MINI PROJECT REPORT ON

STUDENT REGISTRATION SYSTEM FOR MCA DEPT

Submitted in partial fulfilment of the requirement for the award of degree in

MASTER OF COMPUTER APPLICATIONS

of the

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted by

ANOOP B K - NCE21MCA2009

Semester 3 MCA (2021-23)

Under the guidance of

Mr.Pramod K

ASSOCIATE PROFESSOR

DEPARTMENT OF MCA

NEHRU COLLEGE OF ENGINEERNG AND RESEARCH CENTRE,

NOV 2022

(NAAC Re-Accredited with "A" grade) PAMPADY,

THIRUVILWAMALA, THRISSUR – 680567 NOVEMBER 2022

MINI PROJECT REPORT ON

STUDENT REGISTRATION SYSTEM FOR MCA DEPT

Submitted in partial fulfilment of the requirement for the award of degree in

MASTER OF COMPUTER APPLICATIONS

of the

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted by

ANOOP BK - NCE21MCA2009

Under the guidance of

Mr.Pramod K

ASSOCIATE PROFESSOR

DEPARTMENT OF MCA

NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE

DEPARTMENT OF MCA

COLLEGE VISION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

COLLEGE MISSION

NCERC is committed to transform itself into a centre of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values. We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

DEPARTMENT VISION

To create a school of distinction for the PG students, prepare them to be industry- ready, and achieve Academic excellence by continuous endorsement of the Faculty team in terms of Academics, Applications & Research.

DEPARTMENT MISSION

The Department of Computer Applications strives to provide quality and competency-based education and fine-tune the younger generation through Curricular, Co-Curricular and Extracurricular activities so as to encounter the Professional and Personnel challenges ahead with Pragmatics skills &courage , thereby ‘Creating the True Citizens’.

DECLARATION

I hereby declare that the project entitled **"STUDENT REGISTRATION SYSTEM FOR DEPT WEBSITE"** submitted to the **Department of MCA at Nehru College of Engineering and Research Centre** in partial fulfillment of the requirement for the award of degree in **MASTER OF COMPUTER APPLICATION** from **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**, is a record of original work done by me under the guidance of **MR.PRAMOD K**, Associate Professor of the Department of MCA, during my Third Semester MCA course period 2021-2023.

ANOOP B K

NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE, PAMPADY

CERTIFICATE

This is to certify that, the project work entitled **"STUDENT REGISTRATION SYSTEM FOR MCA DEPT WEBSITE"** has been presented by **ANOOP B K, NCE21MCA2009** of Third Semester MCA in Partial Fulfilment of the requirement for the award degree **MASTER OF COMPUTER APPLICATIONS, APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY.**

We also certify that the work done is original.

Project Guide **Head of the Department**

Principal **External Examiner**

ACKNOWLEDGEMENT

First and foremost, I thank the God Almighty for showing me the path to the completion of project work. I thank Prof.Dr. Ambikadevi Amma T, the principal of NCERC, for providing good atmosphere for project completion and presentation. I thank Prof.Dr.Sudheer S Marar, Head of MCA Department, NCERC for his valuable suggestions and support throughout the project work. I wish to acknowledge my deep sense of gratitude and heartfelt thanks to My Project Guide Mr.Pramod K, Associate professor, MCA Department for his valuable suggestions, precious time, kind hearted motivations throughout the project work period. I express my immense gratitude to all my friends, without whom I would have never been able to do my project well.

ABSTRACT

Nehru College of Engineering and Research Centre is a huge campus with lot of courses and corresponding departments. The Campus is a place where many activities takes place in a day. So Department of MCA could make an official website to showcase the skills and achievements of the department Master of Computer Application. This Website will have a potential to interact directly with the department and become a part of events occur in the MCA Department. The Department website could help to get admission enquires as well as students who are interested to get admission will be guided by this website. Department Announcements and updates will be published on the website for students and staffs only or for the public. Tech events, Conferences and other events can be registered through the MCA Department website. The registered candidates will be updated occasionally through email and SMS updates.

A Hall of Fame section will be provided to showcase the achievements made by the staffs and students on and off the campus. Ongoing Students and Staff could create account for getting updates on events happening inside the campus. Students and Staffs will have a space for showcasing their artistic and writing skills. Head of the Department will be the charge of all activities happening in the website. An Enquiry and Feedback section will be provided for the users of the websites to directly contact the department. The Queries will be directed towards the corresponding staff. The links for the Social Medias will be provided to have further look into the department.

The Main attraction of this site it will store achievements and will be remembered for ever even students have graduated and left the campus. Students get a public attention for their personal achievements.

CONTENTS

Declaration I

Certificate II

Acknowledgments III

Abstract IV

Contents V

List of Figures VI

List of Tables VII

1. Introduction 1

1.1 Background 1

1.2 Motivation 2

1.3 Objective 2

1.4 Contribution 3

1.5 Report organization 3

2. Literature Survey 4

3. Methodology 4

3.1 Introduction 5

3.2 Workflow 6

3.3 Modules and Descriptions 6

4. Agile Methodology 10

4.1 Introduction 10

4.2 User Story 12

4.3 Product Backlog 13

4.4 Project Plan 14

4.5 Sprint Backlog (Plan) 16

4.6 Sprint Backlog (Actual) 18

4.7 Sprint Review 22

4.8 Testing and Validation 23

4.9 Git 26

5. System Design and Implementation 27

5.1 Data Model Diagram 27

5.2 Implementation 27

6. Conclusion 37

6.1 Summary 37

6.2 Limitations 37

6.3 Future Scope 38

7. Bibliography 39

7.1 **Annexure** 40

7.2 Source Code 40

7.3 Output 57

7.4 Git History 61

LIST OF FIGURES

3.1	Workflow diagram	6
4.1	Github	26
5.1	Data Model Diagram	27
5.2	Result Analysis	33
7.1	Homepage	57
7.2	Login	57
7.3	Signup	58
7.4	Announcements	58
7.5	Posts	59
7.6	Feedback	59
7.7	Prayaana	60
7.8	Pizada	60
7.9	Git History	61

LIST OF TABLES

4.1	User Story	12
4.2	Product Backlog	13
4.3	Project Plan	14
4.4	Sprint Backlog (Plan) – Sprint 1	16
4.5	Sprint Backlog (Plan) – Sprint 2	17
4.6	Sprint Backlog (Plan) – Sprint 3	18
4.7	Sprint Backlog (Plan) – Sprint 4	19
4.8	Sprint Backlog (Actual) – Sprint 1	19
4.9	Sprint Backlog (Actual) – Sprint 2	20
4.10	Sprint Backlog (Actual) – Sprint 3	21
4.11	Sprint Backlog (Actual) – Sprint 4	22
4.12	Testing and Validation(Sprint 1)	24
4.13	Testing and Validation(Sprint 2)	25
4.14	Testing and Validation(Sprint 3)	25
4.15	Testing and Validation(Sprint 4)	25
5.1	user	34
5.2	announcement	34
5.3	notification	34
5.4	pizada	35
5.5	post	35
5.6	prayaana	35
5.7	feedback	36
5.8	toppers	36
5.9	placements	36

Chapter 1

INTRODUCTION

Student Registration is a role-based registration for students who are enrolled on current academic year. Registration is mandatory to access exclusive contents available on the web application.

1.1 Background

Basically a website for a college/department are limited to the officials and staffs. So developed and build a web application where students and staffs can equally participate to create and release contents to public. Their works will be kept forever even though they have graduated from the campus. It gives a special space for the students those who have unique interests.

Nehru College of Engineering and Research Centre has around 20+ departments so it difficult to manage a single official website for the college. So building a site especially for the MCA Department helps the department to explore various aspects of creating and publishing their works. It will be less hectic to publish contents instead of relying on the college official website. Another feature is that it is difficult to contact details of a particular department using a college website so official department website will helps others to interact directly with the department of MCA.

- There was no space to create and publish contents related to MCA Department
- Contacting directly with the department is hard
- No space for showcase department activities occurring inside and outside the department

1.2 Motivation

Students having extraordinary skills does not get an official platform to publish and advertise themselves. Not everyone is good at studies, each individuals have their own unique set of skills which make them unique. Those skills can be of any type like writing, photography, playing games etc. So providing a space for them to prove themselves to the world. Those published contents are worth for a resume to express each individuals to the employer. Making good moments and keeping memories as well as after moving out keeping updates about the department helps to have a strong emotional connection with the department forever.

A Student who tends to get admission to MCA at NCERC can look over the activities and decide whether he/she should join the campus. Direct contact to the HOD and other staffs will provide a better and efficient way of taking admissions easily and know about all the aspects of the department.

- Express each students with their own unique talents
- Leave a footprint behind even when the students are graduated from the campus
- Easy access to the staffs
- Enquires will be easy especially related to the MCA Department

1.3 Objective

The idea of this project is to build an official web application especially for the department of MCA.

The Objectives of the project are:

- Official website to showcase the skills and achievements of the Department of MCA
- Potential to interact directly with the department

- Helps to get admission enquiries
- Department Announcements and updates will be published on the website
- Students and Staffs will have a space for showcasing their artistic skills

1.4 Contribution

The major contributions of the project is:

- Designed and developed a web application entirely dedicated to the MCA Department of NCERC

1.5 Report Organization

The project report is divided into six sections. Section 2 describes literature survey. Section 3 describes the methodology and section 4 describes agile methodology used for implementing the project. Section 5 gives the results and discussions. Finally, Section 6 gives the conclusion

Considering a web application like this project gives a goodwill about the certain department in front of the public. It will boosts the admissions for the department. Direct contact to the department will helps to gives a less hassle for taking admissions and other enquires. Students who have passed out will can revisit and keep updates about the department. Which will helps to have a good alumni connection with the department.

CHAPTER 2

LITERATURE SURVEY

We have a range of apps for chatting and video calling with friends, as well as platforms like Instagram, Facebook, and LinkedIn where we can obtain updates on a variety of topics like entertainment, news, sports, and more. Some apps are specifically for sharing entertainment related activities or activities etc. like Instagram, Facebook, LinkedIn to know updates about many things including entertainment, news, sports. The social media platforms now available are: Face book : Facebook become first released in 2004 as a Harvard social networking website online, accelerated to different universities and in the end reached out to all of us. It became the biggest social media website in 2009.It remains a great photo-sharing site. Marketing strategies have found that Facebook is useful because it encompasses a wide range of personal and organizational interests. Twitter: Twitter became founded in2006 by using Odeon, Inc and changed into in the beginning simplest for Odeo Inc personnel and family individuals. It has become a public network in 2006.Twitter affords an actual-time, web based service that permits users to post brief messages for different customers and to comment on other person posts. A tweet is a small message of no extra than a hundred and forty characters that users create to talk mind.

