



B.Sc (Hons) in Information Technology Specializing in Software Engineering

Year 3 - 2022

SE3040 – Application Frameworks

Group Project: Technical Report

Group Name:

Group Member Details

Student ID number	Name	SLIIT Email
IT20133672	C.D. Adhihetty	it20133672@my.sliit.lk
IT20133122	R.A.E. Wijenayake	it20133122@my.sliit.lk
IT20135966	Gunasekara A.M.P.P.	it20135966@my.sliit.lk
IT20133818	Perera M.P.M.	it20133818@my.sliit.lk

Git Repository: <https://github.com/Chathum10/Research-Project-Management-Tool.git>

Group Presentation Video: https://mysliit-my.sharepoint.com/:f/g/personal/it20133672_my_sliit_lk/EghixPrBCK1Fl3CJLd6Mc30Bx7qEe14OOBL6KXeicQ-58w?e=O0BprU

Project Start Date: 15th April 2022

Project End Date: 04th June 2022

Project Declaration:

I the undersigned solemnly declare that the project technical based on our group members own work carried out during the course of our study under the supervision of Mr. Thusithanana Thilakarathna.

A handwritten signature in black ink, appearing to read 'Dinesh' or 'Adhihetty'.

Group Leader: C.D. Adhihetty, IT20133672

Contents

1. Component Description	2
i. User management and document management – IT20133672 C.D. Adhihetty	2
ii. Student group management and panel management – IT20133122 R.A.E. Wijenayake	3
iii. Topic registration management – IT20135966 Gunasekara A.M.P.P.....	3
iv. Final document submission & Chat management – IT20133818 Perera M.P.M.....	3
2. Screenshots of the Frontend.....	4
i. User management and document management – IT20133672 C.D. Adhihetty	4
ii. Student group management and panel management – IT20133122 R.A.E. Wijenayake	15
iii. Topic registration management – IT20135966 Gunasekara A.M.P.P.....	20
iv. Final document submission management & Chat management – IT20133818 Perera M.P.M.	24
3. Screenshots/ Code Snippets of the Backend.....	28
i. User management and document management – IT20133672 C.D. Adhihetty	28
ii. Student group management and panel management – IT20133122 R.A.E. Wijenayake	42
iii. Topic registration management – IT20135966 Gunasekara A.M.P.P.....	48
iv. Final document submission management & Chat management – IT20133818 Perera M.P.M.	54
4. Screenshots of the MongoDB Schemas	59
i. User management and document management – IT20133672 C.D. Adhihetty	59
ii. Student group management and panel management – IT20133122 R.A.E. Wijenayake	61
iii. Topic registration management – IT20135966 Gunasekara A.M.P.P.....	62
iv. Final document submission management & Chat management – IT20133818 Perera M.P.M.	63

1. Component Description

- i. User management and document management – IT20133672 C.D. Adhihetty

1. Students, supervisors/co-supervisors and panel members can come to the tool and register to the system. After that they can log in. Admin can view all the user profiles and update, delete profiles. Admin can access to all the dashboards.
 2. Admin upload document templet and presentation guidelines to student and students can download those.
 3. Admin upload marking scheme documentation and supervisors can download it.
 4. Admin can upload presentation marking schema and panel members can download it.
-
- ii. Student group management and panel management – IT20133122 R.A.E. Wijenayake
 1. Students can add group members, register their groups and admin can view all the groups details.
 2. Admin can allocate a panel to the each and every groups.
 3. Admin can view all the panel members and create panels.
 4. Panel members can view their panels and check their details.

 - iii. Topic registration management – IT20135966 Gunasekara A.M.P.P.
 1. Student group send the topic to the supervisor and if supervisor accept it co-supervisor can choose and assign to suitable research group.
 2. After selecting supervisor and co-supervisor student upload the topic registration document to the relevant panel.
 3. Panel member can download document and view it.
 4. After the panel member accept or ignore the topic and student can check the status of project and if it is accepted student can go with research topic and if it is ignored student have to do the all the previous process again until the panel acceptance.

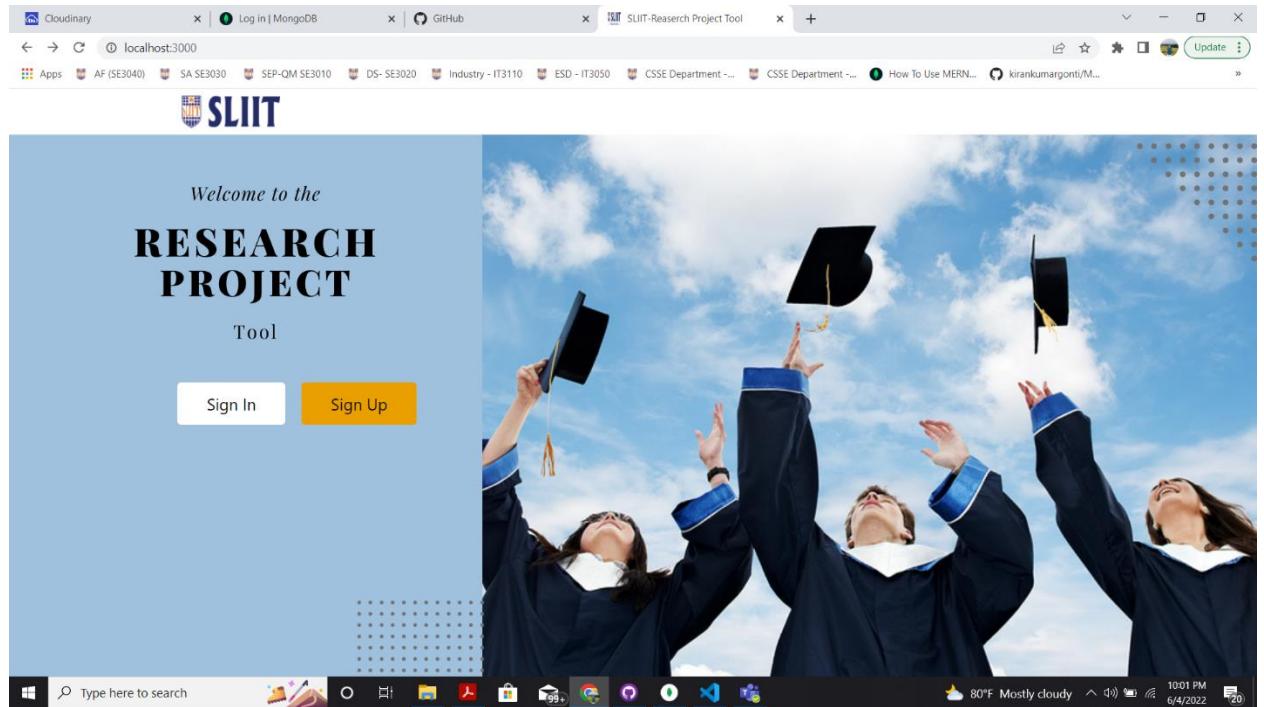
 - iv. Final document submission & Chat management – IT20133818 Perera M.P.M.
 1. Student upload the final research documentation according to the given templet by admin.
 2. Supervisor can download the final documentation and check it.

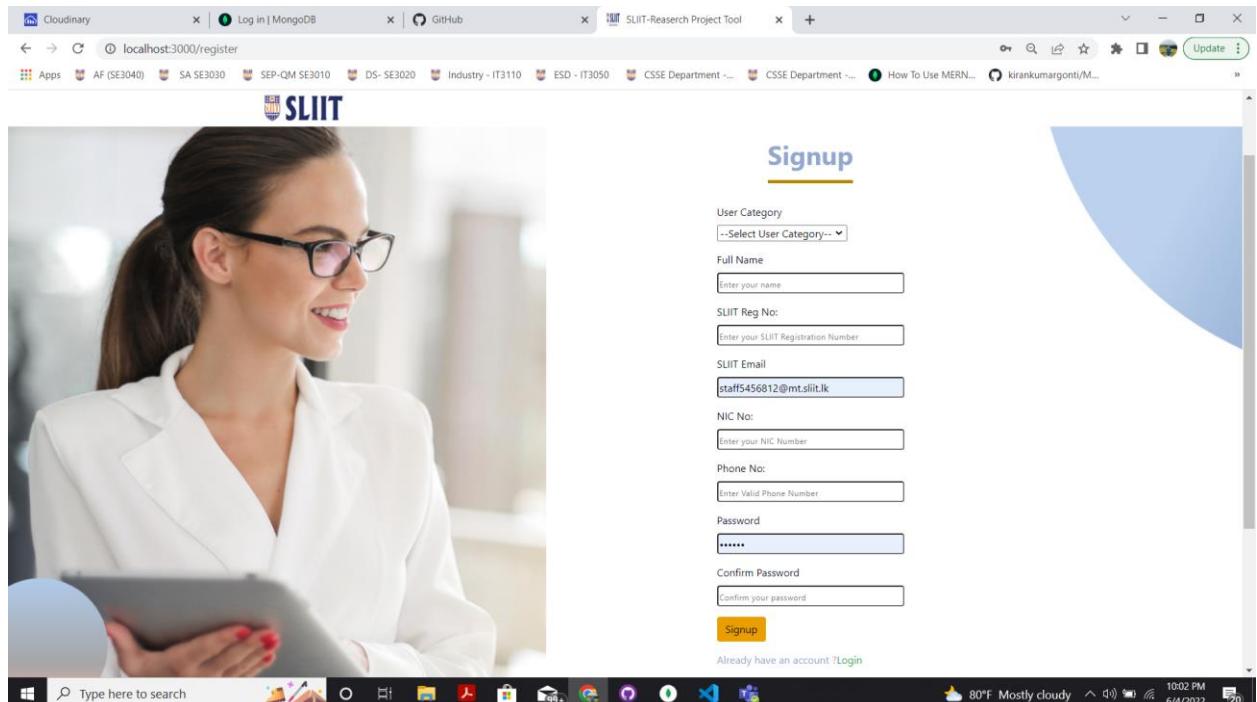
3. Admin assign a chat id to the each and every student groups.
4. Student groups, supervisors and co-supervisors can view chat id and they can navigate to a chat room and chat with relevant groups.

2. Screenshots of the Frontend

- i. User management and document management – IT20133672 C.D. Adhiketty

User management





The screenshot shows the 'Signup' page of the SLIIT Research Project Tool. The page features a background image of a smiling woman in a white blazer and glasses. On the right, there is a form with fields for User Category, Full Name, SLIIT Reg No., SLIIT Email, NIC No., Phone No., Password, and Confirm Password. A 'Signup' button is at the bottom. Below the form, a link says 'Already have an account? Login'.

Signup

User Category
-Select User Category--

Full Name
Enter your name

SLIIT Reg No:
Enter your SLIIT Registration Number

SLIIT Email
staff5456812@mt.sliit.lk

NIC No:
Enter your NIC Number

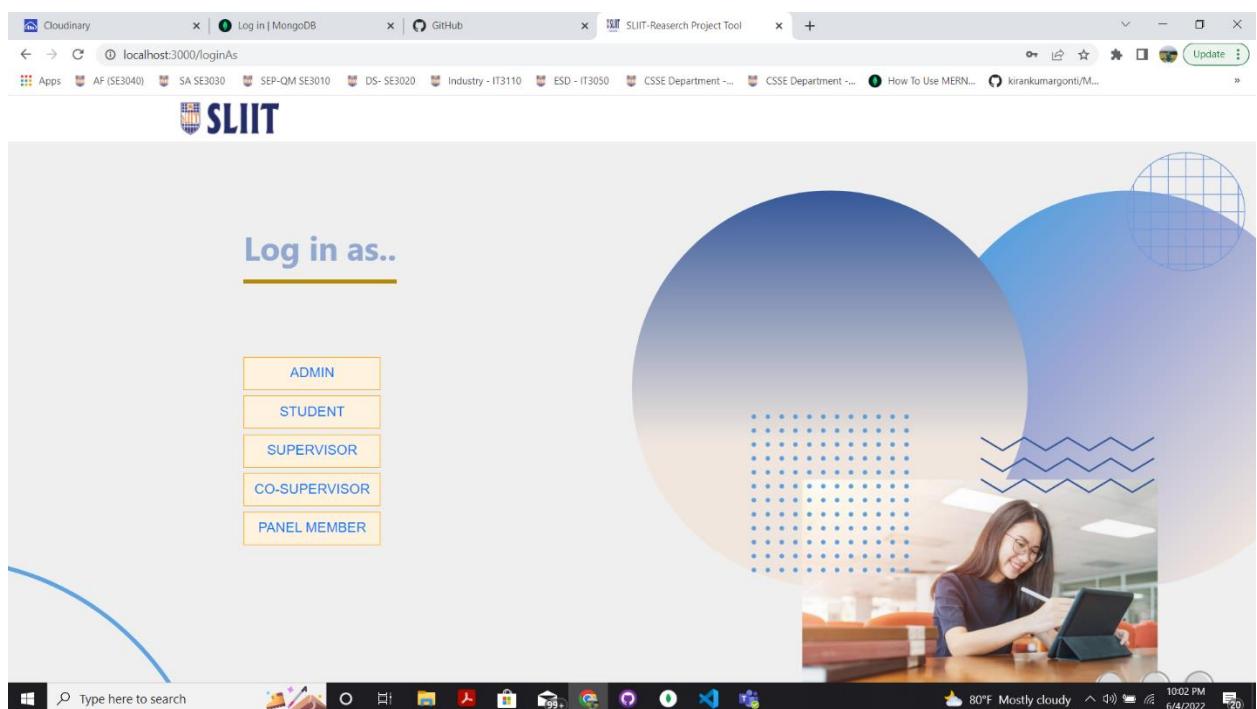
Phone No:
Enter Valid Phone Number

Password
.....

