GOVERNMENT POLYTECHNIC COLLEGE PERUMBAVOOR

Koovappady P.O Ernakulam-683 544 Kerala

Semester - VI

Computer Engineering 2022-23

PROJECT REPORT

on

ACCREDIFY

AN ACCREDITATION MANAGEMENT SYSTEM

Submitted by

VISHNU K.B

20130131

GOVERNMENT POLYTECHNIC COLLEGE
PERUMBAVOOR
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **Accredify** submitted by **Vishnu K.B**.The project report is approved for submission requirement for 6009 -Project and Seminar in 6^{th} semester Computer Engineering at Govt.Polytechnic College,Perumbavoor.

Head of Section **Lecturer in Charge**

Date :

Place :

Internal Examiner **External Examiner**

ACKNOWLEDGMENT

I would like to express my sincere gratitude to all those who have contributed to the successful completion of my project Accredify - The Accreditation Management Application. This opportunity has allowed me to delve into a fascinating field of web development and expand my knowledge in computer engineering.

First and foremost, I extend my heartfelt appreciation to Dr. Aiju Thomas, the Principal of Government Polytechnic College Perumbavoor. I am grateful for his constant support, guidance, and encouragement throughout my academic journey. His visionary leadership and commitment to excellence have created an environment conducive to learning and exploration.

I am indebted to Mr. Biju Peter, the Head of the Department of Computer Engineering, for his invaluable guidance and mentorship. His expertise, patience, and enthusiasm have been instrumental in shaping my understanding of the subject and honing my research skills. I am thankful for his unwavering support and valuable insights that have enriched my seminar.

I would also like to acknowledge the faculty members of the Computer Engineering Department for their constant support and encouragement throughout my academic journey. Their expertise and passion for teaching have played a significant role in shaping my intellectual growth.

I would like to express my gratitude to my classmates and team members for their support and encouragement. Their presence and collaboration have made the learning experience enjoyable and memorable.

Thank you all for your unwavering support and belief in me.

Contents

1 Introduction **4**

2 Problem Statement **6**

2.1 Manual and Time-Consuming Process 6

2.2 Lack of Clarity and Guidance 6

2.3 Limited Self-Assessment Capabilities 6

2.4 Insufficient Data Analysis 6

2.5 Communication and Collaboration Challenges 7

2.6 Lack of Advanced Consultation 7

3 Solutions **8**

3.1 Streamlined Accreditation Process 8

3.2 Comprehensive Guidelines and Standards 8

3.3 Self-Assessment Tools 8

3.4 Data Visualization and Analysis 8

3.5 Collaboration and Communication Platform 9

3.6 Advanced Consultation Services 9

4 Requirement Analysis **10**

4.1 Computer Specifications 10

4.2 Operating System 10

4.3 Node.js 10

4.4 Dependencies and Libraries 10

4.5 Development Tools 11

4.6 Internet Connectivity 11

5 Technologies used **12**

5.1 MongoDB 12

5.2 Express.js 12
5.3 React.js 13
5.4 Node.js 13
5.5 Razorpay 13
5.6 Git 14

6 Code **15**
6.1 Sign up 15
6.2 Login 15
6.3 Validation 16
6.4 Razorpay 17
6.5 Email Automation 18
6.6 Fetch department details 19
6.7 Update department details 19

7 Screenshots **21**
7.1 Home Page 21
7.2 Login 21
7.3 Sign Up 22
7.4 Pre Qualifiers 22
7.5 Premium 22
7.6 Razorpay 23
7.7 Invoice Bill 23
7.8 Admin Dashboard 23

8 Conclusion **26**

9 References **27**

Chapter 1

Introduction

Accreditation plays a crucial role in assessing and ensuring the quality and standards of educational institutions. However, the accreditation process can be complex and time-consuming, requiring meticulous evaluation and compliance with various standards and criteria.

To address these challenges, Accredify web application provides a platform where college departments can easily navigate the accreditation process. Departments can create individual accounts and access a range of features and resources designed to support their accreditation journey. The system offers a structured assessment framework that enables departments to evaluate their readiness for accreditation. By providing a comprehensive set of criteria and guidelines, the system assists departments in identifying areas of improvement and implementing necessary measures to meet accreditation standards.

A key aspect of the system is the integration of pricing and consultation services. Departments that opt for a paid subscription gain access to advanced consultation services, where experienced professionals provide personalized guidance and support. This additional layer of expertise ensures that departments receive tailored assistance in addressing specific challenges and enhancing their accreditation preparedness.