Chapter 3

METHODOLOGY

3.1 Introduction

MERN is MongoDB, Express.js, React.js, and Node.js. Express.js is a web server framework (together with Node.js); React.js is a web purchaser library, and Node.js as server-side platform to MongoDB as a database. MERN combines four cutting-edge technologies in today's development, from front-end to back-end. It saves effort and time for builders to grasp new technology for utility improvement. The stack is supported by a large number of open-source packages and a committed community enabling programmers to boost scalability and maintain software products, thanks to the same JavaScript platform.

The basis of the MERN stack is Node.js, a server-side technology that provides very excessive overall performance and rapid reaction to all duties, such as huge and complicated data. MERN doesn't need Typescript; all it needs is the flexible basis of React, today's most popular and important front-end technology

Redux is a predictable state container for JavaScript applications. Redux derives its ideas form the Flux architecture. It is basically a flux-like approach to React applications. Redux doesn’t necessarily have to be used with React only, It can use it for AngularJS or JS too. But Redux works really well with React. Redux basically provides a way for managing the store, unidirectional flow of data and much more.

3.2 Workflow

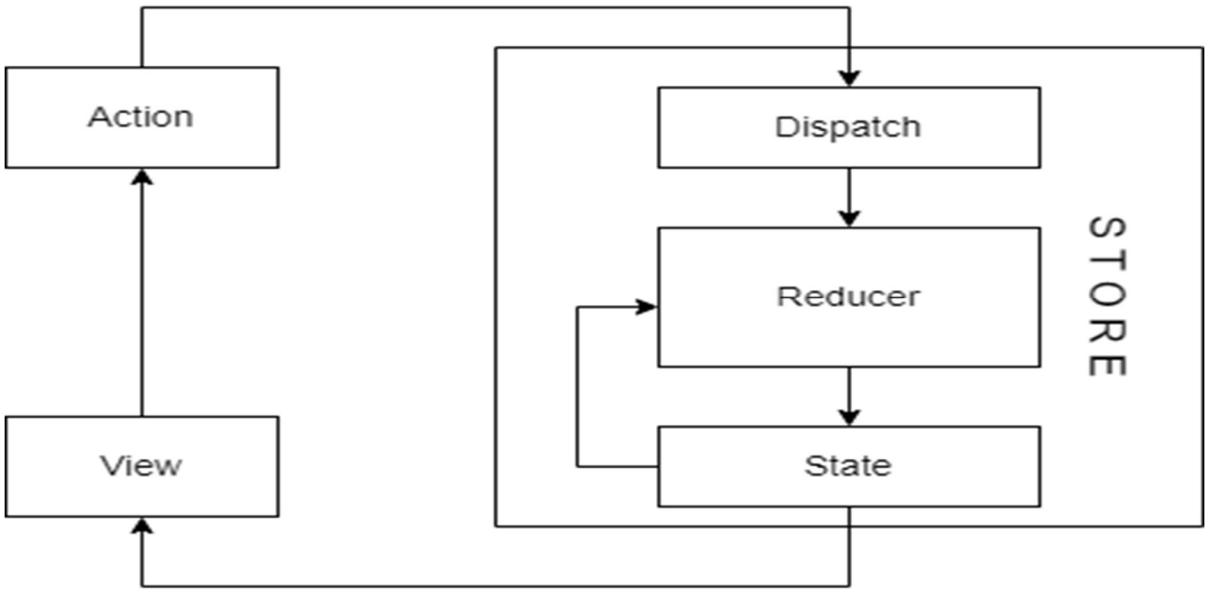

Figure 3.1: Workflow

3.3 Modules and Descriptions

Admin

Admin has the overall control over application. Admin can manage user accounts. Admin can remove, disable and modify the user accounts. Create new announcements, Add various events such as data about events, placements, toppers etc

- Admin have the higher level privileges to control the site
- Managing Posts, Queries, Registrations can be controlled by admin
- Maintains integrity of the application
- Admin Dashboard contains Manage Accounts, Feedback, Prayaana Registration, Pizada file uploads, Post announcements etc

Student

Student can login and post contents to the post section. Gets custom notifications from the staffs. Manage their profile. Manage their posts etc.

- Student need to create a login id and a profile to access the site
- Students can create post and manages those posts.

Staff

Staff has similar to the student account. They can post and manage their posts. Staff can send notifications to a specific batch. Manage their profiles etc

- Staff needs to create a login id and a profile to access the site.
- Staffs can create posts and manages those posts.
- Staffs can interact with groups of students easily using this site.

Prayaana

Prayaana is a tech fest conducted by department of MCA of NCERC. Participants can register for this event through the form provided in the site.

- Prayaana is the annual cultural event hosted by the department of MCA.
- The sites allows students to register for various events which they wish to participate.
- Posts related to Prayaana can also be posted in this site.

Pizada

Pizada is the department magazine of mca. Students and Staffs can submit their works to the magazine. A special tab is provided int website to submit the articles. Only registered students/staff can access this tab

- Pizada is the department magazine which is released every year
- Articles provided by the students are staff are published in the pizada magazine
- The Pizada modules will helps to students who have created accounts in the website could upload their works to publish on the magazine
- Uploaded files are maintained by the Admin
- Easy to collect and use the articles
- Files that can be uploaded should be either Text, Pdf, Docs, Jpg, Jpeg, Png

Announcements

Announcements are posted by admin. Important notifications can be found in this tab. Admin publishes and manages these announcements in the announcements tab

- Make official public and department announcement, Manage those announcements
- Currently Admin can only provide the announcement

Posts

Post is where the feature to post works of students and staff. Public can view these posts but only the authenticated users can post or manage their posts. Students/Staff can showcase their skills.

- Posts is used to publish the work and achievements made by the students and staff
- Only authorized students and staff can publish posts
- A Post contains Topic, Text/Image
- Users can remove and publish post whenever they want
- Admin can remove posts without a notice

Login

Login is section were authenticated users enter their credentials and login to the site. The registered users credentials are verified at the server side and gave access to the site

- Registered members can use their provided username & password for login

Registrations

Registration or signup form where a student/staff setup their account by entering their credentials. Only after registering the becomes the authenticated user

- It is a multiple login inter face.
- User can login as Student or Staff
- Students have to enter their University Registration number and other details to register in the MCA Department Site.

- Staffs have to enter their ID Number and other details to register in the MCA Department Site.

Notifications

Notifications are specially provided for the authenticated student accounts were the staff post custom notification dedicated for a class which is identified by the batch

- Send notification staff to student
- Notification is forwarded based on batch
- Students receive these notification under their profile

Feedback

Feedback is a form where public can query their thoughts or send feedback to the department

- Feedback and queries from the public, students and staffs are accepted by the form provided under the contact us
- Those feedbacks are forwarded to the admin and admin replies or take necessary actions regarding the feedbacks and queries

Chapter 4

AGILE METHODOLOGY

4.1 Introduction

Agile is an iterative approach to project management and software development that helps teams to deliver value to their customers faster and with fewer headaches. Instead of betting everything on "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to changes quickly. Teams choose agile so they can respond to changes in the marketplace or feedback from customers quickly without derailing a year's worth of plans. "Just enough" planning and shipping in small, frequent increments lets team to gather feedback on each change and integrate it into future plans at minimal cost. An Agile team unites under a shared vision, then brings it to life the way they know is best. Each team sets their own standards for quality, usability, and completeness. Their "definition of done" then informs how fast they'll churn the work out Although it can be scary at first, company leaders find that when they put their trust in an agile team, that team feels a greater sense of ownership and rises to meet (or exceed) management's expectations. Agile is one of the most widely used and recognized software development frameworks. The methodology those experts agreed open was described as ‘lightweight’ and fast. Agile is also about being the adaptive and continuous improvement, as much as it is about constant feedback and speed of delivery

Agile is a software development approach where a self-sufficient and cross-functional team works on making continuous deliveries through iterations and evolves throughout the process by gathering feedback from the end users. The major rules in scrum methodology are:

1. The product owner (PO): Who represents the stakeholder and the business.

2. The scrum master Ensures the process followed, removes obstructions, and protects the development system

3. Development team: Cross functional, self-organizing team who actually do the actual analysis, design implementation and testing process

They work together in iterative time boxed durations called sprints. The first step is the creation of the product backlog by the PO. It's a to-do list of stuff to be done by the scrum team. Then the scrum team selects the top priority items and tries to finish them within the timebox called a cat. An easier way to remember all of this is to memorize the 3-3-5 frame-work Is that a scrum project has 3 roles, 3 artifacts, and 5 events These are:-

1. Roles Product Owner, Scrum Master, and development team

2. Artifacts Product Backlog. Sprint Backlog and Product Increment

3. Events Sprint, Sprint planning, Daily Scrum, Sprint review and Sprint retrospective

The framework begins with a simple premise start with what can be seen or known. After that the progress is tracked and tweak as necessary. The three pillars of serum are transparency, inspection and adaptation. In scrum everyone has a role

The Git is used as the version control system for this project Version control is a system that records changes to a file or set of files over time so that a specific version can be recalled liter Version control systems are a category of software tools that help a software team for managing changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing

disruption to all team members A user story is a tool used in agile software development to capture a description of software feature from an end-user perspective The of story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement

4.2 User Story

A user story is a tool used in agile software development to capture a description of software feature from an end-user perspective. The user story describes the type of user, what they want and why A user story helps to create a simplified description of a requirement

User Story ID	As a <Type of user>	I want to perform <some tasks>	So that I can <Achieve some goal>
1	Student and Staff	Register to the system	Access the system
2	Student and Staff	Login	Access the account
3	Student and Staff	Post	Create a post
4	Staff	Notification	Sends notification to a batch of students
5	Student	Submit	Submit articles for magazine
6	User	Feedback	Send feedback to
7	User	Register prayaana	Register for the event prayaana

Table 4.1 User Story

4.3 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome. The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.

It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric. e.g., in the form of user stories. The product backlog of the system is given in the table 4.2

PRODUCT BACKLOG			
ID	Name	Priority	Estimate[Hrs]
1	Registration	1	9
2	Login	2	8
3	Add Post	3	7
4	Notification	4	6
5	Submit	5	6
6	Register Prayaana	6	10
7	Feedback	7	12

Table 4.2 Product Backlog

4.4 Project Plan

A Project plan has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project plan is given in Table 4.3

User story ID	Task name	Start Date	End Date	Days	Status
Sprint 1				**6**	**Completed**
1	Admin section design	12/08/2022	13/08/2022	2	Completed
2	Student section design	14/08/2022	14/08/2022	1	Completed
3	Staff section design	15/08/2022	15/08/2022	1	Completed
4	Page connection (links)	16/08/2022	16/08/2022	1	Completed
5	Testing	17/08/2022	17/08/2022	1	Completed
Sprint 2				**7**	**Completed**
6	Database design & connectivity	24/08/2022	25/08/2022	2	Completed
7	Registration & Testing	26/08/2022	28/08/2022	3	Completed
8	Login & Testing	29/09/2022	30/09/2022	2	Completed
Sprint 3				**5**	**Completed**
9	Admin section announcements	10/09/2022	11/09/2022	2	Completed
10	Posts	12/09/2022	13/09/2022	2	Completed
11	Testing	14/09/2022	14/09/2022	1	Completed
Sprint 4				**3**	**Completed**
12	Reports	19/09/2022	20/09/2022	2	Completed
13	Testing	21/09/2022	21/09/2022	1	Completed

Table 4.3 Project Plan

This project has four sprints:

1. Sprint 1

   Five tasks are planned at this stage. Complete design for this project including admin section, Student section and Staff section. Links between these pages ad overall testing

2. Sprint 2

   Three tasks are planned at this stage. Database Design & Manage Database Connectivity, Registration module, Login Module & overall testing.