Confirm Password
Confirm your password

Signup

Already have an account? [Login](#)

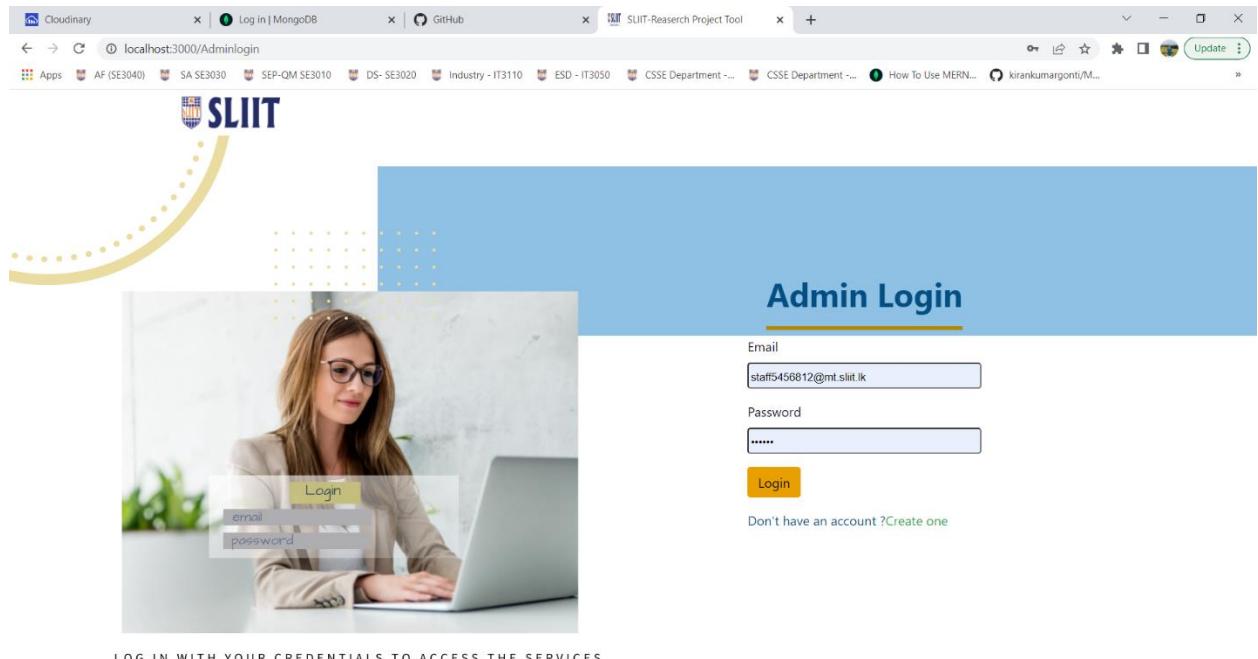


The screenshot shows the 'Log in as..' page of the SLIIT Research Project Tool. It features a background image of a person working on a tablet. On the left, there is a vertical list of user roles: ADMIN, STUDENT, SUPERVISOR, CO-SUPERVISOR, and PANEL MEMBER. Each role is represented by a yellow rectangular button with blue text.

Log in as..

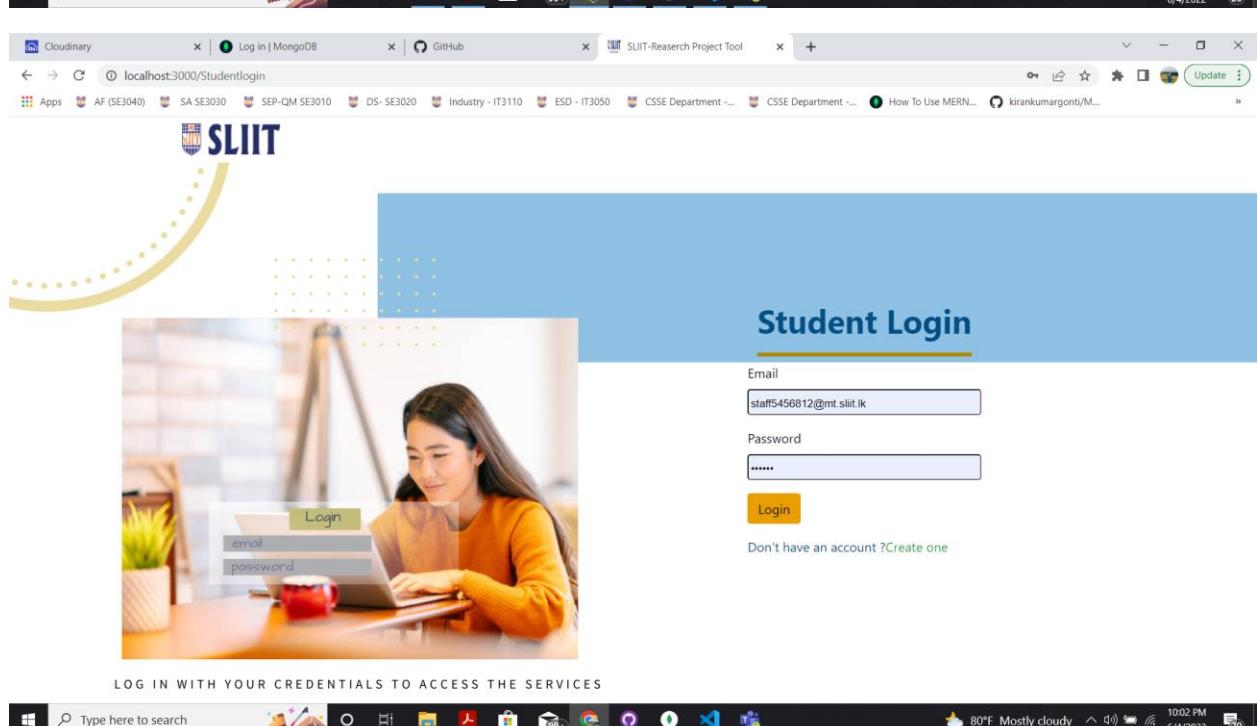
ADMIN
STUDENT
SUPERVISOR
CO-SUPERVISOR
PANEL MEMBER

80°F Mostly cloudy 1002 PM 6/4/2022



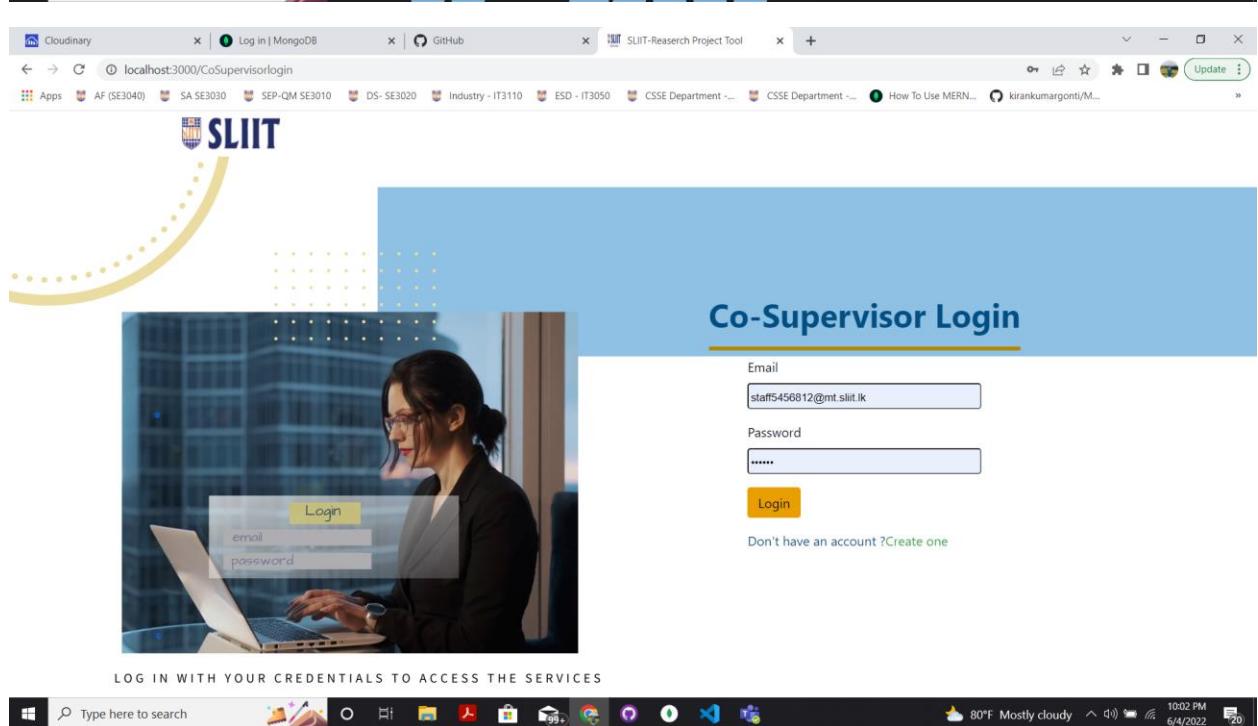
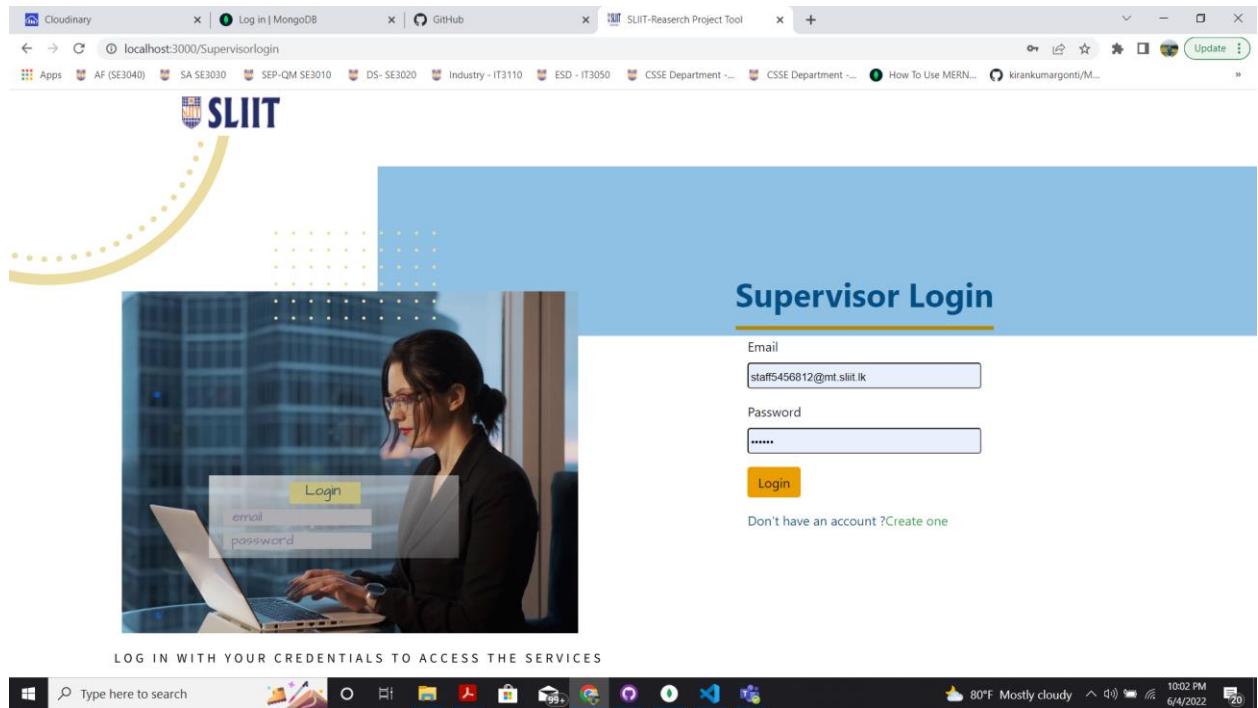
The screenshot shows the Admin Login page of a web application. The header features the SLIIT logo and a decorative background image of a person working at a laptop. The main title is "Admin Login". Below it, there are fields for "Email" (containing "staff5456812@mt.sliit.lk") and "Password" (containing "*****"). A yellow "Login" button is present. At the bottom, a link says "Don't have an account? Create one".