Furthermore, the system incorporates an admin portal that empowers administrators to gain valuable insights into departmental data. Administrators can visualize key metrics such as admissions and graduations per year, enabling them to monitor performance trends and make informed decisions to further improve the quality of education within the institution.

Overall, the Accreditation Management System aims to simplify and streamline the accreditation process for college departments. It promotes collaboration,

provides valuable resources, and offers expert guidance to ensure that departments are well-equipped to meet accreditation standards. By leveraging the power of the MERN stack, the system delivers a user-friendly and efficient solution that contributes to the pursuit of academic excellence within educational institutions.

Chapter 2

Problem Statement

The Accreditation Management System aims to address several challenges and problems commonly faced by college departments during the accreditation process. Some of the key problems that this application seeks to solve include:

2.1 Manual and Time-Consuming Process

Traditional accreditation processes often involve extensive paperwork, manual data entry, and complex administrative tasks. This application automates and streamlines these processes, reducing the time and effort required for accreditation.

2.2 Lack of Clarity and Guidance

Many college departments struggle with understanding the accreditation standards and requirements. The application provides a clear framework and guidelines, helping departments assess their readiness and take necessary steps for compliance.

2.3 Limited Self-Assessment Capabilities

Without proper tools and resources, departments may struggle to conduct a comprehensive self-assessment of their strengths and weaknesses. The application offers self-assessment features, enabling departments to identify areas of improvement and develop actionable plans.

2.4 Insufficient Data Analysis

Analyzing departmental data, such as admissions and graduations, is crucial for identifying trends and making informed decisions. The application's data visualization and analysis capabilities help administrators gain valuable insights into

departmental performance and identify areas for enhancement.

2.5 Communication and Collaboration Challenges

Coordinating accreditation-related tasks and fostering collaboration among department members can be challenging. The application provides a centralized platform for communication, document sharing, and task management, promoting efficient collaboration and seamless information exchange.

2.6 Lack of Advanced Consultation

Departments may require expert guidance and consultation to navigate complex accreditation requirements. The application offers advanced consultation services, providing specialized support and personalized advice to assist departments in meeting accreditation standards.

Chapter 3

Solutions

The Accredify offers a range of solutions to address the challenges faced by college departments during the accreditation process. Some of the key solutions provided by this application include:

3.1 Streamlined Accreditation Process

The application simplifies and automates various accreditation-related tasks, such as document submission, data collection, and assessment. This streamlines the overall process, reducing manual effort and improving efficiency.

3.2 Comprehensive Guidelines and Standards

The application provides departments with clear and up-to-date accreditation guidelines and standards. It helps departments understand the requirements, criteria, and benchmarks they need to meet for successful accreditation.

3.3 Self-Assessment Tools

The application offers self-assessment features that enable departments to evaluate their performance against accreditation standards. It provides assessment questionnaires, checklists, and performance indicators to help departments identify their strengths and areas for improvement.

3.4 Data Visualization and Analysis

The application includes data visualization and analysis tools that allow administrators to view and analyze departmental data, such as admissions, graduations,

and performance metrics. This helps in identifying trends, patterns, and areas that require attention.

3.5 Collaboration and Communication Platform

The application serves as a centralized platform for departments to collaborate, communicate, and share documents related to accreditation. It facilitates seamless communication among department members, streamlines document management, and ensures easy access to relevant information.

3.6 Advanced Consultation Services

Departments that opt for the paid services receive advanced consultation from accreditation experts. This includes personalized guidance, advisory support, and assistance in meeting accreditation requirements effectively.

Chapter 4

Requirement Analysis

4.1 Computer Specifications

- A computer with a minimum of 8 GB RAM is recommended to ensure smooth performance during development and execution of the project.
- Adequate processing power and storage capacity are required to handle the project files and related dependencies.

4.2 Operating System

- The project can be developed and executed on various operating systems, including Windows, macOS, or Linux, as long as the necessary software dependencies are supported.

4.3 Node.js

- Node.js runtime environment is a prerequisite for running the project. It should be installed on the computer.
- The recommended version of Node.js may vary depending on the specific project requirements and dependencies. Please refer to the project documentation or requirements for the appropriate Node.js version.

4.4 Dependencies and Libraries

- The project may rely on specific libraries, frameworks, or packages. These dependencies should be installed according to the project's requirements.
- It is essential to ensure that the required versions of these dependencies are compatible with the project's codebase and specifications.