3. Sprint 3

   Three tasks are planned at this stage. Admin section announcements, Creating posts and testing

4. Sprint 4

   Only two tasks are planned at this stage. Report generation and full testing

4.5 Sprint Backlog (Plan)

The Sprint backlog is a lists of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete.

1. Sprint 1

   Five tasks are planned at this stage. Complete design for this project including admin section, Student section and Staff section. Links between these pages ad overall testing
2. Sprint 2

   Three tasks are planned at this stage. Database Design & Manage Database Connectivity, Registration module, Login Module & overall testing.
3. Sprint 3

   Three tasks are planned at this stage. Admin section announcements, creating posts and testing
4. Sprint 4

   Only two tasks are planned at this stage. Report generation and full testing

Backlog items	User story #1 Hours	Admin section design	Student section design	Staff section design	Page connection (links)	Testing
Completion date		Hours	Hours	Hours	Hours	Hours
Original estimate in hrs	Hours	7	2	2	1	1

Day 1 12/08/2022	Hours	4	0	0	0	0
Day 2 13/08/2022	Hours	3	0	0	0	0
Day 3 14/08/2022	Hours	0	2	0	0	0
Day 4 15/08/2022	Hours	0	0	2	0	0
Day 5 16/08/2022	Hours	0	0	0	1	0
Day 6 17/08/2022	Hours	0	0	0	0	1

Table 4.4 Sprint Backlog (Plan) – Sprint 1

Backlog items	User story #1 Hours	Database design and connectivity	Registration and testing	Login and testing
Completion date		Hours	Hours	Hours
Original estimate in hrs	Hours	7	6	2
Day 1 24/08/2022	Hours	4	0	0
Day 2 25/08/2022	Hours	3	0	0
Day 3 26/08/2022	Hours	0	2	0

Day 4 27/08/2022	Hours	0	2	0
Day 5 28/08/2022	Hours	0	2	0
Day 6 29/08/2022	Hours	0	0	1
Day 7 30/08/2022	Hours	0	0	1

Table 4.5: Sprint Backlog (Plan) - Sprint 2

Backlog items	User story #1 Hours	Admin section announcements	Posts	Testing
Completion date		Hours	Hours	Hours
Original estimate in hrs	Hours	7	3	1
Day 1 10/09/2022	Hours	4	0	0
Day 2 11/09/2022	Hours	3	0	0
Day 3 12/09/2022	Hours	0	2	0
Day 4 13/09/2022	Hours	0	1	0
Day 5 14/09/2022	Hours	0	0	1

Table 4.6: Sprint Backlog (Plan) – Sprint 3

Backlog items	User story #1 Hours	Reports	Testing
Completion date		Hours	Hours
Original estimate in hrs	Hours	5	1
Day 1 19/09/2022	Hours	3	0
Day 2 20/09/2022	Hours	2	0
Day 3 21/09/2022	Hours	0	1

Table 4.7: Sprint Backlog (Plan) – Sprint 4

4.6 Sprint Backlog (Actual)

Actual Sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below

Backlog items	User story #1 Hours	Admin section design	Student section design	Staff section design	Page connection (links)	Testing
Completion date		Hours	Hours	Hours	Hours	Hours
Original estimate in hrs	Hours	7	2	2	1	1
Day 1 12/08/2022	Hours	4	0	0	0	0

Day 2 13/08/2022	Hours	3	0	0	0	0
Day 3 14/08/2022	Hours	0	2	0	0	0
Day 4 15/08/2022	Hours	0	0	2	0	0
Day 5 16/08/2022	Hours	0	0	0	1	0
Day 6 17/08/2022	Hours	0	0	0	0	1

Table 4.8: Sprint Backlog (Actual) – Sprint 1

Backlog items	User story #1 Hours	Database design and connectivity	Registration and testing	Login and testing
Completion date		Hours	Hours	Hours
Original estimate in hrs	Hours	7	6	2
Day 1 24/08/2022	Hours	4	0	0
Day 2 25/08/2022	Hours	3	0	0
Day 3 26/08/2022	Hours	0	2	0
Day 4 27/08/2022	Hours	0	2	0

Day 5 28/08/2022	Hours	0	2	0
Day 6 29/08/2022	Hours	0	0	1
Day 7 30/08/2022	Hours	0	0	1

Table 4.9: Sprint Backlog (Actual) – Sprint 2

Backlog items	User story #1 Hours	Admin section announcements	Posts	Testing
Completion date		Hours	Hours	Hours
Original estimate in hrs	Hours	7	3	1
Day 1 10/09/2022	Hours	4	0	0
Day 2 11/09/2022	Hours	3	0	0
Day 3 12/09/2022	Hours	0	2	0
Day 4 13/09/2022	Hours	0	1	0
Day 5 14/09/2022	Hours	0	0	1

Table 4.10: Sprint Backlog (Actual) – Sprint 3

Backlog items	User story #1 Hours	Reports	Testing
Completion date		Hours	Hours
Original estimate in hrs	Hours	5	1
Day 1 19/09/2022	Hours	3	0
Day 2 20/09/2022	Hours	2	0
Day 3 21/09/2022	Hours	0	1

Table 4.11: Sprint Backlog (Actual) – Sprint 4

4.7 Sprint Review

At the end of each sprint a Sprint Review meeting is held. During this meeting the Scrim Team shows which Scrum Product Backlog items they completed (according to the Definition of Done) during the sprint. This might take place in the form of a demo of the new features Backlog items that are not completed shall not be demonstrated. Otherwise, this might suggest that these items are fished as well. Instead, incomplete items remaining activities shall be taken back into the Scrum Product Backlog, re-estimated and completed in one of the following sprints The Sprint Review meeting should be kept very informal. No PowerPoint slides should be used and time for preparation and performing the meeting should be limited. During the meeting the Scrum Product Owner inspects the implemented backlog entries and accepts the solution or adds new stories to the Scrum Product Backlog to adapt the functionality Participants in the sprint review typically include the Scrum Product Owner, the Scrum Team and the Scrum Master Additionally, management, customers, and developers from other projects might participate as well

4.8 Testing and Validation

Testing is the process by which a developer will generate a test data, which gives probability of finding all types of errors that can occur in the Software Testing is vital to the access of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of testes online response, volume, stress, recovery & security and usability tests A series of testing are performed for the proposed system before the system is ready for user acceptance testing First each program is tested, next a series of modules is tested, then each individual program with all its modules, finally the entire system consisting of a series of programs is tested. In this way, problems at the program level can be corrected before the entire system is used. In a web based application is types of testing performed are:

Unit Testing: Unit testing is a testing method that tests an individual unit of software in isolation. Unit testing for React Apps means testing an individual React Component. Unit testing is important for React Apps, as it helps in testing the individual functionality of React components. Moreover, any error in code can be identified at the beginning itself, saving time to rectify it at later stages

Snapshot Testing: Snapshot tests are a very useful whenever want to make sure UI does not change unexpectedly. A typical snapshot test case renders a UI component, takes a snapshot and then compares it to a reference snapshot file stored alongside the test. The test will fail if the two snapshots do not match: either the change is unexpected, or the reference snapshot needs to be updated to the new version of the UI component.

End-to-end testing (E2E) Testing: End-to-end tests simulate actual user actions and are designed to test how a real user would likely use the application. In React,

E2E testing helps to ensure that the code wrote is functional and app works as intended, allowing to catch bugs in code before app is live.

Preparation of test data

Software testing is a crucial element of software quality assurance and represents the ultimate review of specification, design and coding. Testing represents an interesting anomaly for the software. During earlier definition and development phases, it was attempted to build software from abstract concepts to tangible implementation. The testing responsible for ensure that the product that has built performs the way that the detailed design documentation specifies

Sprint 1

Test #	Date	Action	Expected result	Actual result	Pass? (Yes/No)
1	12/08/2022	Admin component	Complete Admin interface design	Complete Admin interface design	Yes
2	14/08/2022	Student component	Complete Student interface design	Complete Student interface design	Yes
3	15/08/2022	Staff component	Complete Staff interface design	Complete Staff interface design	Yes
4	16/08/2022	Component connection	Links between components	Links between components	Yes

Table 4.12: Testing and Validation – Sprint 1

Sprint 2

Test #	Date	Action	Expected result	Actual result	Pass? (Yes/No)
1	24/08/2022	Database design and connectivity	Design all database schema	Design all database schema	Yes
2	26/08/2022	Registration	All types of user registration	All types of user registration	Yes
3	29/08/2022	Login	All types of login	All types of user login	Yes

Table 4.13: Testing and Validation – Sprint 2

Test #	Date	Action	Expected result	Actual result	Pass? (Yes/No)
1	10/09/2022	Admin announcement	Announcement post	Announcement post	Yes
2	12/09/2022	Post	Post contents	Post contents	Yes

Table 4.14: Testing and Validation – Sprint 3

Test #	Date	Action	Expected result	Actual result	Pass? (Yes/No)
1	21/09/2022	Reports	Generate reports for admin and users	Generate reports for admin and users	Yes

Table 4.15: Testing and Validation – Sprint 4

4.9 Git

Git is free and open source software for distributed version control: tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. To show the continuous development of the project the Github commit histories are shown in Appendix from figure A.12 to A.15