LOG IN WITH YOUR CREDENTIALS TO ACCESS THE SERVICES



The screenshot shows the Student Login page of the same web application. The layout is identical to the Admin Login page, with the SLIIT logo, a background image of a person working at a laptop, and the title "Student Login". The "Email" field contains "staff5456812@mt.sliit.lk" and the "Password" field contains "*****". A yellow "Login" button is visible. A link at the bottom suggests creating a new account.

LOG IN WITH YOUR CREDENTIALS TO ACCESS THE SERVICES



The screenshot shows a web browser window titled "SLIIT-Reaserch Project Tool". The URL is "localhost:3000/PannelMemberlogin". The page features the SLIIT logo at the top left. A large blue header bar contains the text "Pannel Member Login". Below the header is a photograph of a man in a suit working at a desk with a computer. Overlaid on the photo is a login form with fields for "Email" (containing "staff5456812@mt.sliit.lk") and "Password" (containing "*****"). A yellow "Login" button is visible. Below the form is a link "Don't have an account? Create one". At the bottom of the page, a banner reads "LOG IN WITH YOUR CREDENTIALS TO ACCESS THE SERVICES".

The screenshot shows a web browser window titled "SLIIT-Reaserch Project Tool". The URL is "localhost:3000/AdminDashboard". The page features the SLIIT logo at the top left. A large blue header bar contains the text "Hi! Jalini" and "Welcome Back Admin". Below the header is a grid of six buttons arranged in two rows of three. The buttons are: "User Management", "Document Management", "Student Group Management", "Pannel Management", "View Roles", and "Chat Management". A red "Logout" button is located in the top right corner of the dashboard area.

The screenshot shows a web browser window titled "localhost:3000/userProfiles". The page features the SLIIT logo at the top left. A blue header bar contains the text "RESEARCH PROJECT TOOL". Below the header is a search bar with the placeholder "Type here to search". The rest of the page is mostly blank, showing the Windows taskbar at the bottom.

Screenshot of the SLIIT Research Project Tool interface showing the "All User Profiles" page.

The page displays a table of user profiles with the following columns:

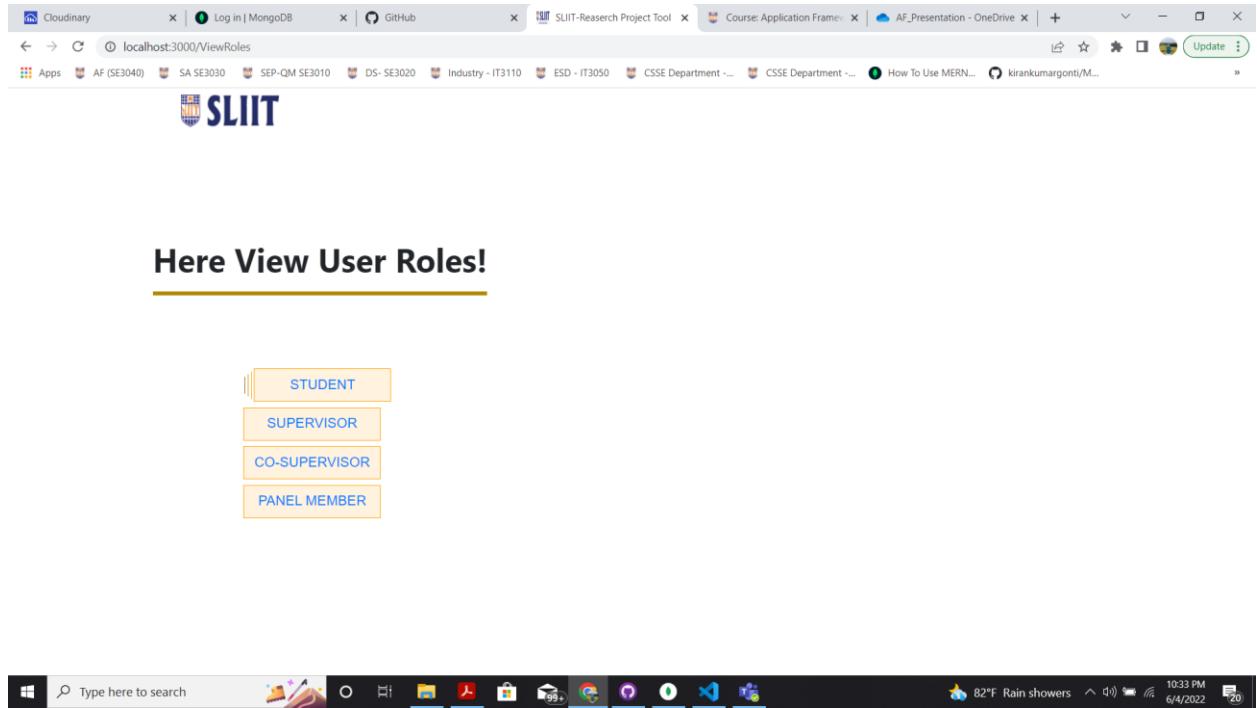
#	User ID	Full Name	SLIIT Email	NIC	Contact Number	Date	User Category	Action
1	it20133672	Chathum Dewanmina Adhihetty	it20133672@my.sliit.lk	200009602428	0773825119	2022-06-01T00:00:00.000Z	Student	
2	staff987615	Dharsha Adhihetty	staff987615@mt.sliit.lk	199909602428	0773221876	2022-06-04T15:36:20.359Z	Supervisor	
3	staff121267	Pawani Perera	staff121267@mt.sliit.lk	20009876678	0765453122	2022-06-04T15:37:18.275Z	Co-Supervisor	
4	staff5456812	Jalini Suriarachchi	staff5456812@mt.sliit.lk	1900987667v	0712228976	2022-06-04T15:38:55.239Z	Pannel Member	
5	IT20135966	Pramoda Piyumali	it20135966@my.sliit.lk	996712575v	0789879234	2022-06-04T15:50:58.091Z	Student	
6	IT20133818	Pawani Perera	it20133818@my.sliit.lk	207812784v	0778452478	2022-06-04T15:52:44.690Z	Student	
7	IT20133122	Ashen Rajapaksha	it20133122@my.sliit.lk	204563247v	0761457825	2022-06-04T15:54:18.028Z	Student	

Screenshot of the SLIIT Research Project Tool interface showing the "Edit User Profile" page.

The form fields are as follows:

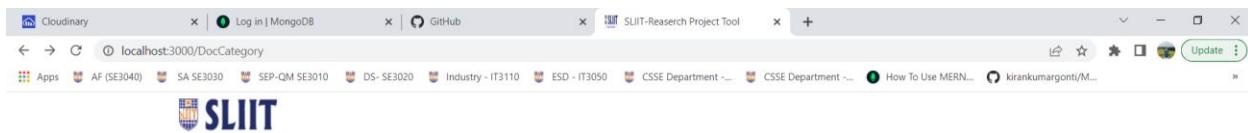
- SLIIT ID: it20133672
- SLIIT Email: it20133672@my.sliit.lk
- User Category: Student
- Full Name: Chathum Dewanmina Adhihetty
- NIC: 200009602428
- Contact Number: 0773825119
- Updating Date: 2022-06-01T00:00:00.000Z

Save Changes button at the bottom right.



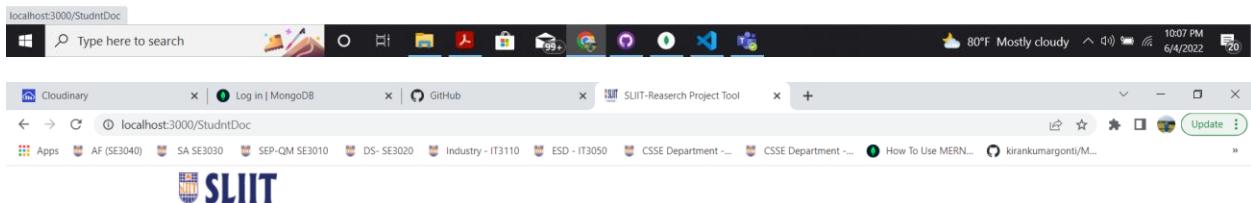
Document Management

A screenshot of a web browser window titled "SLIIT-Reaserch Project Tool". The address bar shows "localhost:3000/AdminDashboard". The page content features the SLIIT logo at the top left. In the center, a greeting "Hi! Jalini" and a message "Welcome Back Admin" are displayed. To the right is a red "Logout" button connected by a dotted line. Below the greeting is a grid of six rectangular buttons arranged in two columns of three: "User Management", "Document Management", "Student Group Management", "Panel Management", "View Roles", and "Chat Management". At the bottom of the page, the text "RESEARCH PROJECT TOOL" is centered. The browser's taskbar at the bottom shows various open tabs and system icons.



