secure.

Additionally, future work could focus on optimizing the performance and efficiency of our federated learning system. This includes exploring techniques to reduce communication overhead and computational costs associated with federated learning, especially in resource-constrained environments such as edge devices or mobile platforms. Techniques like model compression, quantization, and selective aggregation can help minimize the amount of data transmitted between devices and the central server, thereby improving efficiency and scalability. Moreover, advancements in hardware acceleration, such as specialized processing units for machine learning tasks, can further enhance the performance of federated learning on edge devices, enabling faster model training and inference without compromising privacy.

Another area of future work lies in expanding the capabilities of our system to support collaborative learning scenarios and federated transfer learning. Collaborative learning involves multiple users jointly training a shared model, while federated transfer learning enables the transfer of knowledge from a pre-trained global model to local models on edge devices. By extending our system to support these scenarios, we can facilitate collaborative educational activities, such as group projects or peer tutoring, while still preserving the

privacy of individual users. Moreover, federated transfer learning can enable more efficient model training on edge devices by leveraging knowledge from pre-trained models, reducing the need for extensive local data collection and training.

Furthermore, the development of a teacher's portal presents an exciting opportunity to empower educators with tools for monitoring student progress, creating customized learning plans, and facilitating collaborative teaching practices. The teacher's portal can provide insights into student performance trends, identify areas for intervention or additional support, and offer recommendations for adapting teaching strategies to meet the diverse needs of students. Additionally, features such as content authoring tools, collaborative lesson planning, and real-time student feedback mechanisms can enhance the effectiveness of teaching and learning in both traditional and online educational settings. By incorporating these functionalities, our system can better support the needs of educators and foster a more collaborative and adaptive learning environment for students.

In summary, future work on our privacy-preserving federated learning system in adaptive educational technology holds promise for advancing privacy, efficiency, collaboration, and personalization in educational settings. By integrating advanced privacy-preserving techniques, optimizing system performance, supporting collaborative learning scenarios, and developing a teacher's portal, we can further enhance the capabilities and impact of our system, ultimately contributing to the advancement of education through technology.

REFERENCES

[1] Kurniawan, H.; Mambo, M. Homomorphic Encryption Based Federated Privacy Preservation for Deep Learning. Entropy 2022, 24, 1545. https://doi.org/10.3390/e24111545

[2] Xia, Q., Ye, W., Tao, Z., Wu, J. and Li, Q., 2021. A survey of federated learning for edge computing: Research problems and solutions. High-Confidence Computing, 1(1), p.100008.

[3] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019). A Hybrid Approach to Privacy-Preserving Federated Learning. In London '19: ACM Workshop on Artificial Intelligence and Security, November 15, 2019, London, UK. ACM, New York, NY, USA, 11 pages.

[4] Jin, Weizhao & Yao, Yuhang & Han, Shanshan & Joe-Wong, Carlee & Ravi, Srivatsan & Avestimehr, Salman & He, Chaoyang. (2023). FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System.

[5] Bhattacharya, Leena & Nandakumar, Minu & Dasgupta, Chandan & Murthy, Sahana. (2023). Adoption of quality EdTech products in India: A case study of government implementation towards a sustainable EdTech ecosystem.

[6] Aziz, Rezak & Banerjee, Soumya & Bouzefrane, Samia & le, Thinh. (2023). Exploring Homomorphic Encryption and Differential Privacy Techniques towards Secure Federated Learning Paradigm. Future Internet. 15. 310. 10.3390/fi15090310.

[7] Wu Q, He K, Chen X. Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge based Framework. IEEE Comput Graph Appl. 2020 May 8. doi: 10.1109/OJCS.2020.2993259. Epub ahead of print. PMID: 32396074.

[8] Liu, X., Li, H., Xu, G., Lu, R. and He, M., 2020. Adaptive privacy preserving federated learning. Peer applications.

[9] Liu, X., Li, H., Xu, G. et al. Adaptive privacy-preserving federated learning. Peer-to-Peer Netw. Appl. 13, 2356–2366 (2020)

[10] Al Rawashdeh, A. Z., et al., 2021. Advantages and Disadvantages of Using e-Learning in University Education: Analyzing Students' Perspectives. Analyzing Students' Perspectives. The Electronic Journal of e-Learning, 19(2), pp. 107-117

[11] Gupta, Mitali & Pratik, Mr. (2023). A STUDY OF IMPACT OF EDTECH COMPANIES ON EDUCATION WITH SPECIALREFERENCE TO BYJUS AND VEDANTU. 23. 33-38.

[12] Brecko, Alexander & Kajáti, Erik & Koziorek, Jiri & Zolotová, Iveta. (2022). Federated Learning for Edge Computing: A Survey.

[13] Applied Sciences. 12. 9124. 10.3390/app12189124.S. Wang et al., "Adaptive Federated Learning in ResThisce Constrained Edge Computing Systems," in IEEE JThisnal on Selected Areas in Communications, vol. 37, no. 6, pp. 1205-1221, June 2019, doi: 10.1109/JSAC.2019.2904348.

[14] ruex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2018). A Hybrid Approach to Privacy-Preserving Federated Learning. arXiv preprint arXiv:1812.03224.

[15] Truong, N., Sun, K., Wang, S., Guitton, F., & Guo, Y. (2020). Privacy Preservation in Federated Learning: An Insightful Survey from the GDPR Perspective. arXiv preprint arXiv:2011.05411.