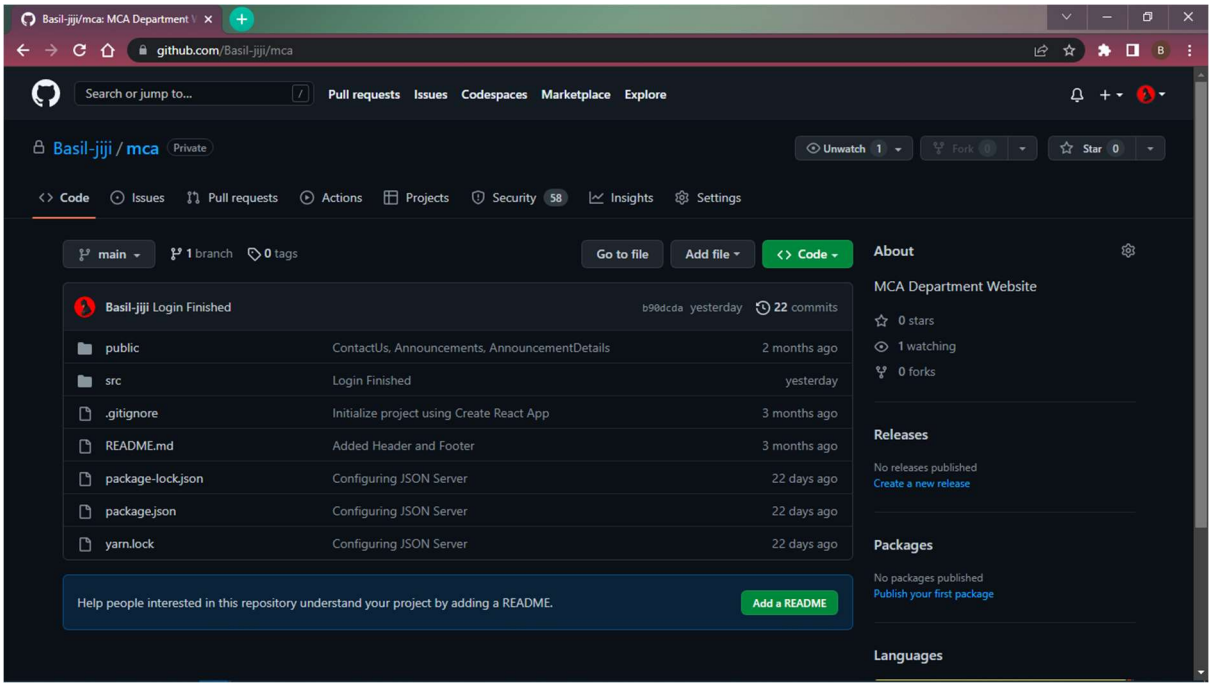

Figure 4.1: Github

Chapter 5

SYSTEM DESIGN AND IMPLEMENTATION

5.1 Data Model Diagram

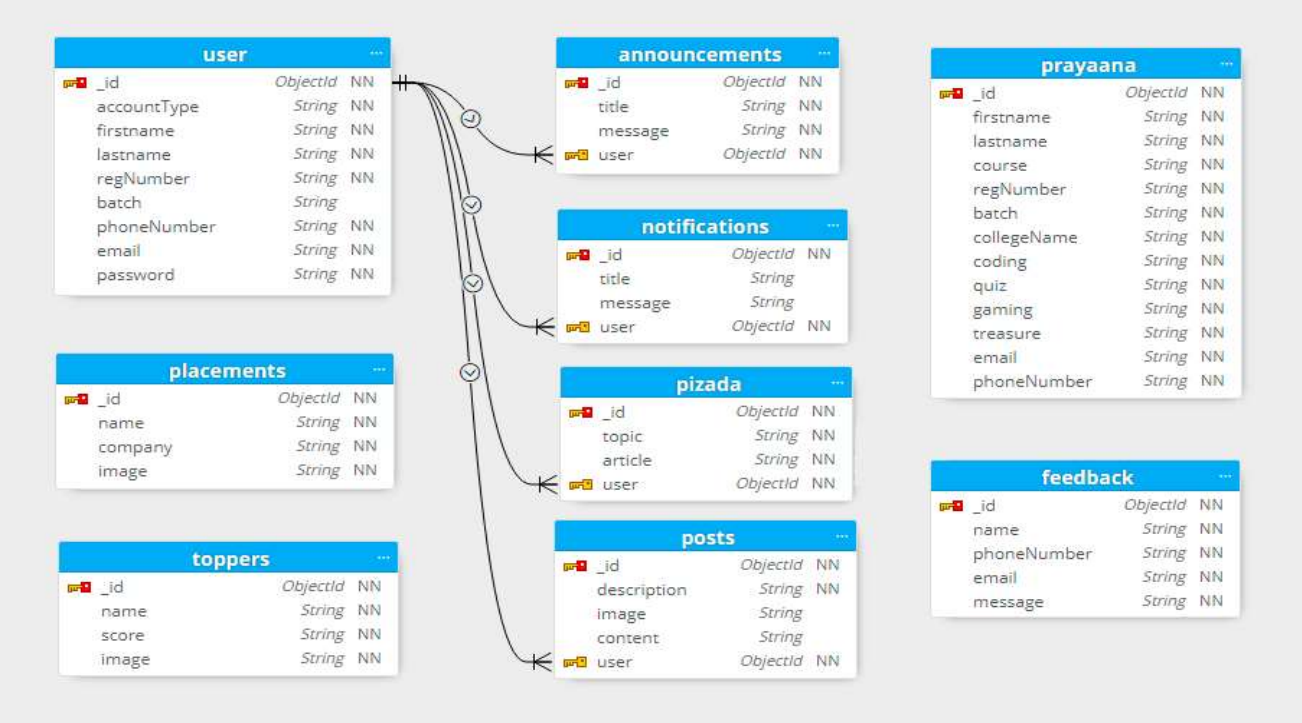

Figure 5.1 Data Model Diagram

5.2 Implementation

React

React is a JavaScript library for building user interfaces. React.js is one of the most talked about JavaScript web frameworks in years. It have a big impact on the way we build web applications. React.js is generally thought of as the view layer in an application. Some might have used library like Handlebars, or jQuery in the past. Just as jQuery manipulates UI elements, or Handlebars templates are inserted onto the page, React components change what the user sees. This is literally all there is to React. We want to render this data to the UI,

so we pass it to a React component which handles the job of getting the HTML into the page. On the surface, React appears to be another rendering technology. But it's much more than that. It can make application development incredibly simple. React doesn't have many moving parts for us to learn about and understand. The advantage to having a small API to work with is that it can spend more time familiarizing with it, experimenting with it, and so on. The opposite is true of large frameworks, where all the time is devoted to figuring out how everything works. React is divided into two major APIs. First, there's the React DOM. This is the API that's used to perform the actual rendering on a web page. Second, there's the React component API. These are the parts of the page that are actually rendered by React DOM. The syntax used by React components is called JSX (JavaScript XML). The idea is actually quite simple. A component renders content by returning some JSX. The JSX itself is usually HTML Markup, mixed with custom tags for the React components. What's absolutely ground breaking here is that we don't have to perform little micro-operations to change the content of a component. React components don't require executing steps in an imperative way to render content. This is why JSX is so central to React components. The XML-style syntax makes it easy to describe what the UI should look like. That is, what are the HTML elements that this component is going to render? This is called declarative programming, and is very well-suited for UI development.

- Easy creation of dynamic applications
- Improved performance
- Reusable components
- Unidirectional data flow
- Small learning curve
- It can be used for the development of both web and mobile apps
- Dedicated tools for easy debugging

Advantages of React

- **Selective Rendering with Virtual DOM:** Manipulating virtual DOM is quicker than updating real DOM. It can be compared to editing blueprints for a room versus moving actual materials physically.
- **Code Reusability**: React can keep the state out of the DOM. React.js utilizes components for development for the majority of its development process which are reusable.
- **Benefits of React SSR Rendering support:** React supports SSR Rendering, i.e., server-side rendering. Through SSR, Single Page Applications are rendered on the server-side in place of the client's browser.
- **Useful React Developer Toolset:** React comes with features and tools required for development, designing, testing, debugging, and other uses for developers to fully utilize the platform for app development.
- **Architecture Free Approach:** Only need to learn about the component's life cycle, props, and states. The React gives complete freedom to set up the architecture as per their preferences and requirements.

Disadvantages of React

- **Limitations of React being a 'library':** It raises the concern of overall React web app size with different libraries that bring their packages. Moreover, it exposes the app to security and consistency concerns as developers need to rely on the continuity of the external libraries to ensure their React app functions properly at all times.
- **Creating a Long List of Items:** App needs to render a considerable number of cards with multimedia content such as images, links, titles, and they also need to be responsive. In that case, There may notice

performance slowing down on low specification laptops and mobile devices.

- **Excessive Freedom and Verbose Code:** It also need to hand over the project from one React company to another or change the development team, it can be difficult as both the teams may have different approaches to the same problem.

NodeJS

Node.js is an open-source server environment. Node.js is cross-platform and runs on Windows, Linux, Unix, Mac OS, etc. Node.js is a back-end JavaScript runtime environment. Node.js runs on a JavaScript Engine (i.e. V8 engine) and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games). The Node.js distributed development project was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS Foundation. OpenJS Foundation is facilitated by the Linux Foundation's Collaborative Projects program. Node modules provide a way to re-use code in a Node application. In some ways, they're similar to a class in other languages, like C# or Java. In many ways, they're completely different from a class. When writing a Node application, used of modules just may not have known it. At its core, a module is a piece of

re-usable code with a defined interface. It bring a module into scope using require() like this. The Node core modules include modules like path, stream, etc. Core modules are installed when installing Node itself. If a core module can't be found, Node assumes the identifier refers to a Node package. When attempting to locate a Node package, require() looks for a node_modules directory in the current directory and checks it for the package. **npm** (originally short for **Node Package Manager**) is a package manager for the JavaScript programming language maintained by npm, Inc. npm is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website. The package manager and the registry are managed by npm, Inc.

Express

Fast, Unopinionated, Minimalist web framework for Node.js. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. With a myriad of HTTP utility methods and middleware at disposal, creating a robust API is quick and easy. Express provides a thin layer of fundamental web application features, without obscuring Node.js features. Express.js lets users build their application with their own preferences. This is especially advantageous when creating a novel personal project with no historical burden, but as the developing team grows, lack of standardization may lead to extra work for project management, and in the worst case scenario; it may lead to the inability to maintain. Express.js has a generator that generates specific folder structure. After installing express-generator npm (package manager for Node.js packages) package and creating a skeleton with generator command, an application folder with clear hierarchy will be made to aid to manage front-end static JavaScript, stylesheet

files, images, and also HTML template files. An Express application is basically Node.js with a host of middleware functions, where it take advantage of the built-in middlewares of the framework and create a REST API Server. Express has made the process natural and intuitive. Express-generator makes use of Pug (that is originally known as 'Jade') template engine by default, and other options like Mustache and EJS also work with Express flawlessly. Express does not consider database integration as a needed aspect within its package, thus it leans toward no specific database usage of any kind.

MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL) which is deemed non-free by several distributions. MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and function which is the equivalent of relational database tables. MongoDB is a database which came into light around the mid-2000s. Each database contains collections which in turn contains documents. Each document can be different with a varying number of fields. The size and content of each document can be different from each other. The document structure is more in line with how developers construct their classes and objects in their respective programming languages. Developers will often say that their classes are not rows and columns but have a clear structure with key-value pairs. The rows (or documents as called in MongoDB) doesn't need to have a schema defined beforehand. Instead, the fields can be created on the fly. The data model available within MongoDB allows

to represent hierarchical relationships, to store arrays, and other more complex structures more easily. The MongoDB environments are very scalable. Companies across the world have defined clusters with some of them running 100+ nodes with around millions of documents within the database.