Upload Documents TO

STUDENT
SUPERVISOR
PANNEL MEMBER



Upload Documentation Templates For Students

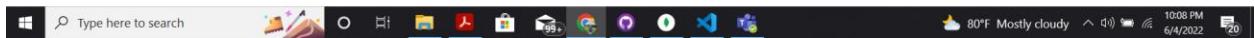
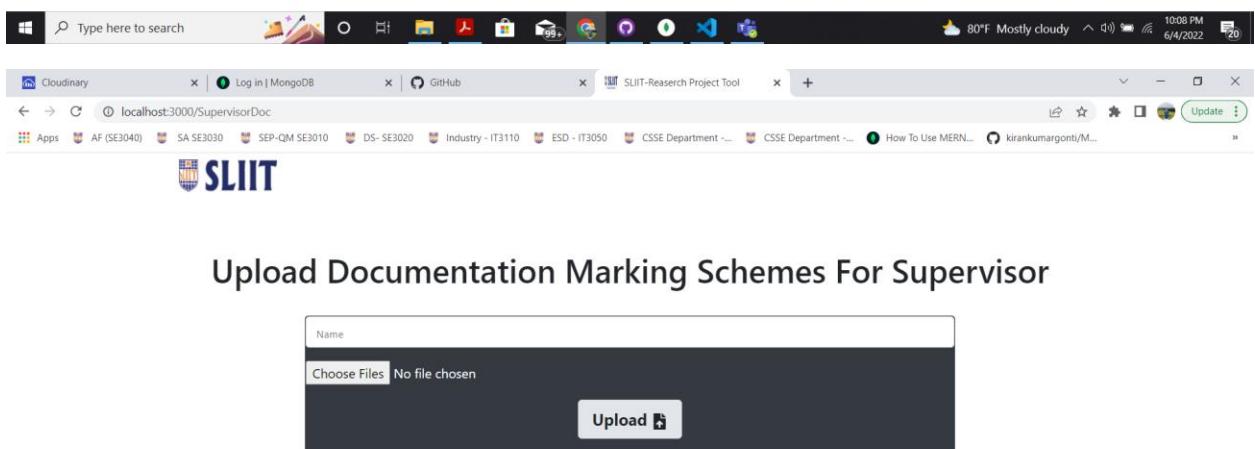
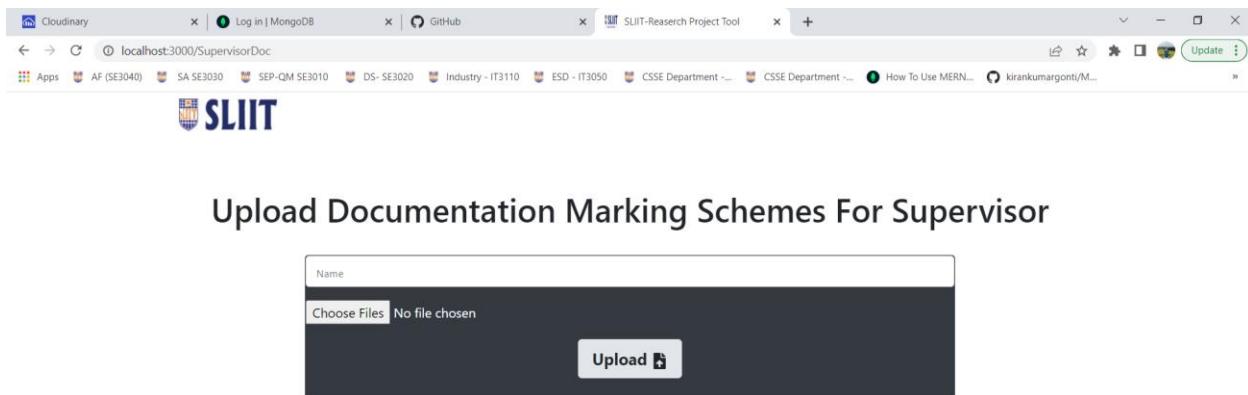
Name

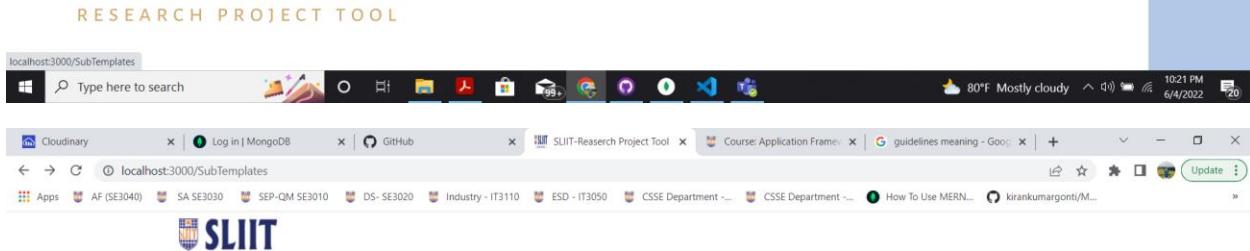
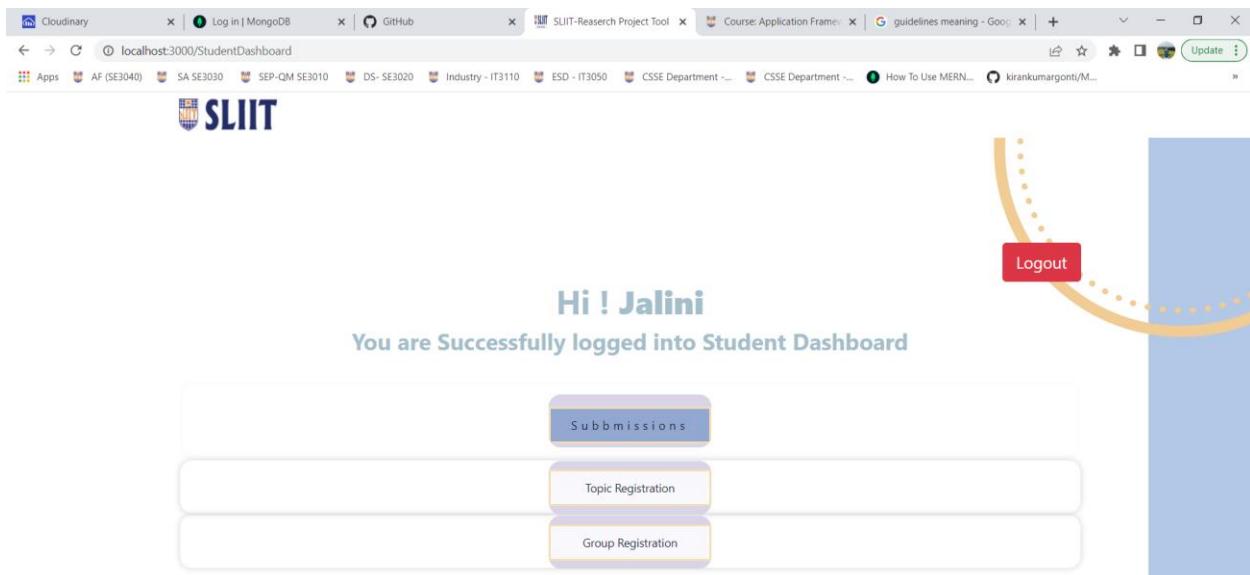
description

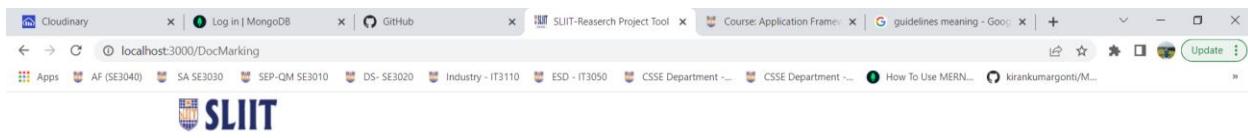
Choose Files No file chosen

Upload



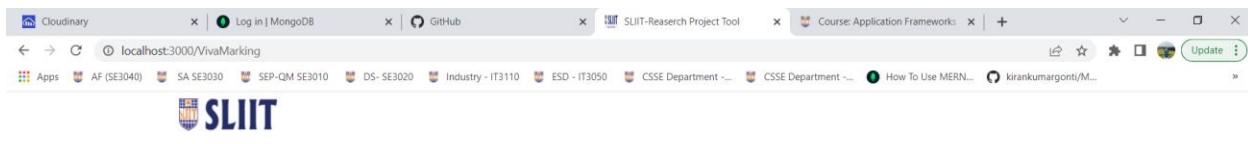
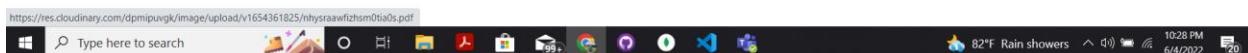






All Documentation Marking Schemes

MarkingScheme_Document_2021 [MarkingScheme_Document_2021](#)

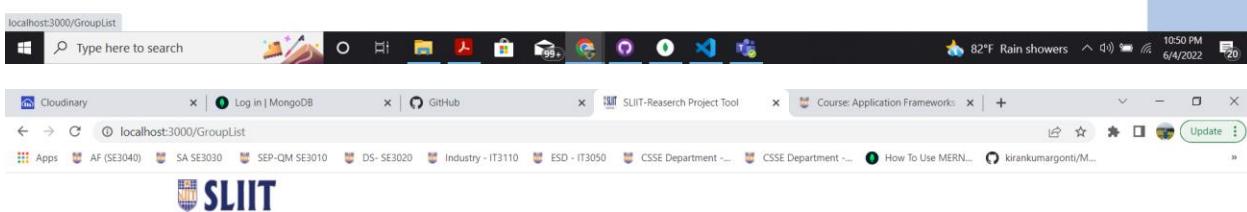
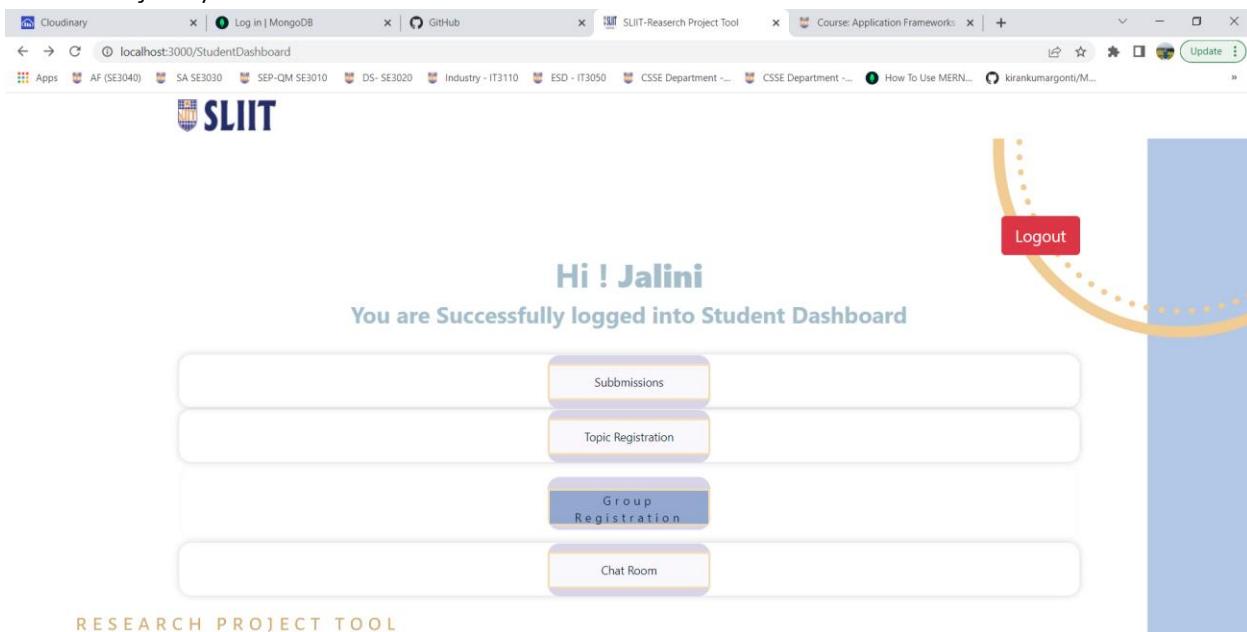


All Presentation Marking Schemes

markingScheme_Presentation_2021 [markingScheme_Presentation_2021](#)



ii. Student group management and panel management – IT20133122 R.A.E.
Wijenayake



Leader's ID	Member O1	Member O2	Member O3	Member O4	Group ID	Panel No.
IT20147856	IT20147856 Fernando P.M	IT20157823 Kahandawala A.T	IT20254812 Gunarathne P.P.T	IT20147856 Perera Y.M.N		
IT20186417	IT20186417 Arachchi A.H.M	IT20127836 Bentharla S.M.P	IT20107482 Gamage W.V.C	IT20254617 Balasooriya L.D		
IT20124867	IT20124867 Punsala S.N.Y	IT20271523 Prabhash Y.M.A	IT20145144 Dewpura S.M.P	IT2012781 Wijesinghe A.M		
IT20133672	IT20133672 Adhiketty C.D.	IT20133899 Jayawardana C.D.	IT20133818 Wijenayake A.D.	IT20133122 Perera M.P.M.		



Create New Group

Group leader's ID
Reg No.