4.5 Development Tools

- An integrated development environment (IDE) or a text editor of your choice should be available to facilitate code editing and development.
- Git, a version control system, is recommended for managing source code revisions, collaborating with team members, and tracking project changes.

4.6 Internet Connectivity

- Although not mandatory for the core functionality of the project, internet connectivity may be required for installing dependencies, accessing external APIs, or utilizing cloud-based services.

It is crucial to review and fulfill these system requirements to ensure a smooth setup and execution of the project. Additionally, regularly updating the dependencies and software versions as recommended by the project maintainers will help maintain compatibility and security.

Chapter 5

Technologies used

Accredify utilizes the MERN stack for full stack web development and razorpay for payment integration.

5.1 MongoDB

MongoDB is an open-source, NoSQL database management system that is designed to handle large volumes of unstructured and semi-structured data. It utilizes a flexible document-oriented approach, storing data in JSON-like documents called BSON. This allows for easy manipulation and querying of data, providing schema flexibility and accommodating evolving data requirements. MongoDB excels in scalability, allowing for horizontal scaling by distributing data across multiple servers or shards. It also offers high performance with optimization techniques such as indexing, sharding, and in-memory caching. MongoDB supports a powerful query language with advanced filtering, sorting, and aggregation capabilities, enabling efficient data retrieval and analysis. Application uses mongodb cloud for data storage.

5.2 Express.js

Express.js is a minimalistic and flexible web application framework for Node.js. It provides a robust set of features and functionalities that simplify the development of web applications and APIs. With its unopinionated approach, Express.js allows developers to structure and organize their projects according to their specific needs and preferences. It provides a thin layer of abstraction over Node.js, making it easy to handle HTTP requests, define routes, and implement middleware.

Express.js is known for its simplicity and ease of use. It offers a straightforward API that allows developers to quickly build server-side applications with minimal boilerplate code. It supports various HTTP methods, such as GET, POST, PUT,

and DELETE, and provides a simple way to handle routing logic and parameter extraction. Middleware functions in Express.js allow developers to add additional functionality to their applications, such as authentication, logging, error handling, and request/response manipulation.

5.3 React.js

React.js is a popular JavaScript library for building user interfaces. It provides a component-based approach to web development, allowing developers to create reusable UI components and efficiently manage the state of their applications. React.js follows a declarative and efficient programming paradigm, making it easier to build complex and interactive UIs.

One of the key features of React.js is its virtual DOM (Document Object Model), which optimizes the rendering process by efficiently updating only the necessary components when the underlying data changes. This results in faster rendering and improved performance, especially for applications with dynamic and frequently changing content.

5.4 Node.js

Node.js is a runtime environment and a popular JavaScript platform that allows developers to build server-side applications. Unlike traditional server-side technologies, Node.js uses an event-driven, non-blocking I/O model, which makes it highly efficient and scalable for handling concurrent requests.

One of the key advantages of Node.js is its ability to utilize JavaScript as a unified language for both frontend and backend development. This allows developers to share code between the client and server, promoting code reusability and reducing development time. Node.js also benefits from the vast ecosystem of JavaScript libraries and frameworks, enabling developers to leverage existing tools and resources.

5.5 Razorpay

Razorpay is a popular payment gateway solution that provides seamless online payment processing for businesses. It offers a wide range of features and services to facilitate secure and efficient payment transactions.

One of the key advantages of Razorpay is its simplicity and ease of integration. It provides developer-friendly APIs and SDKs that allow businesses to integrate payment functionality into their web or mobile applications with minimal effort. With clear documentation and well-designed APIs, developers can quickly set up payment gateways and start accepting payments from customers.

5.6 Git

Git is a widely used distributed version control system (VCS) that played a crucial role in the development of our project. With Git, we were able to effectively manage and track changes to our source code throughout the development lifecycle.

One of the key advantages of Git is its ability to track changes made to files over time. This comprehensive history allowed us to easily review, revert, and analyze modifications, additions, and deletions in our codebase. By providing detailed information about who made specific changes and when, Git empowered us to collaborate efficiently and maintain a clear record of the project's progress.

Git's branching and merging capabilities were instrumental in our development workflow. We could create separate branches to work on new features, bug fixes, or experiments without affecting the stability of the main codebase. This allowed us to experiment, iterate, and test new ideas without disrupting the main development stream. When the time came to integrate these changes, Git's merging functionality enabled us to seamlessly combine different branches and ensure a smooth integration of new code.