Result Analysis

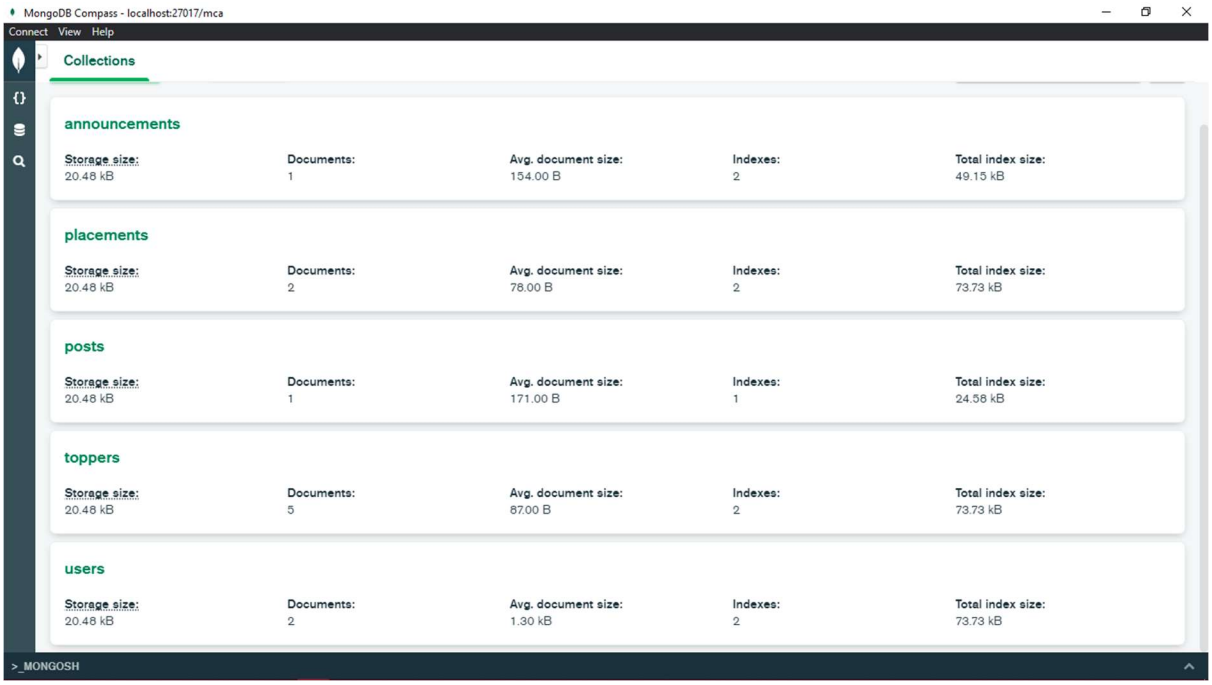

Figure 5.2: Result Analysis

Structure of Collections

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
accountType	String	T	-	-
firstname	String	T	-	-
lastname	String	T	-	-
regNumber	String	T	-	T
batch	String	T	-	-
phoneNumber	String	T	-	-
email	String	T	-	T
password	String	T	-	-

Table 5.1: user

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
title	String	T	-	T
message	String	T	-	-

Table 5.2: announcements

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
title	String	T	-	T
message	String	T	-	-

Table 5.3: notifications

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
topic	String	T	-	T
article	String	T	-	-

Table 5.4: pizada

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
description	String	T	-	T
image	String	T	-	-
content	String	T	-	-

Table 5.5: posts

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
firstname	String	T	-	-
lastname	String	T	-	-
course	String	T	-	-
regNumber	String	T	-	T
batch	String	T	-	-
collegeName	String	T	-	-
coding	Boolean	T	-	-
quiz	Boolean	T	-	-
gaming	Boolean	T	-	-
treasure	Boolean	T	-	-
email	String	T	-	T
phoneNumber	String	T	-	T

Table 5.6: prayaana

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
name	String	T	-	-
phoneNumber	String	T	-	-
email	String	T	-	-
message	String	T	-	-

Table 5.7: feedback

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
name	String	T	-	-
score	String	T	-	-
image	String	T	-	-

Table 5.8: toppers

FIELD NAME	DATATYPE	REQUIRED	DEFAULT	UNIQUE
_id	ObjectID	T	-	T
name	String	T	-	-
company	String	T	-	-
image	String	T	-	-

Table 5.9: placements

Chapter 6

CONCLUSION

6.1 Summary

The MCA Department website is designed and suited for student/staff activities also including various provisions for public to know more about the department of MCA in Nehru College of Engineering and Research Centre. It helps us to overcome to stick with only the official page of the campus is be unveiled. So department gets full authority and freedom to post and manage achievements and other activities held at MCA department. Posting various activities to the public will helps to boost the admissions and helps outsiders to know more about what is happening inside the department. This site acts as a bridge between alumini students as well. Overall the site has potentially used for showcase talents which comes over and over every year.

6.2 Limitations

1. Security Risks Posed by threat vectors

Cyber threats are undoubtedly one of the biggest limitations posed by the e-auction process. Intruder may perform various types of attacks to gain access or to deface the application.

2. Managing Passout Students

Every year around 60 students graduate from the campus so hibernating their accounts are counterproductive when compare to doing this every year

3. Limited number of roles

Student, Staff and Admin are the major roles. There are many scope for specialized users in this project. Basically this is a 1.0 version so making such changes may expand the project into a larger scale

6.3 Future Scope

A department site is not limited on what it is being build. Specialized users who conducts various programs could do a massive work force and make less hassle for the administrator. Another one is support for new initiatives on the website without adding a single line of code. Because so much initiatives occurs on the department every month. So there are many such future scopes available. The upcoming changes on what we see on other websites has always can be adopted so it future scope is a never ending process.

BIBILOGRAPHY

References

- MongoDB: The Definitive Guide
  Book by Kristina Chodorow and Michael Dirolf
- The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React. Js
  Book by Robin Wieruch
- Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node
  Book by Vasan Subramanian
- https://reactjs.org/
- https://nodejs.org/en/
- https://expressjs.com/
- https://www.mongodb.com/

ANNEXURE

Source code

1. HomeComponent.js

```
import React from "react";
import { Card, CardImg, CardText, CardBody, CardTitle } from "reactstrap";
import { Loading } from "../Loading/LoadingComponent";
import { baseUrl } from "../../shared/baseUrl";
import Feedback from "../Feedback/FeedbackComponent";

function RenderTopperContent({ toppers, isLoading, errMess }) {
  if (isLoading) {
    return <Loading />;
  } else if (errMess) {
    return <h4>{errMess}</h4>;
  } else
    return toppers.toppers.map((topper) => (
      <RenderTopperItem key={topper.id} top={topper} />
    ));
}

function RenderPlacementContent({ placements, isLoading, errMess }) {
  if (isLoading) {
    return <Loading />;
  } else if (errMess) {
    return <h4>{errMess}</h4>;
  } else
```

```
    return placements.placements.map((placement) => (
      <RenderPlacementItem key={placement.id} placemt={placement} />
    ));
}

function RenderTopperItem({ top }) {
  return (
    <Card>
      <CardImg src={baseUrl + top.image} alt={top.name} />
      <CardBody>
        <CardTitle>{top.name}</CardTitle>
        <CardText>{top.score}</CardText>
      </CardBody>
    </Card>
  );
}

function RenderPlacementItem({ placemt }) {
  return (
    <div>
      <div>
        <Card>
          <CardImg src={baseUrl + placemt.image} alt={placemt.name} />
          <CardBody>
            <CardTitle>{placemt.name}</CardTitle>
            <CardText>{placemt.company}</CardText>
          </CardBody>
```

```
        </Card>
      </div>
    </div>
  );
}

const Home = (props) => {
  return (
    <>
      <div className="jumbotron">
        <div className="container">
          <div className="row row-header">
            <div className="col-12 col-sm-6">
              <h1>Master of Computer Application</h1>
              <p>
                Master of Computer Application is the best course in Nehru
                College of Engineering and Research Centre.
              </p>
            </div>
          </div>
        </div>
      </div>
      <div className="container">
        <h3>TOPPERS</h3>
        <hr />
        <div className="d-flex p-2 justify-content-around flex-wrap">
          <RenderTopperContent
```

```
            toppers={props.topper}
            isLoading={props.toppersLoading}
            errMess={props.toppersErrMess}
          />
        </div>
      </div>
      <div className="container">
        <h3>PLACEMENTS</h3>
        <hr />
        <div className="d-flex p-2 justify-content-around flex-wrap">
          <RenderPlacementContent
            placements={props.placement}
            isLoading={props.placementsLoading}
            errMess={props.placementsErrMess}
          />
        </div>
      </div>
    </>
  );
};

export default Home;
```

2. PostComponent.js

```
import React from "react";
import { Breadcrumb, BreadcrumbItem, Card, CardBody, CardText,
CardHeader, CardImg, CardFooter } from 'reactstrap';
import { Link } from 'react-router-dom';
```

```
import { Loading } from '../Loading/LoadingComponent';
import { baseUrl } from '../../shared/baseUrl';

function RenderPostContent({posts}) {
   return (
      posts.posts.map(post => (
        <RenderPostItem key={post.id} post={post} />
      ))
     );
}

function RenderPostItem({post}){
   return(
      <Card className="mb-3">
         <CardHeader><h6>{post.name}</h6></CardHeader>
         {post.image ? <CardImg src={baseUrl + post.image} width="100%"
height="250px"/> : null}
               <CardBody>
                  <CardText>{post.content}
                  </CardText>
               </CardBody>
                <CardFooter className="text-right text-muted small"><p>
                   
                   {new Intl.DateTimeFormat('en-US', { year: 'numeric', month:
'short', day:'2-digit'}).format(new Date(Date.parse(post.updatedAt)))}
               </p></CardFooter>
      </Card>
   )
```

```
}

const Post = (props) => {
   return(
      <div className="container">
         <div className="row">
            <Breadcrumb className="m-1">
               <BreadcrumbItem><Link
to="/home">Home</Link></BreadcrumbItem>
               <BreadcrumbItem active>Posts</BreadcrumbItem>
            </Breadcrumb>
            <div className="col-12">
               <h3>Posts</h3>
               <hr />
            </div>
         </div>
         <div className="col-6 m-3">
         <div className="d-flex flex-column-reverse">
         <RenderPostContent
         posts={props.post}
         />
         </div>
         </div>
      </div>
   )
}

export default Post;
```