Member Details

Member 01
Group Leader's Reg.No | Member's Name |

Member 02
Reg.No | Member's Name |

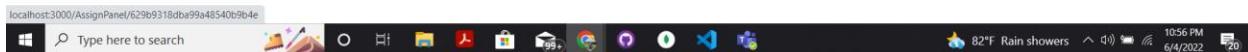
Member 03
Reg.No | Member's Name |

Member 04
Reg.No | Member's Name |

Register this group

All Research groups

Leader's ID	Member O1	Member O2	Member O3	Member O4	Group ID	Panel No.	Action
IT20147856	IT20147856 Fernando P.M	IT20157823 Kahandawala A.T	IT20254812 Gunarathne P.P.T	IT20147856 Perera Y.M.N			Assign a panel
IT20186417	IT20186417 Arachchi A.H.M	IT20127836 Benthera S.M.P	IT20107482 Gamage W.V.C	IT20254617 Balasooriya L.D			Assign a panel
IT20124867	IT20124867 Punnsala S.N.Y	IT20271523 Prabhash Y.M.A	IT20145144 Dewpura S.M.P	IT2012781 Wijesinghe A.M			Assign a panel
IT20133672	IT20133672 Adhihetty C.D.	IT20133899 Jayawardana C.D.	IT20133818 Wijenayake A.D.	IT20133122 Perera M.P.M.			Assign a panel



Assign Panel Members

Team Details

Assign A Group ID
[Generate Group ID]
DA-2022-Feb-IT20133672

Assign Panel
--Select a Panel--

Member Details

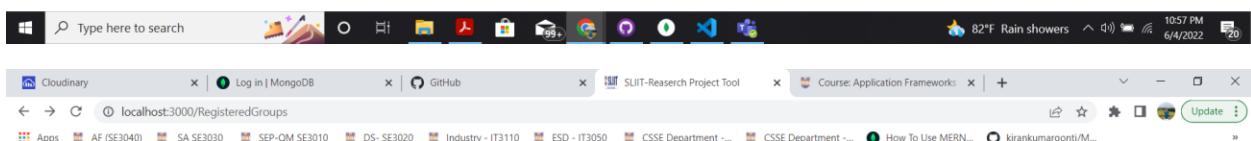
Member 01
IT20147856 | Fernando P.M.

Member 02
IT20157823 | Kahandawala A.T.

Member 03
IT20254812 | Gunarathne P.P.T

Member 04
IT20147856 | Perera Y.M.N

Assign



All Research groups

Leader's ID	Member O1	Member O2	Member O3	Member O4	Group ID	Panel No.	Action
IT20147856	IT20147856 Fernando P.M	IT20157823 Kahandawala A.T	IT20254812 Gunarathne P.P.T	IT20147856 Perera Y.M.N	2022_Feb_IT20147856_	Panel 01	Assign a panel
IT20186417	IT20186417 Arachchi A.H.M	IT20127836 Benthara S.M.P	IT20107482 Gamage W.V.C	IT20254617 Balasooriya L.D	2022_Feb_IT20186417_	Panel 02	Assign a panel
IT20124867	IT20124867 Punsala S.N.Y	IT20271523 Prabhash Y.M.A	IT20145144 Dewpura S.M.P	IT2012781 Wijesinghe A.M	2022_Feb_IT20124867_	Panel 03	Assign a panel
IT20133672	IT20133672 Adhihetty C.D.	IT20133899 Jayawardana C.D.	IT20133818 Wijenayake A.D.	IT20133122 Perera M.P.M.	2022_Feb_IT20133672_	Panel 04	Assign a panel



All Panels

Panel No.	Member O1	Member O2	Member O3	Action
Panel 01	Gunasekara A.M.P.P	Alagiyawanna A.M.T.P	Ranasinghe R.A.M	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>
Panel 02	Dananjaya K.A.K.D	Iresha P.M.R	Perera N.A.P	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>
Panel 03	C.D. Adhihetty	A.S. Wijenayake	A.D. Samaranayake	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>
Panel 04	C.D. Perera	C.D. Dissanayake	C.D. Jayanayake	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>

Create a New panel

Create New Panel

Panel Number
--Select the Panel Number--

Panel Member 01
Name

Panel Member 02
Name

Panel Member 03
Name

Create Panel

Panel Number
Panel 01

Panel Member 01
Gunasekara A.M.P.P

Panel Member 02
Alagiyawanna A.M.T.P

Panel Member 03
Ranasinghe R.A.M

Edit Panel Members

Update

Leader's ID	Member 01	Member 02	Member 03	Member 04	Group ID	Panel No.
IT20147856	IT20147856 Fernando P.M	IT20157823 Kahandawala A.T	IT20254812 Gunarathne P.P.T	IT20147856 Perera Y.M.N	2022_Feb_IT20147856_	Panel 01
IT20186417	IT20186417 Arachchi A.H.M	IT20127836 Bentharla S.M.P	IT20107482 Gamage W.V.C	IT20254617 Balasooriya L.D	2022_Feb_IT20186417_	Panel 02
IT20124867	IT20124867 Punsalu S.N.Y	IT20271523 Prabhash Y.M.A	IT20145144 Dewpura S.M.P	IT2012781 Wijesinghe A.M	2022_Feb_IT20124867_	Panel 03
IT20133672	IT20133672 Adhiketty C.D.	IT20133899 Jayawardana C.D.	IT20133818 Wijenayake A.D.	IT20133122 Perera M.P.M.	2022_Feb_IT20133672_	Panel 04

All Research groups

Search By Group leader's ID



iii. Topic registration management – IT20135966 Gunasekara A.M.P.P.

#	Group ID	Research Area	Topic	Description	Status	Comments	Supervisor's Name	Action
1	2022_Feb_IT20135966	Artificial Intelligence	Objective Bayesian Nets for Integrating Consistent Datasets	This paper addresses a data integration problem: given several mutually consistent datasets each of which measures a subset of the variables	Approved	nice topic	Pramoda	<button>Reply</button>
2	2022_Feb_IT20147856_	Cloud computing	Cloud Cryptography	Data in the cloud is needed to be protected and secured from foreign attacks and breaches. cryptography in the cloud is a widely used technique	Pending	Well done	Mr. Nihal Thissa	<button>Reply</button>
3	2022_Feb_IT20186417_	Green computing	Green on-demand computing models	Green Cloud Computing is an expansive subject that makes virtualized server farms and workers save energy.	Pending	Good topic	Ms. Nalaka Godahewa	<button>Reply</button>
4	2022_Feb_IT20124867	Machine learning	Supervised Machine Learning	Modeling has become the most commonly used method in fisheries science, with numerous types of models and approaches available today.	Pending	More complex	Ms. Palitha	<button>Reply</button>
5	2022_Feb_IT20133672	Internet of Things	Internet of Things (IoT) Cybersecurity	As an emerging technology, the Internet of Things (IoT) revolutionized the global network comprising of people, smart devices.	Pending	Good	Mr. Tharindu	<button>Reply</button>

Group ID
2022_Feb_IT20135966

Research Area
Artificial Intelligence

Sub Topic
Objective Bayesian Nets for Integrating Consistent D

Brief Description
This paper addresses a data integration problem: giv

Status
Approved

Supervisor's Name
Pramoda

Co-Supervisor's Name
Senuli

Assign





Topic Details

Group ID
2022_Feb_IT20186417_

Research Area
Green computing

Sub Topic
Green on-demand computing models

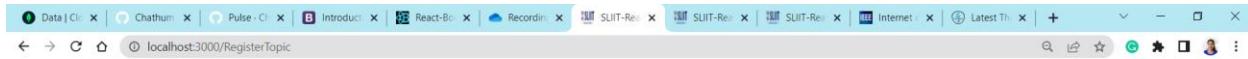
Brief Description
Green Cloud Computing is an expansive subject that

Status
Approved Rejected

Comments
Good topic

Supervisor's Name
Ms. Nalaka Godahewa

Publish



Group ID

Research Area
--Select Your Research Area--

Sub Topic

Brief Description

Post to forum





Topic List

Search by Research Area

#	Group ID	Research Area	Topic	Description	Status	Comments	Supervisor's Name	Co-Supervisor's Name	Action
1	2022_Feb_IT20135966	Artificial Intelligence	Objective Bayesian Nets for Integrating Consistent Datasets	This paper addresses a data integration problem: given several mutually consistent datasets each of which measures a subset of the variables	Approved	nice topic	Pramoda	Senuli	<button>Assign</button>
2	2022_Feb_IT20147856_	Cloud computing	Cloud Cryptography	Data in the cloud is needed to be protected and secured from foreign attacks and breaches. cryptography in the cloud is a widely used technique	Pending	Well done	Mr. Nihal Thissa	Not yet assigned	<button>Assign</button>
3	2022_Feb_IT20186417_	Green computing	Green on-demand computing models	Green Cloud Computing is an expansive subject that makes virtualized server farms and workers save energy.	Pending	Good topic	Ms. Nalaka Godahewa	Not yet assigned	<button>Assign</button>
4	2022_Feb_IT20124867	Machine learning	Supervised Machine Learning	Modeling has become the most commonly used method in fisheries science, with numerous types of models and approaches available today.	Pending	More complex	Ms.Palitha	Not yet assigned	<button>Assign</button>
5	2022_Feb_IT20133672	Internet of Things	Internet of Things (IoT) Cybersecurity	As an emerging technology, the Internet of Things (IoT) revolutionized the global network comprising of people, smart devices,	Pending	Good	Mr. Tharindu	Not yet assigned	<button>Assign</button>



Topic List

Search by Research Area

#	Group ID	Research Area	Topic	Description	Status	Comments	Supervisor's Name	Co-Supervisor's Name	Final feedback from Panel
1	2022_Feb_IT20135966	Artificial Intelligence	Objective Bayesian Nets for Integrating Consistent Datasets	This paper addresses a data integration problem: given several mutually consistent datasets each of which measures a subset of the variables	Approved	nice topic	Pramoda	Senuli	Accepted
2	2022_Feb_IT20147856_	Cloud computing	Cloud Cryptography	Data in the cloud is needed to be protected and secured from foreign attacks and breaches. cryptography in the cloud is a widely used technique	Pending	Well done	Mr. Nihal Thissa	Not yet assigned	Not yet finalized
3	2022_Feb_IT20186417_	Green computing	Green on-demand computing models	Green Cloud Computing is an expansive subject that makes virtualized server farms and workers save energy.	Pending	Good topic	Ms. Nalaka Godahewa	Not yet assigned	Not yet finalized
4	2022_Feb_IT20124867	Machine learning	Supervised Machine Learning	Modeling has become the most commonly used method in fisheries science, with numerous types of models and approaches available today.	Pending	More complex	Ms.Palitha	Not yet assigned	Not yet finalized
5	2022_Feb_IT20133672	Internet of Things	Internet of Things (IoT) Cybersecurity	As an emerging technology, the Internet of Things (IoT) revolutionized the global network comprising of people, smart devices.	Pending	Good	Mr. Tharindu	Not yet assigned	Not yet finalized

[Register Topic](#)