Chapter 6

Code

6.1 Sign up

Node.js code to insert user details into mongoDB database.

```
doSignup:(user)=>{
    console.log(user);
    return new Promise(async(resolve,reject)=>{
        db.collection(collections.USER_DETAILS).insertOne(user)
        .then(()=>{
            resolve(user);
        })
        .catch((err)=>{
            reject(err);
        });
    });
}
```

6.2 Login

Node.js code to login using username or email and password.

```
doLogin:(user)=>{
        return new Promise(async(resolve,reject)=>{
            db.collection(collections.USER_DETAILS)
            .findOne({
                $or: [{ username: user.username_or_email },
                { email: user.username_or_email }]
            })
```

```
            .then((data)=>{
                if (data) {
                    bcrypt.compare(
                    user.password, data.password, (err, isMatch) => {
                      if (err) {
                        reject(err);
                      } else if (isMatch) {
                        resolve(data);
                      } else {
                        reject("Invalid Password");
                      }
                    });
                  } else {
                    reject("User not found");
                  }
            })
            .catch((err)=>{
                reject(err)
            })
        })
    }
```

6.3 Validation

Node.js code to check the username and email is already exists or not

```
checkEmail:(email)=>{
        return new Promise(async(resolve,reject)=>{
            db.collection(collections.USER_DETAILS)
            .findOne({ email }).then((data)=>{
                resolve(data)
            })
            .catch((err)=>{
                reject(err)
            })
```

```
        })
    },
    checkUsername:(username)=>{
        return new Promise(async(resolve,reject)=>{
            db.collection(collections.USER_DETAILS)
            .findOne({ username }).then((data)=>{
                resolve(data)
            })
            .catch((err)=>{
                reject(err)
            })
        })
    }
```

6.4 Razorpay

Code for razorpay payment integration

```
generateRazorpay:(user)=>{
        return new Promise((resolve,reject)=>{
            instance.orders.create({
                amount: user.price*100,
                    currency: "INR",
                    receipt: user.userId,
                  },(err,response)=>{
                      if(err){
                          console.log(err);
                          reject(err);
                      }else{
                          console.log(response);
                          resolve(response)
                        }
                    })

        })
```

```
    },
    verifyPayment:(details)=>{
        return new Promise((resolve,reject)=>{
            hmac.update(details['payment[razorpay_order_id]']
            + '|' + details['payment[razorpay_payment_id]'])
            hmac=hmac.digest('hex')
            if(hmac==details['payment[razorpay_signature]']){
                resolve()
            }else{
                reject()
            }
        })
    },
```

6.5 Email Automation

Code to send automated email to admin when departments creates account or become premium users.

```
sendMailToAdmin:(user,msg)=>{
        return new Promise(async(resolve,reject)=>{
            let transporter = nodemailer.createTransport({
                service:'gmail',
                auth: {
                    user: EMAIL,
                    pass: EMAIL_PASSWORD
                },
              });
              let info = await transporter.sendMail({
                from: user.email, // sender address
                to: EMAIL, // list of receivers
                subject: "NBA Accreditation", // Subject line
                text: msg
              });
```

```
            console.log("Message sent: %s", info.messageId);
            console.log("Preview URL: %s",
             nodemailer.getTestMessageUrl(info));
            resolve(info)
        })
    }
```

6.6 Fetch department details

```
getUserDetails:(userId)=>{
        return new Promise(async(resolve,reject)=>{
            let user2 = await db.collection(collections.USER_DETAILS)
            .findOne({_id:ObjectId(userId)})
            if(user2){
                resolve(user2)
            }
            else{
                console.log("user does not exist");
                reject(err)
            }
        })
    }
```

6.7 Update department details

```
doAddDetails:(data,userId)=>{
        return new Promise(async (resolve, reject) => {
            db.collection(collections.USER_DETAILS)
            .updateOne({ _id: ObjectId(userId) },
                {
                    $set: { details : data }
                }
            ).then((user)=>{
                console.log("added new details")
                resolve(user)
```