3. ContactComponent.js

```
import React, { Component } from 'react';
import { Link } from 'react-router-dom';
import { Breadcrumb, BreadcrumbItem, Button, Label, Col, Row } from 'reactstrap';
import { Control, Form, Errors } from 'react-redux-form';


const required= (val) => val && val.length;
const maxLength = (len) => (val) => !(val) || (val.length <= len);
const minLength = (len) => (val) => (val) && (val.length >= len);
const isNumber = (val) => !isNaN(Number(val));
const validEmail = (val) => /^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$/i.test(val);


class Contact extends Component{
   constructor(props){
      super(props);


      this.handleSubmit = this.handleSubmit.bind(this);
   }
   handleSubmit(values){
      // console.log("Current state is: " + JSON.stringify(values))
      // alert("Current state is: " + JSON.stringify(values));
      this.props.resetFeedbackForm();
      this.props.postFeedback( values.firstname, values.lastname, values.telnum, values.email, values.agree, values.contactType, values.message);
   }
   render(){
```

```
    return(
        <>
        <div className='container'>
            <div className='row'>
                <Breadcrumb className="m-1">
                    <BreadcrumbItem><Link
to="/home">Home</Link></BreadcrumbItem>
                    <BreadcrumbItem active>Contact Us</BreadcrumbItem>
                </Breadcrumb>
                <div className="col-12">
                <h3>Contact US</h3>
                <hr />
                </div>
            </div>
            <div className="row row-content">
                <div className="col-12">
                    <h3>Office</h3>
                </div>
                <div className="col-12 col-sm-4 offset-sm-1">
                    <h5>Nehru College of Engineering and Research Centre</h5>
                    <address>
                        Nila Gardens, Pampady, Thiruvilwamala<br/>
                        Thrissur, Kerala, India<br/>
                        PIN: 680 588<br/>
                        <i className="fa fa-phone fa-lg"></i>: +9199 1234 5678<br />
                                <i className="fa fa-fax fa-lg"></i>: +9199 8765
4321<br />
```

```
                        <i className="fa fa-envelope fa-lg"></i>: <a
href="mailto:mca@ncerc">
                         mca@ncerc.in</a>
                    </address>
                </div>
                <div className="col-12 col-sm-6 offset-sm-1">
                    <h5>Random Pic or Something</h5>
                </div>
            </div>
            <div className="row row-content">
                <div className="col-12">
                    <h3>MCA Department</h3>
                </div>
                <div className="col-12 col-sm-4 offset-sm-1">
                    <h5>Department</h5>
                    <address>
                        Kapila Block<br/>
                        Nila Gardens, Pampady, Thiruvilwamala<br/>
                        Thrissur, Kerala, India<br/>
                        PIN: 680 588<br/>
                        <i className="fa fa-phone fa-lg"></i>: +9199 1234 5678<br />
                                <i className="fa fa-fax fa-lg"></i>: +9199 8765
4321<br />
                                <i className="fa fa-envelope fa-lg"></i>: <a
href="mailto:mcahod@ncerc">
                        departmentmca@ncerc.in</a>
                    </address>
                </div>
```

```
            <div className="col-12 col-sm-6 offset-sm-1">
                <h5>Random Pic or Something</h5>
            </div>
        </div>
        <div className="row row-content">
            <div className="col-12">
                <h3>Admission Queries</h3>
            </div>
            <div className="col-12 col-sm-11 offset-sm-1">
                <h5>Officer in Charge: name</h5><br/>
                <div className="btn-group" role="group">
                    <a role="button" className="btn btn-primary"
href="tel:+919912345678"><i className="fa fa-phone"></i> Call</a>
                    <a role="button" className="btn btn-success"
href="mailto:admissions@ncerc.in"><i className="fa fa-envelope-o"></i>
Email</a>
                </div>
            </div>
        </div>
        <div className="row row-content">
            <div className="col-12">
                <h3>Send Us Your Feedback & Queries</h3>
                <hr />
            </div>
            <div className="col-12 col-md-9">
                <Form model="feedback" onSubmit={(values) =>
this.handleSubmit(values)}>
                    <Row className="form-group">
                        <Label htmlFor="firstname" md={2}>First Name</Label>
```

```
                    <Col md={10}>
                        <Control.text model=".firstname" id="firstname"
name="firstname" placeholder="First Name" className="form-control"
                        validators={{
                            required, minLength: minLength(3), maxLength:
maxLength(15)
                        }} />
                    <Errors className="text-danger" model=".firstname"
show="touched" messages={{
                        required: 'Required',
                        minLength: 'Must be greater than 2 characters',
                        maxLength: 'Must be 15 characters or less'
                    }}/>
                    </Col>
                </Row>
                <Row className="form-group">
                    <Label htmlFor="lastname" md={2}>Last Name</Label>
                    <Col md={10}>
                        <Control.text model=".lastname" id="lastname"
name="lastname" placeholder="Last Name" className="form-control"
                        validators={{
                            required, minLength: minLength(3), maxLength:
maxLength(15)
                        }} />
                        <Errors className="text-danger" model=".lastname"
show="touched" messages={{
                        required: 'Required',
                        minLength: 'Must be greater than 2 characters',
                        maxLength: 'Must be 15 characters or less'
```

```
                    }}/>
                    </Col>
                </Row>
                <Row className="form-group">
                    <Label htmlFor="telnum" md={2}>Contact
Number</Label>
                    <Col md={10}>
                        <Control.text model=".telnum" id="telnum"
name="telnum" placeholder="Contact Number" className="form-control"
                        validators={{
                            required, minLength: minLength(3), maxLength:
maxLength(15), isNumber
                        }}
                        />
                        <Errors className="text-danger" model=".telnum"
show="touched" messages={{
                        required: 'Required',
                        minLength: 'Must be greater than 2 Numbers',
                        maxLength: 'Must be 15 Numbers or less',
                        isNumber: 'Must be a Number'
                    }}/>
                    </Col>
                </Row>
                <Row className="form-group">
                    <Label htmlFor="email" md={2}>Email</Label>
                    <Col md={10}>
                        <Control.text model=".email" id="email" name="email"
placeholder="Email" className="form-control"
                        validators={{
```

```
                        required, validEmail
                    }} />
                    <Errors className="text-danger" model=".email"
show="touched" messages={{
                    required: 'Required ',
                    validEmail: 'Invalid Email Address'
                }} />
                </Col>
            </Row>
            <Row className="form-group">
                <Label htmlFor="message" md={2}>Your
Feedback/Query</Label>
                <Col md={10}>
                    <Control.textarea model=".message" id="message"
name="message" rows="12" className="form-control" />
                </Col>
            </Row>
            <Row className="form-group">
                <Col md={{size: 10, offset: 2}}>
                    <Button type="submit" color="primary">
                        Send
                    </Button>
                </Col>
            </Row>
        </Form>
    </div>
</div>
</div>
```

```
      </>
    )
}
}
```

4. LoginComponent.js

```
import React, { Component } from "react";
import {
  Row,
  Col,
  Label,
  Button,
  Input,
  Form,
  FormGroup
} from "reactstrap";

const required = (val) => val && val.length;

class Login extends Component {
  constructor(props) {
    super(props);
    this.state = {

    };
    this.handleLogin = this.handleLogin.bind(this);
  }
```

```
  handleLogin(event) {
    this.props.loginUser({username: this.username.value, password:
this.password.value});
    event.preventDefault();
}



  render() {
    return (
      <>
          <div className="container">
            <div className="row row-content">
              <div className="col-12">

                  {/* Login Form Starts from here */}

                   <Form onSubmit={this.handleLogin}>
                   <FormGroup>
                   <Row className="form-group">
                    <Label htmlFor="username" md={2}>Username</Label>
                    <Col md={12}>
                    <Input
                    type="text"
                      id="username"
                      name="username"
                      placeholder="Username"
                      innerRef={(input) => this.username = input}
```

```
  />
  </Col>
</Row>
</FormGroup>
<FormGroup>
<Row className="form-group">
  <Label htmlFor="password" md={2}>Password</Label>
  <Col md={12}>
  <Input
    type="password"
    id="password"
    name="password"
    placeholder="Password"
    className="form-control"
    innerRef={(input) => this.password = input}
  />
  </Col>
</Row>
</FormGroup>
<FormGroup check>
<Row className="form-group">
  <Col md={12}>
    <div className="form-check">
    <Label check>
    <Input type="checkbox" name="remember"
          innerRef={(input) => this.remember = input}  />
    {/* {' '} */}
```

```
                    Remember Me
                  </Label>
                   </div>
                  </Col>
                 </Row>
                 </FormGroup>
                 <Row className="form-group">
                 <Col md={2}>
                    <Button type="submit"value="submit" color="primary">
                       Login
                    </Button>
                 </Col>
               </Row>
                 </Form>
             </div>
           </div>
         </div>
     </>
   );
  }
}

export default Login;
```