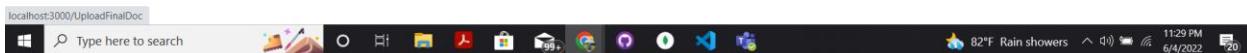
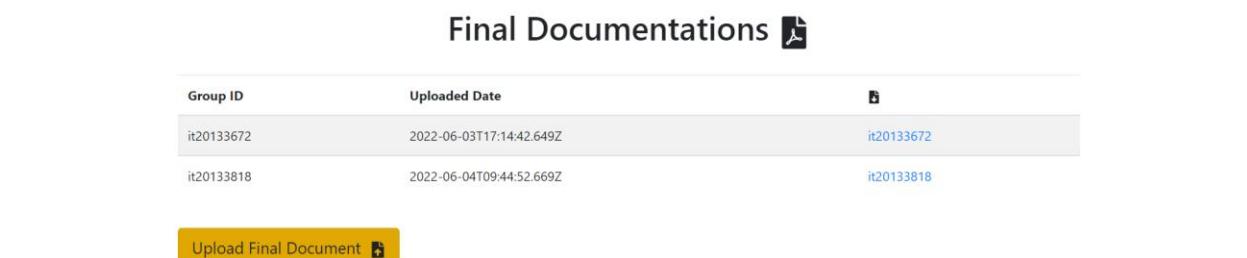
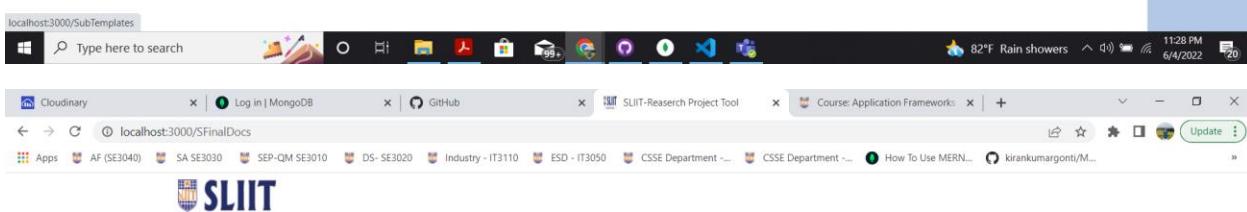
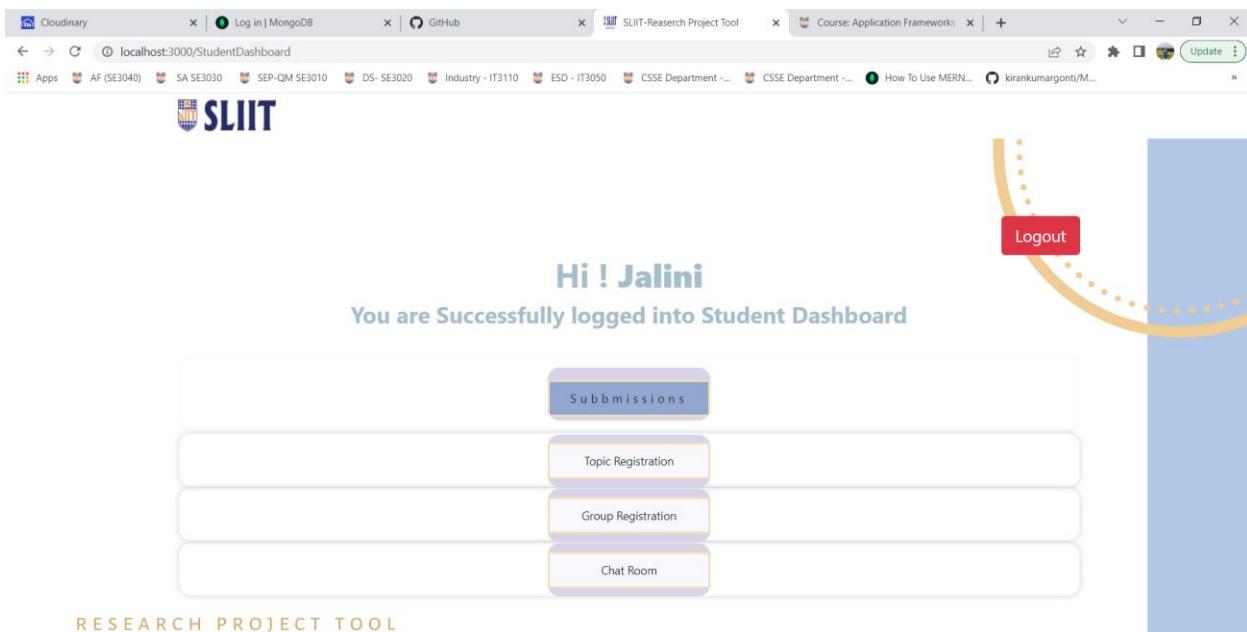


The screenshot shows a web browser window with multiple tabs open. The active tab displays a table titled "Topic Registration Documentations" with two rows of data. The columns are "Group ID", "Uploaded Date", and "File". The first row has Group ID "it20133818", Uploaded Date "2022-06-04T17:57:52.436Z", and File "it20133818". The second row has Group ID "it20133672", Uploaded Date "2022-06-04T17:57:54.798Z", and File "it20133672". Below the table is a yellow button labeled "Upload Topic Registration Document" with a file icon.

The screenshot shows a web browser window with multiple tabs open. The active tab displays a form titled "Upload Topic Registration Document". It includes a "Name" input field, a "Choose Files" button, and a message "No file chosen". Below these is an "Upload" button with a file icon.



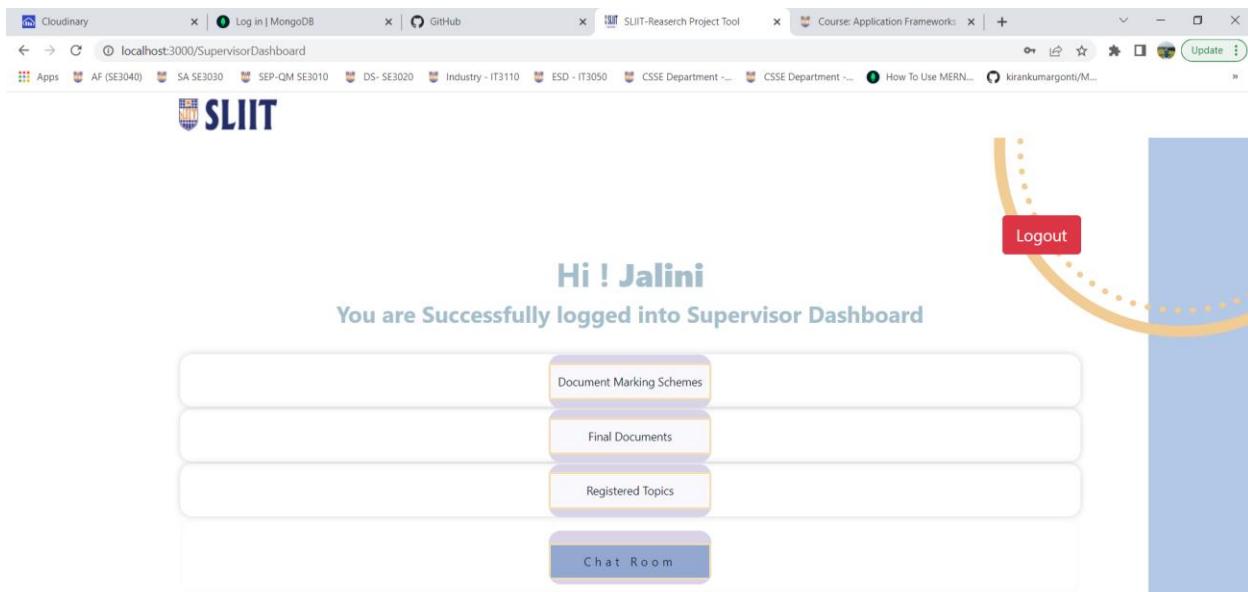
iv. Final document submission management & Chat management – IT20133818
 Perera M.P.M.



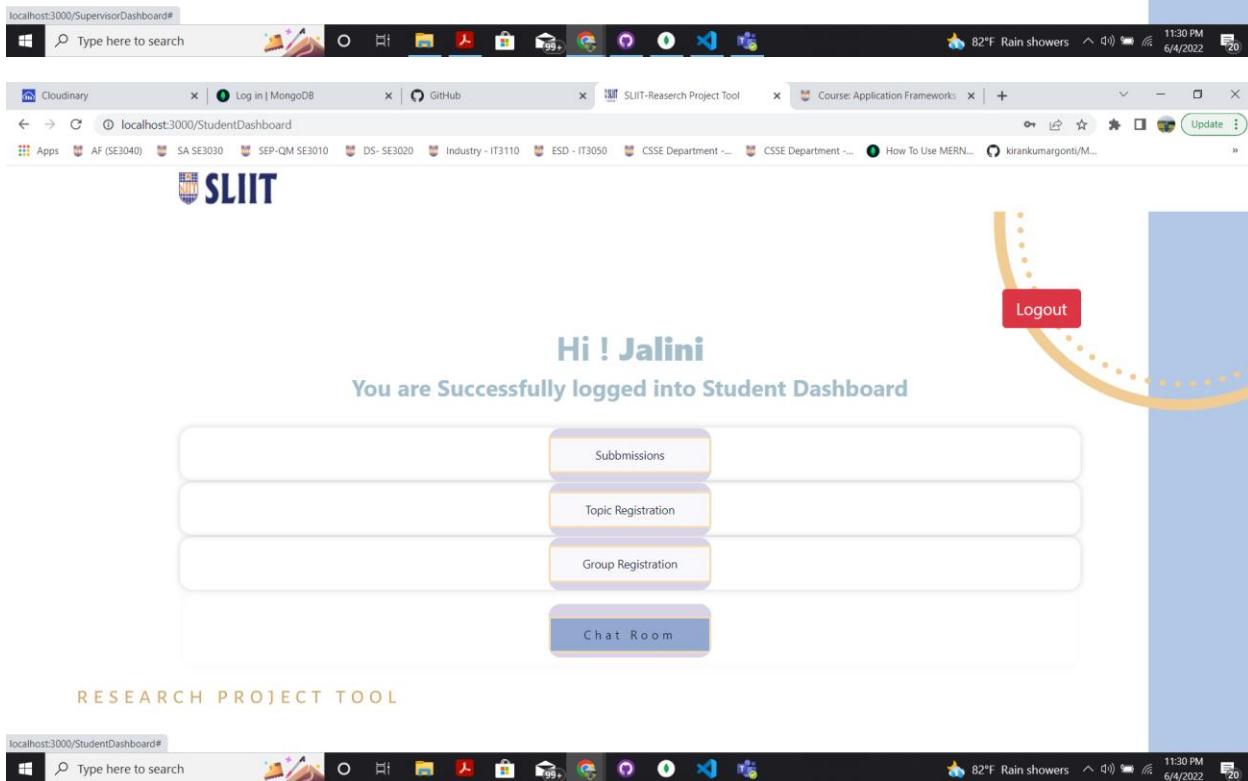
The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Upload Final Document". The page content includes a form with a "Group ID" input field, a "Choose Files" button with the message "No file chosen", and an "Upload" button.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "CoSupervisor Dashboard". The page content displays a greeting "Hi!Jalini" and a message "You are Successfully logged into Co-Supervisor Dashboard". It features two buttons: "Assign To A Group" and "Chat Room". A red "Logout" button is located in the top right corner of the dashboard area.

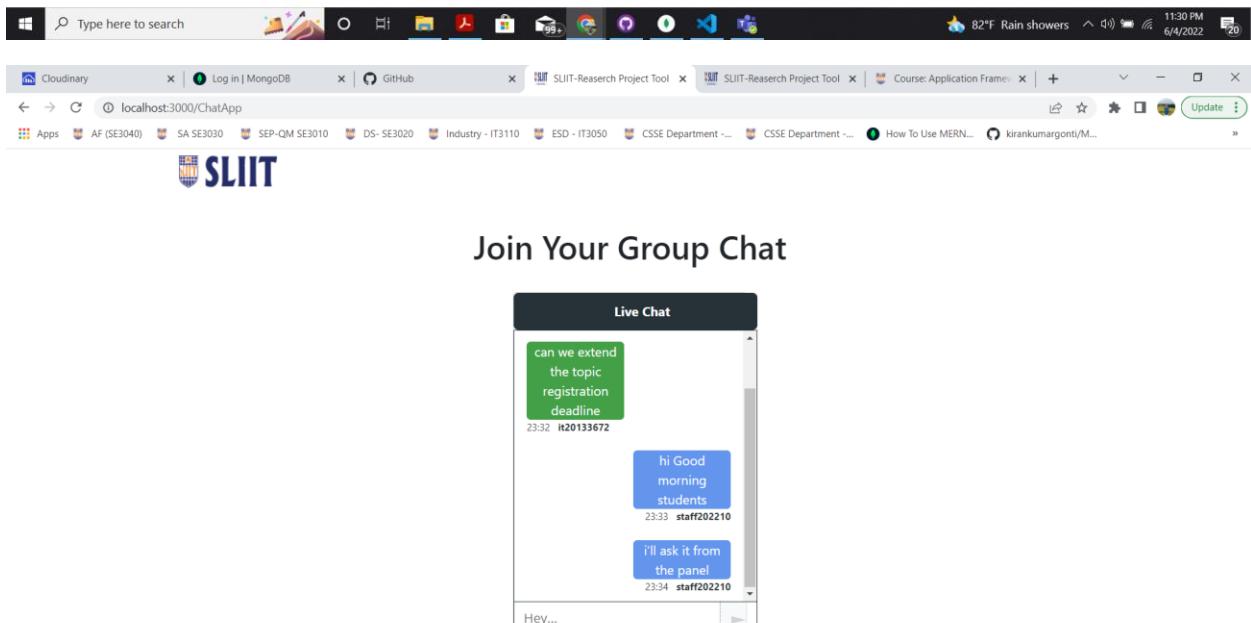
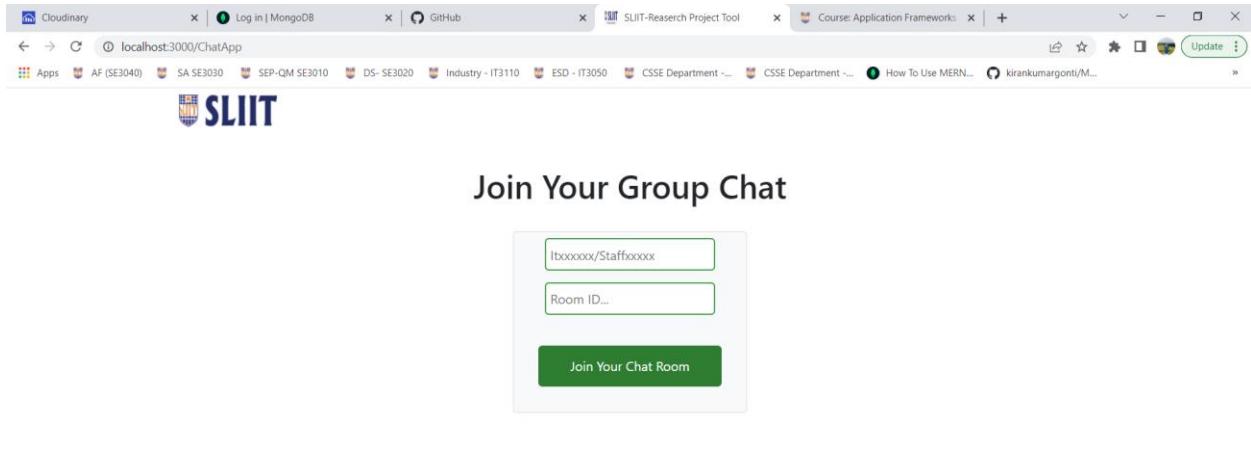
The screenshot shows a web browser window with multiple tabs open. The active tab is titled "RESEARCH PROJECT TOOL". The page content is minimal, showing the title "RESEARCH PROJECT TOOL" at the top. The browser's taskbar at the bottom shows the URL "localhost:3000/ChatApp".