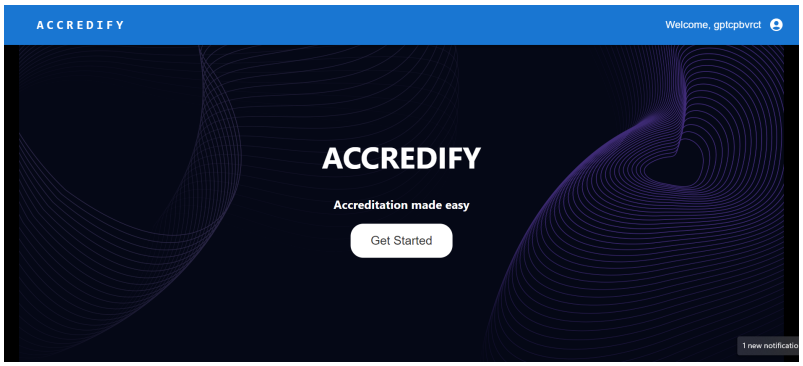
```
            })
            .catch((err)=>{
                console.log("error adding")
                reject(err)
            })
        });
    }
```

Chapter 7

Screenshots

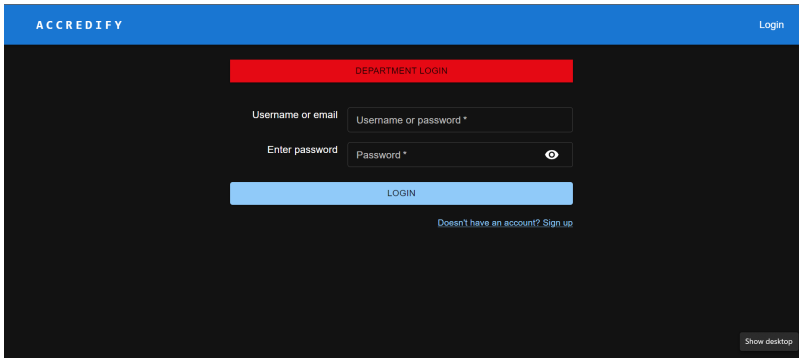
Various works have been performed for self-supervised learning using graph data. By taking an overview from these works, we can say that methods under this field can be divided into three categories:

7.1 Home Page



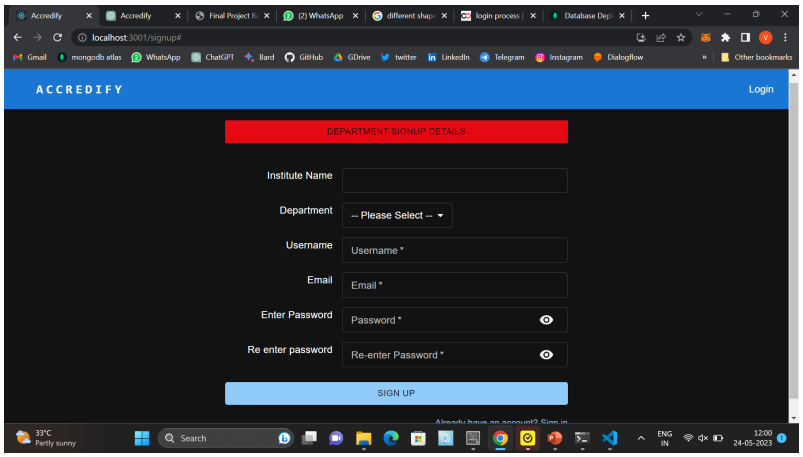
7.2 Login

The departments can login using username or email. After login department will be directed to the pre-qualifier checking page. If the department doesn't have an account, they need to sign up to use the application.



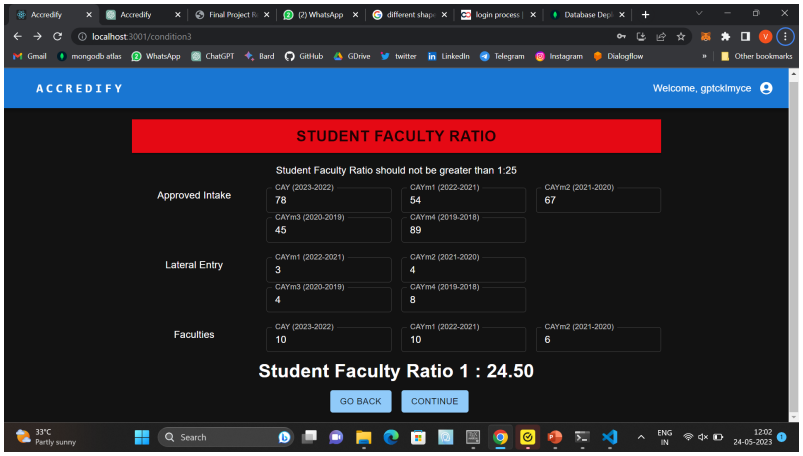
7.3 Sign Up

If the department doesn't have an account, they need to register. Departments can register using college name, department name, username, email and password. The already registered departments, username and email cannot be duplicated. After signup you will be direced to the pre qualification checking page.