7.3 Output

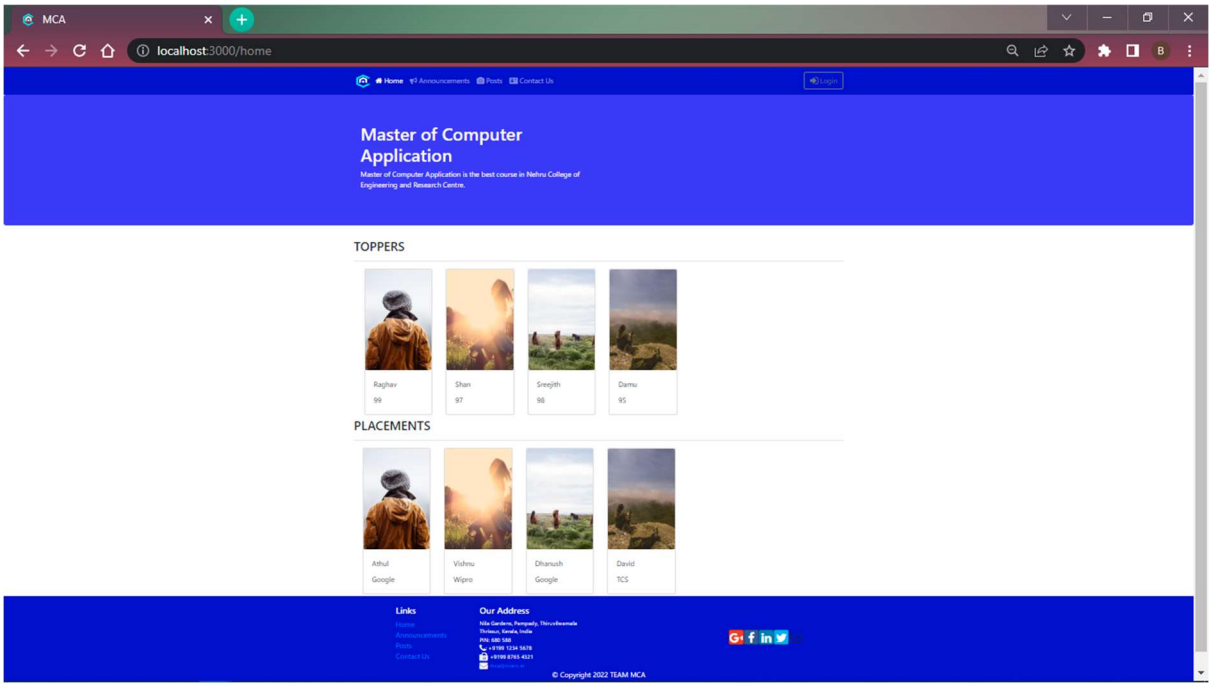

Figure 7.1: Homepage

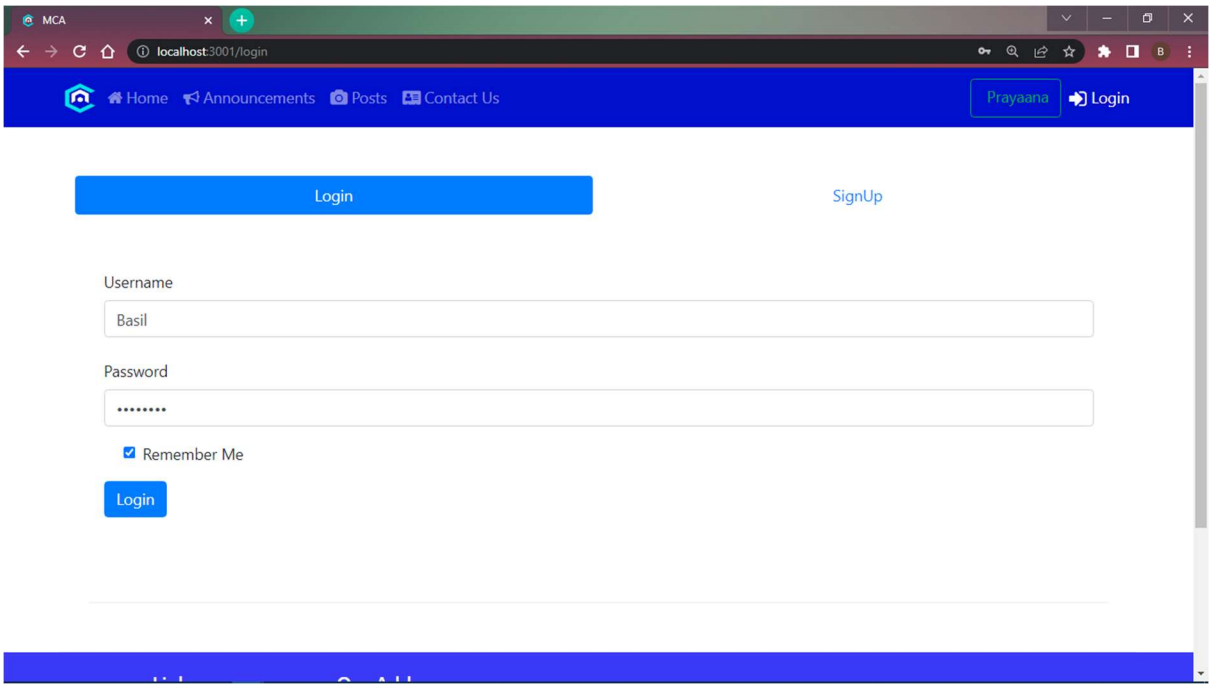

Figure 7.2: Login

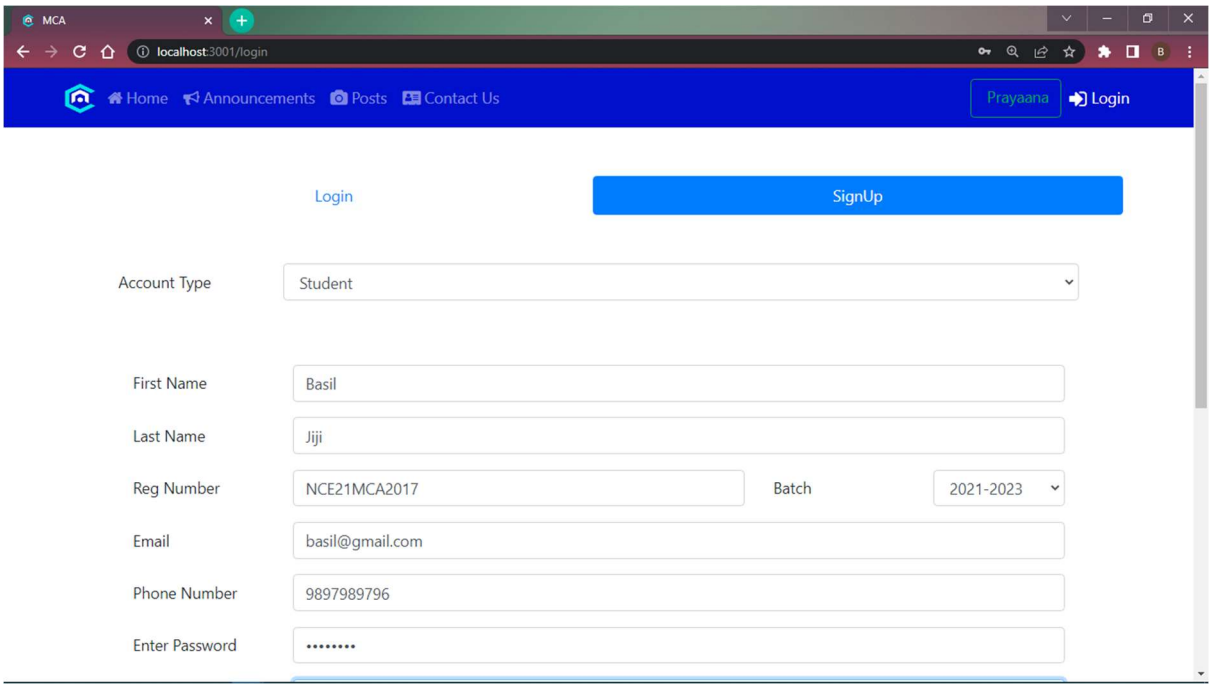

Figure 7.3: Signup

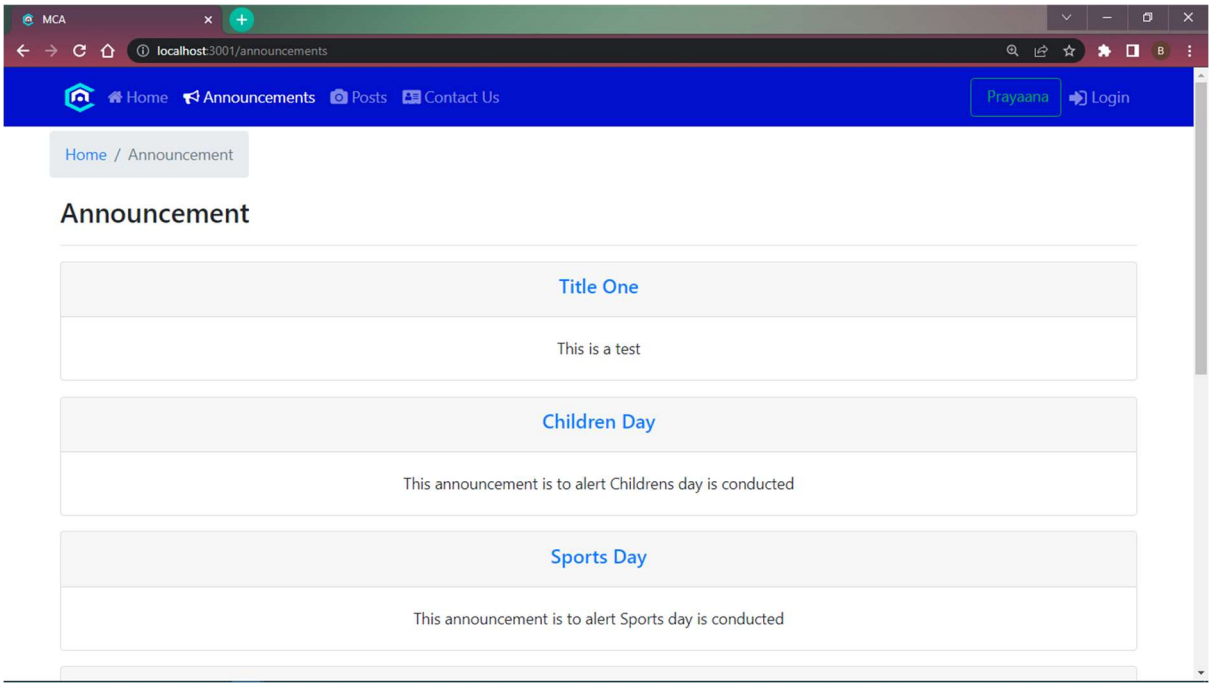

Figure 7.4: Announcements

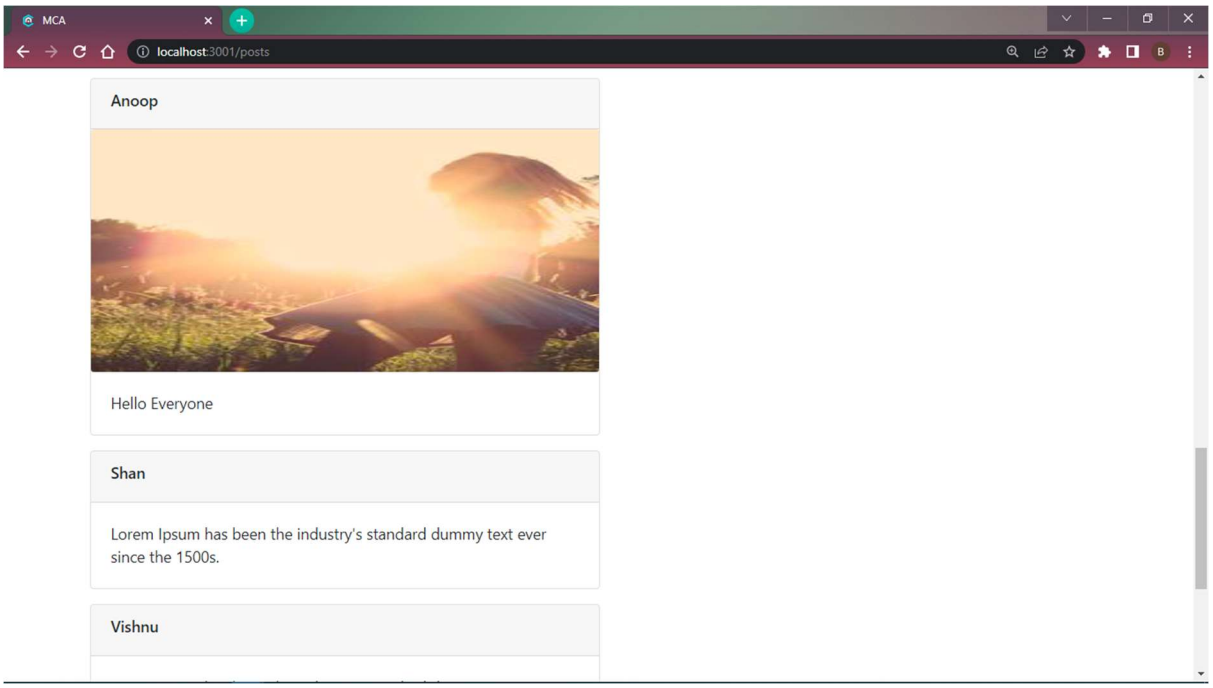