RESEARCH PROJECT TOOL



RESEARCH PROJECT TOOL



3. Screenshots/ Code Snippets of the Backend

- i. User management and document management – IT20133672 C.D. Adhiketty

Backend

1) User management & Document Management

```
2) const mongoose = require('mongoose');
3)
4) const UserSchema = new mongoose.Schema({
5)   name: {
6)     type: String,
7)     required: true
8)   },
9)   email: {
10)    type: String,
11)    required: true
12)  },
13)   password: {
14)     type: String,
15)     required: true
16)   },
17)   userCategory: {
18)     type: String,
19)     required: true
20)   },
21)   regNo: {
22)     type: String,
23)     required: true
24)   },
25)   nic: {
26)     type: String,
27)     required: true
28)   },
29)   pNo: {
30)     type: String,
31)     required: true
32)   },
33)   date: {
34)     type: Date,
35)     default: Date.now
36) }
```

```
37));
38)
39) const User = mongoose.model('User', UserSchema);
40)
41) module.exports = User;
```

```
const express = require('express');
const Router = express.Router();
const bcrypt = require('bcryptjs');
const jwt = require('jsonwebtoken');
const config = require('config');

//Load Input Validation
const validateRegisterInput = require('../validation/register');
const validateLoginInput = require('../validation/login');

// Load User model
const User = require('../models/User');

//Post Router api/users/register
Router.post('/register', (req, res) => {
    //Form Validation
    //Destructuring Values
    const {
        errors,
        isValid
    } = validateRegisterInput(req.body);

    //Check Validation
    if (!isValid) {
        return res.status(400).json(errors);
    }

    User.findOne({
        email: req.body.email
    })
    .then(user => {
        if (user) {
            return res.status(400).json({
                email: "Email already exists"
            });
        }
    })
    .catch(err => console.log(err));
});
```

```
    } else {
        const newUser = new User({
            name: req.body.name,
            email: req.body.email,
            userCategory: req.body.userCategory,
            regNo: req.body.regNo,
            nic: req.body.nic,
            pNo: req.body.pNo,
            password: req.body.password
        });

        //Hash password before saving in database
        bcrypt.genSalt(10, (err, salt) => {
            bcrypt.hash(newUser.password, salt, (err, hash) => {
                if (err) throw err;
                newUser.password = hash;
                newUser.save()
                    .then(user => res.json(user))
                    // res.redirect('/users/login')
                    .catch(err => console.log(err));
            });
        });
    });
};

//Post Router api/users/login

Router.post('/login', (req, res) => {
    //Login Validation
    const {
        errors,
        isValid
    } = validateLoginInput(req.body);

    //Check Validation
    if (!isValid) {
        return res.status(400).json(errors);
    }

    const email = req.body.email;
    const password = req.body.password;

    //Find User By Email
```

```

User.findOne({
  email:email
}).then(user => {
  //Check if Your Exists
  if (!user) {
    return res.status(404).json({
      emailNotFound: "Email is not registered"
    });
  }

  //Match Password
  bcrypt.compare(password, user.password)
    .then(isMatch => {
      if (isMatch) {
        //User Matched
        //Create JWT Payload
        const payload = {
          id: user.id,
          name: user.name
        };

        //Sign Token
        jwt.sign(payload, config.get('secretOrKey'), {
          expiresIn: 63113852 //2 years in seconds
        }, (err, token) => {
          res.json({
            success: true,
            token: "Bearer" + token
          });
        });
      } else {
        return res.status(400).json({
          passwordIncorrect: "Password incorrect"
        });
      }
    });
});

module.exports = Router;

```

```

const express = require('express');
const User = require('../models/User');

```

```
const router = express.Router();

//get User Profiles

router.get('/users',(req,res) =>{
    User.find().exec((err,posts) =>{
        if(err){
            return res.status(400).json({
                error:err
            });
        }
        return res.status(200).json({
            success:true,
            existingPosts:posts
        });
    });
});

//get a specific User Profile

router.get("/User/:id",(req,res) =>{
    let postId = req.params.id;

    User.findById(postId,(err,post) =>{
        if(err){
            return res.status(400).json({success:false, err})
        }

        return res.status(200).json({
            success:true,
            post
        });
    });
});

//update a User Profile

router.put('/User/update/:id',(req,res)=>{
    User.findByIdAndUpdate(
```

```

    req.params.id,
    {
      $set:req.body
    },
    (err,post)=>{
      if(err){
        return res.status(400).json({error:err});
      }

      return res.status(200).json({
        success:"Updated Successfully"
      });
    }
  );
});

//Delete a User Profile

router.delete('/User/delete/:id',(req,res) =>{
  User.findByIdAndRemove(req.params.id).exec((err,deletePost) =>{

    if(err) return res.status(400).json({
      message:"Delete Unsuccessful",err
    });

    return res.json({
      message:"Delete Successful",deletePost
    });
  });
});

module.exports = router;

```

```

const mongoose = require('mongoose');

const marksPanelSchema = new mongoose.Schema({
  name:{
    type: String,
  },
  description:{
    type:String,
  },
  pdf:{
    type: String,
  }
})

```

```

    },
    cloudinary_id_pdf: {
      type: String,
    }
  });

module.exports = mongoose.model("MarksPannel", marksPannelSchema);

```

```

const router = require("express").Router();
const cloudinary = require("../utils/cloudinary");
const multer = require('multer')
const uuid=require('uuid').v4;
const path = require('path');
const MarksPannel = require("../models/marksPannel");

const files=[];
const fileInArray=[]

const storage=multer.diskStorage({
  destination:(req,file,cb)=>{
    cb(null,"uploads")
},
  filename:(req,file,cb)=>{
    let filePath=[];
    console.log("MULTER ENTRY ",file.originalname)
    console.log("files",req.files)

    const ext = path.extname(file.originalname);
    const id = uuid();
    filePath = `${id}${ext}`;
    fileInArray.push([(filePath)])
    console.log("IN ARRAY ",filePath)
    files.push(fileInArray)
    console.log("PUSHED MAIN ARRAY", fileInArray)
    cb(null,filePath)
    console.log("current length",files.length)
  }
})

const upload=multer({
  fileFilter: (req, file, cb) => {

```

```
        if (file.mimetype == "image/png" || file.mimetype == "image/jpg" ||
file.mimetype == "image/jpeg" || file.mimetype == "video/mp4" || file.mimetype ==
"application/pdf") {
            cb(null, true);
        } else {
            cb(null, false);
            return cb(new Error('Only .png, .jpg, .jpeg .mp4 and .pdf format
allowed!'));
        }
    },
storage:storage,
})

router.post("/", upload.array('uploaded_Image', 10), async (req, res) => {
    try {

        console.log(req.files.length)
        console.log("Files",fileInArray)
        let pdff;

        for(let i=0;i<fileInArray.length;i++){
            let fileext = fileInArray[i][0].split('.')[1];
            console.log(path.resolve(__dirname, "../uploads"))
            if(fileext=="pdf")
                pdff = await cloudinary.uploader.upload(` ${path.resolve(__dirname,
"../uploads")}/${fileInArray[i][0]}`,{ pages: true });
        }

        let marksPannel = new MarksPannel({
            name: req.body.name,
            description:req.body.description,
            pdf : pdff.secure_url,
            cloudinary_id_pdf: pdff.public_id,
        });

        await marksPannel.save();
        res.json(marksPannel);
    } catch (err) {
        console.log(err);
    }
});

router.get("/", async (req, res) => {
    try {
```

```
let marksPannel = await MarksPannel.find();
res.json(marksPannel);
} catch (err) {
  console.log(err);
}
});

router.delete("/:id", async(req,res)=>{

  try {
    const id=req.params.id;
    let marksPannel = await MarksPannel.findByIdAndDelete(id);
    res.json(marksPannel);
  } catch (err) {
    console.log(err)
  }
}

module.exports = router;
```

```
const mongoose = require('mongoose');

const marksSupervisorSchema = new mongoose.Schema({
  name:{
    type: String,
  },
  description:{
    type:String,
  },
  pdf:{
    type: String,
  },
  cloudinary_id_pdf: {
```

```

        type: String,
    }
});

module.exports = mongoose.model("MarksSupervisor", marksSupervisorSchema);

```

```

const router = require("express").Router();
const cloudinary = require("../utils/cloudinary");
const multer = require('multer')
const uuid=require('uuid').v4;
const path = require('path');
const MarksSupervisor = require("../models/marksSupervisor");

const files=[];
const fileInArray=[]

const storage=multer.diskStorage({
    destination:(req,file,cb)=>{
        cb(null,"uploads")
},
    filename:(req,file,cb)=>{
        let filePath=[];
        console.log("MULTER ENTRY ",file.originalname)
        console.log("files",req.files)

        const ext = path.extname(file.originalname);
        const id = uuid();
        filePath = `${id}${ext}`;
        fileInArray.push([(filePath)])
        console.log("IN ARRAY ",filePath)
        files.push(fileInArray)
        console.log("PUSHED MAIN ARRAY", fileInArray)
        cb(null,filePath)
        console.log("current length",files.length)
    }
})

const upload=multer({
    fileFilter: (req, file, cb) => {
        if (file.mimetype == "image/png" || file.mimetype == "image/jpg" ||
file.mimetype == "image/jpeg" || file.mimetype == "video/mp4" || file.mimetype ==
"application/pdf") {

```

```
        cb(null, true);
    } else {
        cb(null, false);
        return cb(new Error('Only .png, .jpg, .jpeg .mp4 and .pdf format
allowed!'));
    }
},
storage:storage,
})

router.post("/", upload.array('uploaded_Image', 10), async (req, res) => {
    try {

        console.log(req.files.length)
        console.log("Files",fileInArray)
        let pdff;

        for(let i=0;i<fileInArray.length;i++){
            let fileext = fileInArray[i][0].split('.')[1];
            console.log(path.resolve(__dirname, "../uploads"))
            if(fileext=="pdf")
                pdff = await cloudinary.uploader.upload(` ${path.resolve(__dirname,
"../uploads")}/${fileInArray[i][0]}`,{ pages: true });
        }

        let marksSupervisor = new MarksSupervisor({
            name: req.body.name,
            description:req.body.description,
            pdf : pdff.secure_url,
            cloudinary_id_pdf: pdff.public_id,
        });

        await marksSupervisor.save();
        res.json(marksSupervisor);
    } catch (err) {
        console.log(err);
    }
});

router.get("/", async (req, res) => {
    try {
        let marksSupervisor = await MarksSupervisor.find();
        res.json(marksSupervisor);
    } catch (err) {
```

```

        console.log(err);
    }
});

router.delete("/:id", async(req,res)=>{

    try {
        const id=req.params.id;
        let marksSupervisor = await MarksSupervisor.findByIdAndDelete(id);
        res.json(marksSupervisor);
    } catch (err) {
        console.log(err)
    }

})

module.exports = router;