7.4 Pre Qualifiers

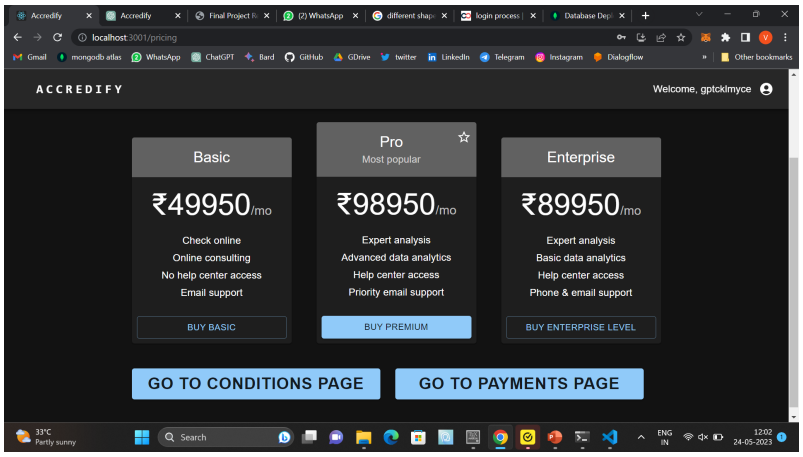
There are certain conditions for the qualification of a department such as student faculty ratio, faculty PhD ratio, student graduation rate etc. You can check it here.



7.5 Premium

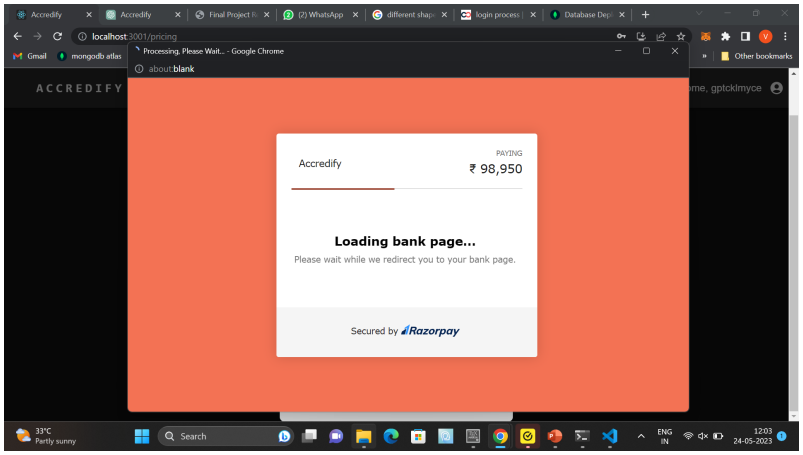
Our website also provides premium services. Advanced consultation services will be provided by our company if the department is a premium member. Razorpay is

used for the payment i



7.6 Razorpay

Razorpay is used to implement payment.

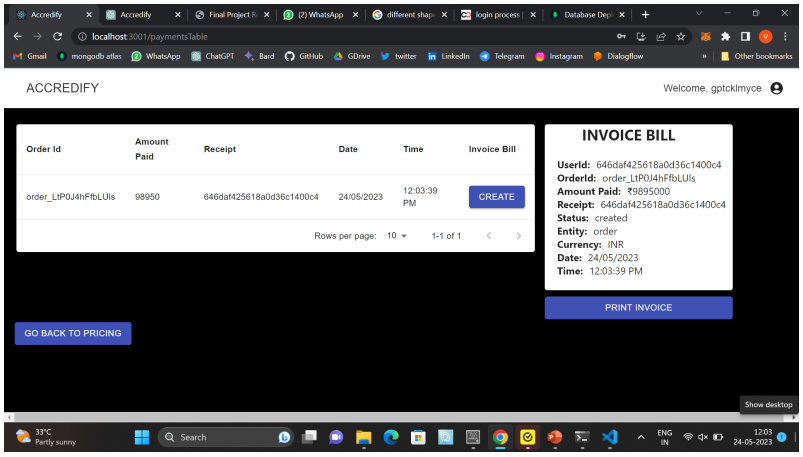


7.7 Invoice Bill

Departments can print the transaction bill after each payments.