Figure 7.5: Posts

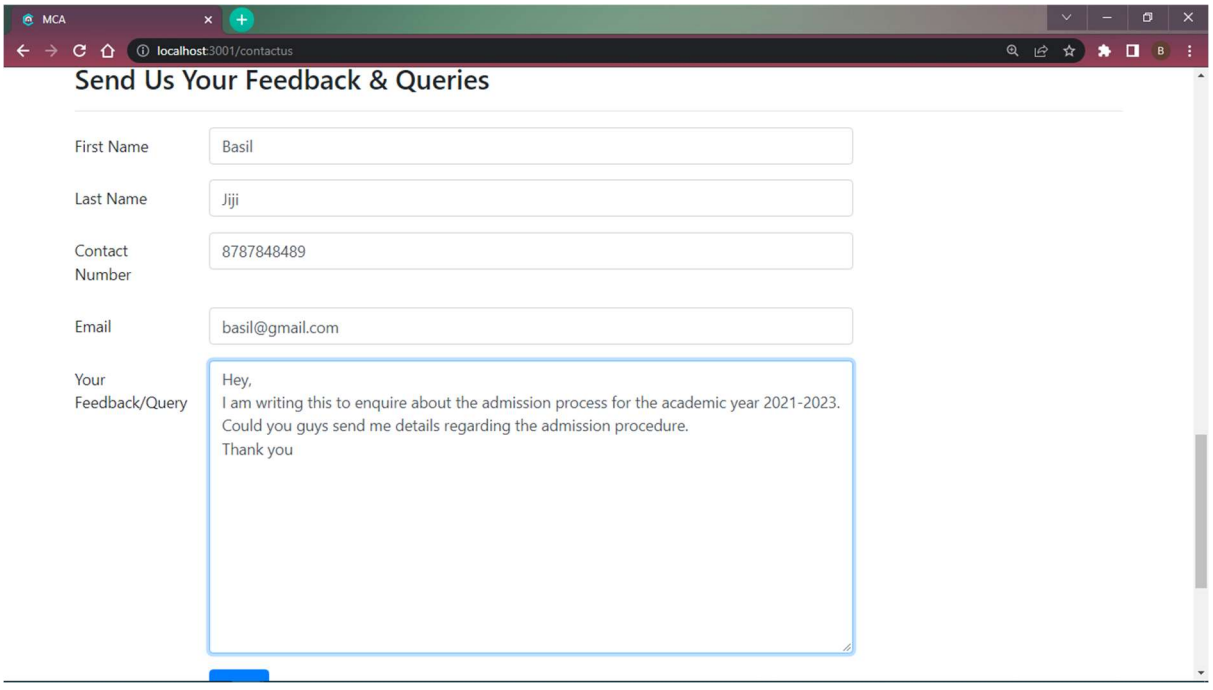

Figure 7.6: Feedback

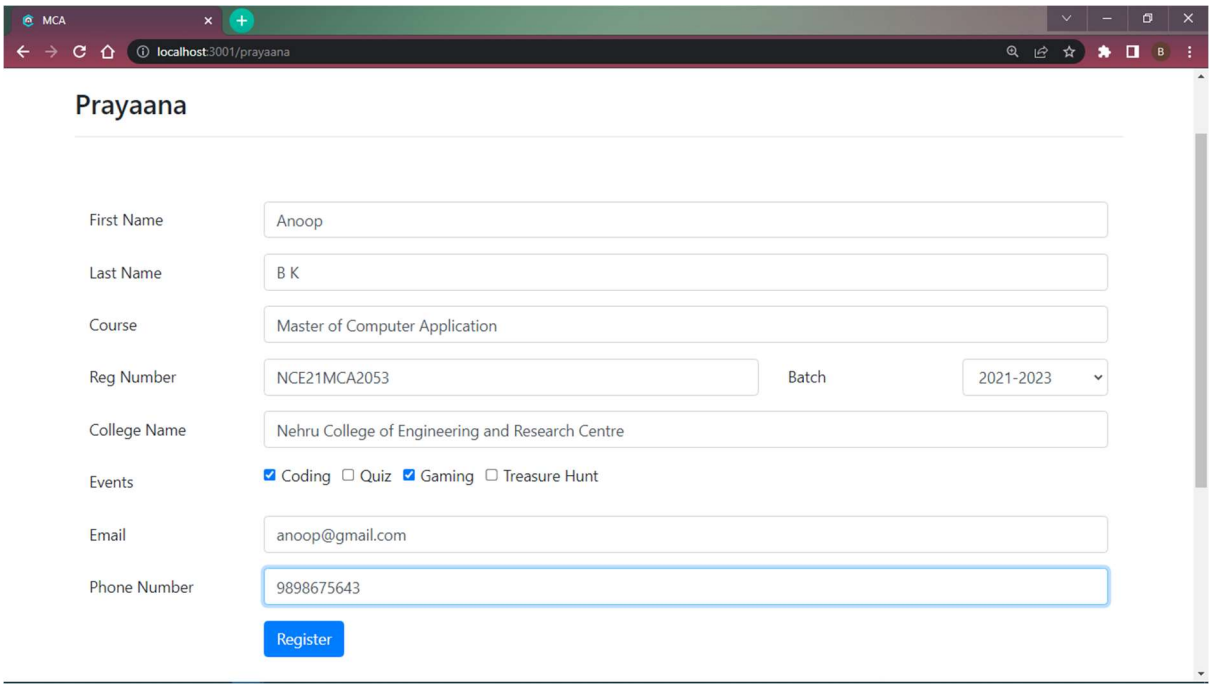

Figure 7.7: Prayaana

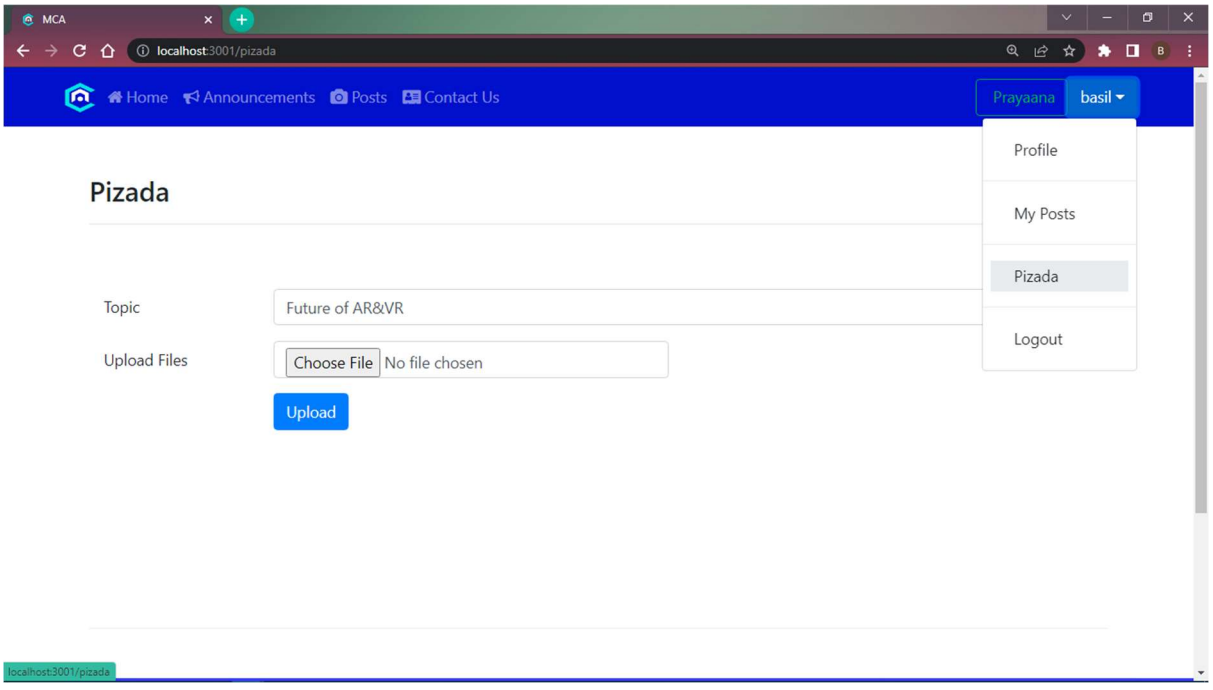

Figure 7.8: Pizada

7.4 Git History

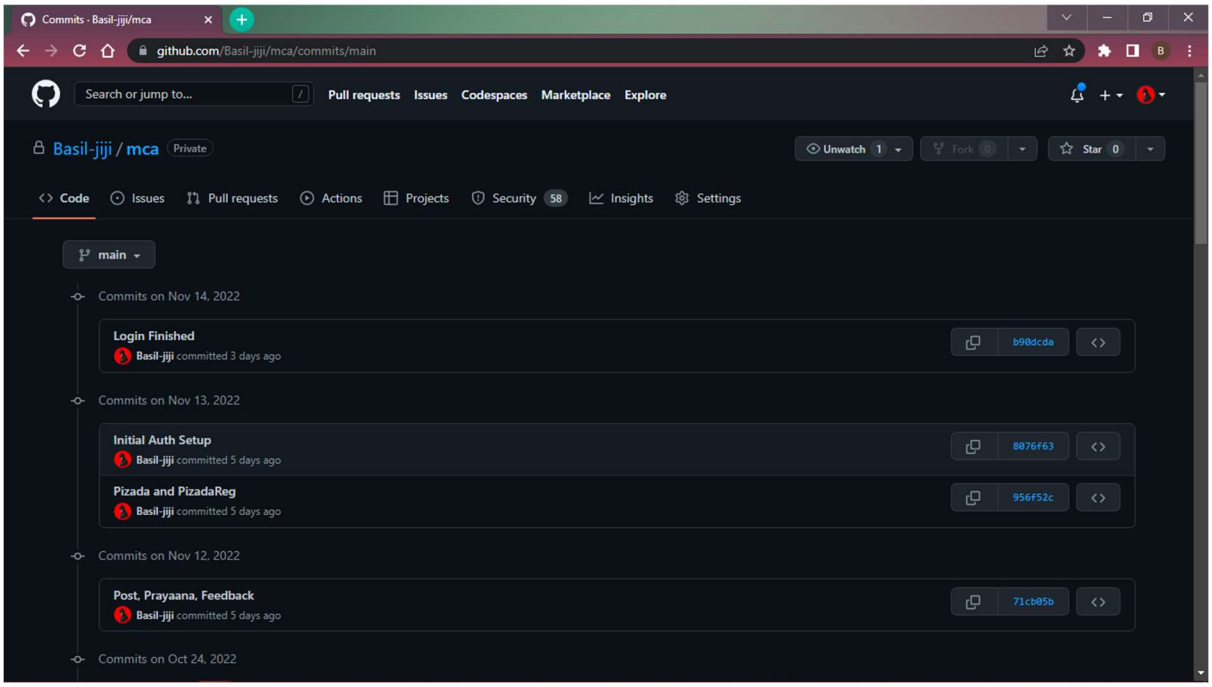

Figure 7.9: Git History