7.8 Admin Dashboard

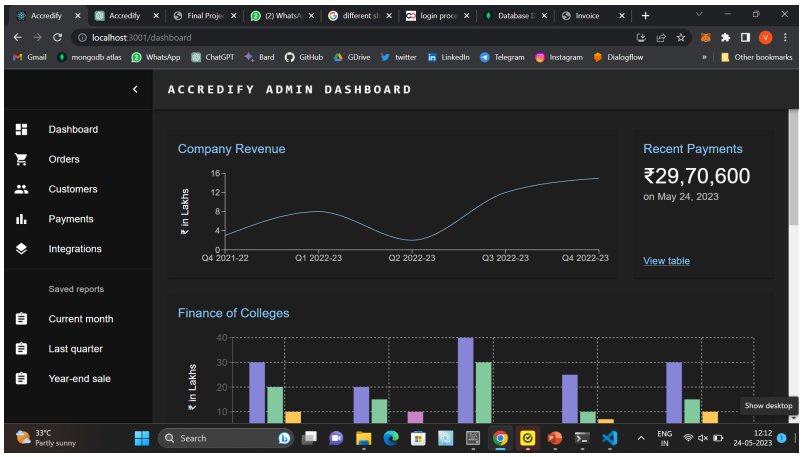
This is exclusive for admins. Admins can visualize the data of the departments. Pie charts, bar charts and graph charts are available. The revenue of the company, details of the departments registered is visible in the admin dashboard.

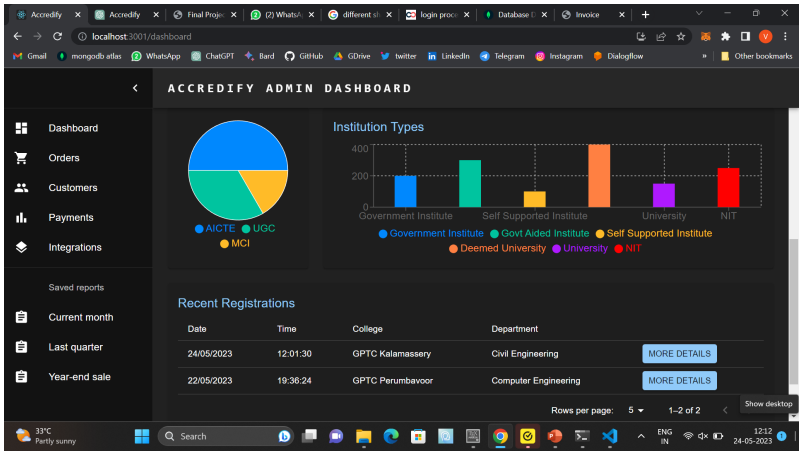


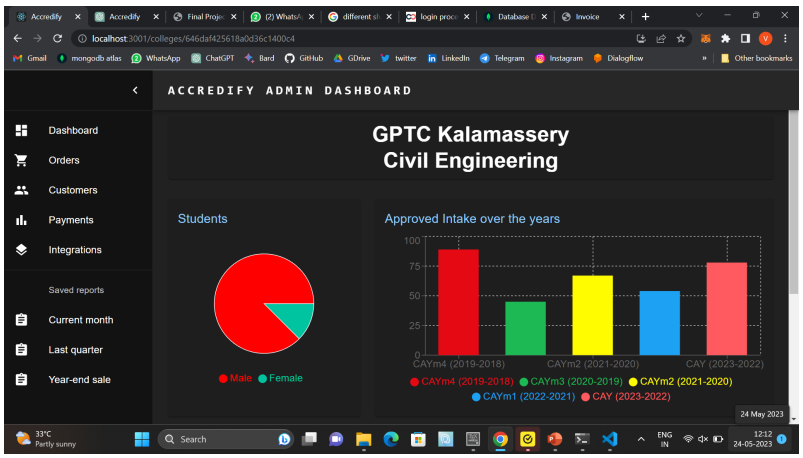
5/24/23, 12:03 PM

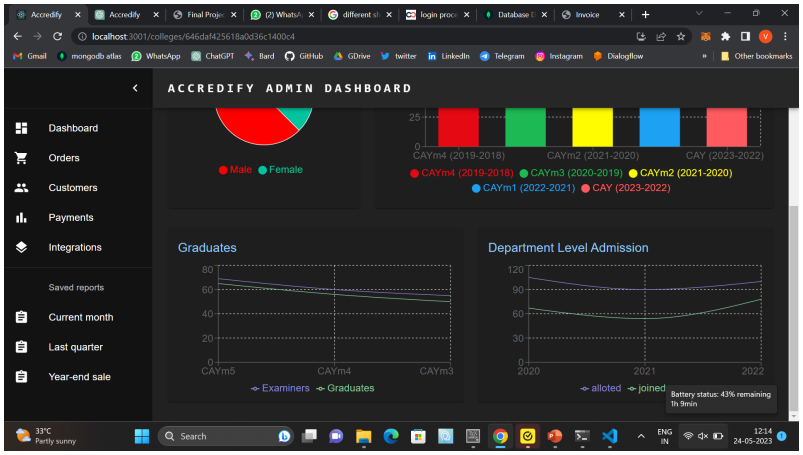
INVOICE BILL

UserId: 646daf425618a0d36c1400c4
OrderId: order_LtP0J4hFfbLUls
Amount Paid: ₹9895000
Receipt: 646daf425618a0d36c1400c4
Status: created
Entity: order
Currency: INR
Date: 24/05/2023
Time: 12:03:39 PM




ACCREDIFY ADMIN DASHBOARD
Dashboard
Orders
Customers
Payments
Integrations
Saved reports
Current month
Last quarter
Year-end sale
AICTE
UGC
MCI
Institution Types
Government Institute
Self Supported Institute
University
NIT
Government Institute
Govt Aided Institute
Self Supported Institute
Deemed University
University
NIT
Recent Registrations
Date
Time
College
Department
24/05/2023
12:01:30
GPTC Kalamassery
Civil Engineering
MORE DETAILS
22/05/2023
19:36:24
GPTC Perumbavoor
Computer Engineering
MORE DETAILS
Rows per page:
1–2 of 2
Show desktop


ACCREDIFY ADMIN DASHBOARD
GPTC Kalamassery
Civil Engineering
Students
Male
Female
Approved Intake over the years
CAYm4 (2019-2018)
CAYm2 (2021-2020)
CAY (2023-2022)
CAYm4 (2019-2018)
CAYm3 (2020-2019)
CAYm2 (2021-2020)
CAYm1 (2022-2021)
CAY (2023-2022)
24 May 2023


ACCREDIFY ADMIN DASHBOARD
Male
Female
CAYm4 (2019-2018)
CAYm2 (2021-2020)
CAY (2023-2022)
CAYm4 (2019-2018)
CAYm3 (2020-2019)
CAYm2 (2021-2020)
CAYm1 (2022-2021)
CAY (2023-2022)
Graduates
CAYm5
CAYm4
CAYm3
Examiners
Graduates
Department Level Admission
2020
2021
2022
alloted
joined
Battery status: 43% remaining
1h 9min

Chapter 8

Conclusion

In conclusion, the development of this application has been driven by a set of clear objectives that have guided the entire process. By addressing a specific problem or need, the application aims to provide a valuable solution that meets the requirements and preferences of its users. The focus on enhancing user experience has resulted in intuitive interfaces, smooth navigation, and visually appealing elements that create a seamless and enjoyable interaction for users.

Reliability and performance have been prioritized, ensuring that the application operates efficiently and minimizes downtime or errors. The implementation of robust security measures underscores the commitment to protecting user data and maintaining their privacy. Scalability and maintainability have also been considered, allowing the application to adapt and grow alongside changing user demands and technological advancements.

By aligning with business goals, the application contributes to the organization's success, whether through revenue generation, customer expansion, or operational improvements. Through these objectives, the development of the application has been driven by a clear vision and purpose, ensuring that it delivers value to its users and remains relevant in the long run.

Overall, the successful achievement of these objectives demonstrates the dedication and commitment of the development team, stakeholders, and all involved in the creation of the application. With the objectives as the guiding force, the application is poised to make a positive impact, provide value, and meet the needs of its users, contributing to their success and satisfaction.

Chapter 9

References

- Pro MERN Stack
  https://link.springer.com/book/10.1007/978-1-4842-4391-6
- MERN Stack Web Development
  https://www.annalsofrscb.ro/index.php/journal/article/view/7719
- Accreditation system for technical education programmes in India: A critical review
  https://www.tandfonline.com/doi/abs/10.1080/03043790903497294
- Seaborn: Statistical data visualization
  https://joss.theoj.org/papers/10.21105/joss.03021.pdf
- Effectiveness of NBA accreditation processes
  https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e7223fd9946ca8a76bc188f84fa71753f7836b57#page=100