```

```

const mongoose = require('mongoose');

const templatesSchema = new mongoose.Schema({
    name:{
        type: String,
    },
    description:{
        type:String,
    },
    pdf:{
        type: String,
    },
    cloudinary_id_pdf: {
        type: String,
    }
});

module.exports = mongoose.model("Templates", templatesSchema);

```

```

const router = require("express").Router();
const cloudinary = require("../utils/cloudinary");
const multer = require('multer')
const uuid=require('uuid').v4;
const path = require('path');

```

```

const Templates = require("../models/templates");

const files=[];
const fileInArray=[]

const storage=multer.diskStorage({
  destination:(req,file,cb)=>{
    cb(null,"uploads")
  },
  filename:(req,file,cb)=>{
    let filePath=[];
    console.log("MULTER ENTRY ",file.originalname)
    console.log("files",req.files)

    const ext = path.extname(file.originalname);
    const id = uuid();
    filePath = `${id}${ext}`;
    fileInArray.push([(filePath)])
    console.log("IN ARRAY ",filePath)
    files.push(fileInArray)
    console.log("PUSHED MAIN ARRAY", fileInArray)
    cb(null,filePath)
    console.log("current length",files.length)
  }
})

const upload=multer({
  fileFilter: (req, file, cb) => {
    if (file.mimetype == "image/png" || file.mimetype == "image/jpg" ||
    file.mimetype == "image/jpeg" || file.mimetype == "video/mp4" || file.mimetype ==
    "application/pdf") {
      cb(null, true);
    } else {
      cb(null, false);
      return cb(new Error('Only .png, .jpg, .jpeg .mp4 and .pdf format
allowed!'));
    }
  },
  storage:storage,
})

router.post("/", upload.array('uploaded_Image', 10), async (req, res) => {

```

```
try {

    console.log(req.files.length)
    console.log("Files",fileInArray)
    let pdff;

    for(let i=0;i<fileInArray.length;i++){
        let fileext = fileInArray[i][0].split('.')[1];
        console.log(path.resolve(__dirname, "../uploads"))
        if(fileext=="pdf")
            pdff = await cloudinary.uploader.upload(` ${path.resolve(__dirname,
"../uploads")}/${fileInArray[i][0]}`,{ pages: true });
    }

    let templates = new Templates({
        name: req.body.name,
        description:req.body.description,
        pdf : pdff.secure_url,
        cloudinary_id_pdf: pdff.public_id,
    });

    await templates.save();
    res.json(templates);
} catch (err) {
    console.log(err);
}
});

router.get("/", async (req, res) => {
    try {
        let templates = await Templates.find();
        res.json(templates);
    } catch (err) {
        console.log(err);
    }
});

router.delete("/:id", async(req,res)=>{

    try {
        const id=req.params.id;
        let templates = await Templates.findByIdAndDelete(id);
        res.json(templates);
    } catch (err) {
        console.log(err)
    }
});
```

```
    }

})

module.exports = router;
```

- ii. Student group management and panel management – IT20133122 R.A.E. Wijenayake

Backend

2) Student Group management and panel Management

```
const mongoose = require("mongoose");

const groupSchema = new mongoose.Schema({
  l_id: {
    type: String,
  },
  g_id: {
    type: String,
  },

  member01: {
    type: String,
  },
  member02: {
    type: String,
  },
  member03: {
    type: String,
  },
  member04: {
    type: String,
  },

  panel: {
```

```
        type: String,
    },
    panelNo: {
        type: String,
    },
    pMember01: {
        type: String,
    },
    pMember02: {
        type: String,
    },
    pMember03: {
        type: String,
    },
});
module.exports = mongoose.model("Groups", groupSchema);
```

```
const express = require("express");
const Groups = require("../models/groups");

const router = express.Router();

//save groups

router.post("/group/save", (req, res) => {
    let newGroup = new Groups(req.body);

    newGroup.save((err) => {
        if (err) {
            return res.status(400).json({
                error: err,
            });
        }
        return res.status(200).json({
            success: "Groups saved successfully",
        });
    });
});

//get groups

router.get("/groups", (req, res) => {
```

```
Groups.find().exec((err, groups) => {
  if (err) {
    return res.status(400).json({
      error: err,
    });
  }
  return res.status(200).json({
    success: true,
    existingGroups: groups,
  });
});

//get a specific group

router.get("/group/:id", (req, res) => {
  let groupId = req.params.id;

  Groups.findById(groupId, (err, group) => {
    if (err) {
      return res.status(400).json({ success: false, err });
    }

    return res.status(200).json({
      success: true,
      group,
    });
  });
});

//update group

router.put("/group/update/:id", (req, res) => {
  Groups.findByIdAndUpdate(
    req.params.id,
    {
      $set: req.body,
    },
    (err, group) => {
      if (err) {
        return res.status(400).json({ error: err });
      }

      return res.status(200).json({
        success: "Updated Successfully",
      });
    }
  );
});
```

```
        });
    }
);
});

//delete group

router.delete("/group/delete/:id", (req, res) => {
  Groups.findByIdAndRemove(req.params.id).exec((err, deleteGroup) => {
    if (err)
      return res.status(400).json({
        message: "Delete Unsuccessful",
        err,
      });

    return res.json({
      message: "delete Successfull",
      deleteGroup,
    });
  });
);

module.exports = router;
```

```
const mongoose = require("mongoose");

const panelSchema = new mongoose.Schema({
  panelNo: {
    type: String,
  },
  pMember01: {
    type: String,
  },
  pMember02: {
    type: String,
  },
  pMember03: {
    type: String,
  },
});

module.exports = mongoose.model("Panels", panelSchema);
```

```
const express = require("express");
const Panels = require("../models/panels");

const router = express.Router();

router.post("/panel/save", (req, res) => {
  let newPanel = new Panels(req.body);

  newPanel.save((err) => {
    if (err) {
      return res.status(400).json({
        error: err,
      });
    }
    return res.status(200).json({
      success: "Panel Created successfully",
    });
  });
});

router.get("/panels", (req, res) => {
  Panels.find().exec((err, panels) => {
    if (err) {
      return res.status(400).json({
        error: err,
      });
    }
    return res.status(200).json({
      success: true,
      existingPanels: panels,
    });
  });
});

router.get("/panel/:id", (req, res) => {
  let panelId = req.params.id;

  Panels.findById(panelId, (err, panel) => {
    if (err) {
      return res.status(400).json({ success: false, err });
    }

    return res.status(200).json({

```

```
        success: true,
        panel,
    });
});
});

router.put("/panel/update/:id", (req, res) => {
    Panels.findByIdAndUpdate(
        req.params.id,
        {
            $set: req.body,
        },
        (err, panel) => {
            if (err) {
                return res.status(400).json({ error: err });
            }

            return res.status(200).json({
                success: "Updated Successfully",
            });
        }
    );
});

router.delete("/panel/delete/:id", (req, res) => {
    Panels.findByIdAndRemove(req.params.id).exec((err, deleteTopic) => {
        if (err)
            return res.status(400).json({
                message: "Delete Unsuccessful",
                err,
            });

        return res.json({
            message: "Delete Successful",
            deleteTopic,
        });
    });
});

module.exports = router;
```

iii. Topic registration management – IT20135966 Gunasekara A.M.P.P.

Backend

3) Topic registration Management

```
const mongoose = require("mongoose");

const topicSchema = new mongoose.Schema({
  gid: {
    type: String,
  },

  area: {
    type: String,
  },

  name: {
    type: String,
  },
  description: {
    type: String,
  },

  status: {
    type: String,
    default: "Pending",
  },

  comment: {
    type: String,
  },

  sName: {
    type: String,
    default: "Pending",
  },

  csName: {
    type: String,
  }
});
```

```
        default: "Not yet assigned ",
    },
    feedback: {
        type: String,
        default: "Not yet finalized "
    },
});

module.exports = mongoose.model("Topics", topicSchema);
```

```
const express = require("express");
const Topics = require("../models/topics");

const router = express.Router();

//save topics

router.post("/topic/save", (req, res) => {
    let newTopic = new Topics(req.body);

    newTopic.save((err) => {
        if (err) {
            return res.status(400).json({
                error: err,
            });
        }
        return res.status(200).json({
            success: "Topics saved successfully",
        });
    });
});

//get topics

router.get("/topics", (req, res) => {
    Topics.find().exec((err, topics) => {
        if (err) {
            return res.status(400).json({
                error: err,
            });
        }
    });
});
```

```
        }
      return res.status(200).json({
        success: true,
        existingTopics: topics,
      });
    });
  });

//get a specific topic

router.get("/topic/:id", (req, res) => {
  let topicId = req.params.id;

  Topics.findById(topicId, (err, topic) => {
    if (err) {
      return res.status(400).json({ success: false, err });
    }

    return res.status(200).json({
      success: true,
      topic,
    });
  });
});

//update topics

router.put("/topic/update/:id", (req, res) => {
  Topics.findByIdAndUpdate(
    req.params.id,
    {
      $set: req.body,
    },
    (err, topic) => {
      if (err) {
        return res.status(400).json({ error: err });
      }

      return res.status(200).json({
        success: "Updated Successfully",
      });
    });
});
```

```
//delete topic

router.delete("/topic/delete/:id", (req, res) => {
  Topics.findByIdAndRemove(req.params.id).exec((err, deleteTopic) => {
    if (err)
      return res.status(400).json({
        message: "Delete Unsuccessful",
        err,
      });

    return res.json({
      message: "delete Successfull",
      deleteTopic,
    });
  });
});

module.exports = router;
```

```
const mongoose = require('mongoose');

const topicRegDocSchema = new mongoose.Schema({
  name: {
    type: String,
  },
  pdf: {
    type: String,
  },
  cloudinary_id_pdf: {
    type: String,
  },
  date: {
    type: Date,
    default: Date.now
  }
});

module.exports = mongoose.model("TopicRegDoc", topicRegDocSchema);
```

```
const router = require("express").Router();
const cloudinary = require("../utils/cloudinary");
const multer = require('multer')
```

```
const uuid=require('uuid').v4;
const path = require('path');
const TopicRegDoc = require("../models/topicRegDoc");

const files=[];
const fileInArray=[]

const storage=multer.diskStorage({
  destination:(req,file,cb)=>{
    cb(null,"uploads")
},
  filename:(req,file,cb)=>{
    let filePath=[];
    console.log("MULTER ENTRY ",file.originalname)
    console.log("files",req.files)

    const ext = path.extname(file.originalname);
    const id = uuid();
    filePath = `${id}${ext}`;
    fileInArray.push([(filePath)])
    console.log("IN ARRAY ",filePath)
    files.push(fileInArray)
    console.log("PUSHED MAIN ARRAY", fileInArray)
    cb(null,filePath)
    console.log("current length",files.length)
  }
})

const upload=multer({

  fileFilter: (req, file, cb) => {
    if (file.mimetype == "image/png" || file.mimetype == "image/jpg" ||
    file.mimetype == "image/jpeg" || file.mimetype == "video/mp4" || file.mimetype ==
    "application/pdf") {
      cb(null, true);
    } else {
      cb(null, false);
      return cb(new Error('Only .png, .jpg, .jpeg .mp4 and .pdf format
allowed!'));
    }
  },
  storage:storage,
})
```

```
router.post("/", upload.array('uploaded_Image', 10), async (req, res) => {
  try {

    console.log(req.files.length)
    console.log("Files",fileInArray)
    let pdff;

    for(let i=0;i<fileInArray.length;i++){
      let fileext = fileInArray[i][0].split('.')[1];
      console.log(path.resolve(__dirname, "../uploads"))
      if(fileext=="pdf")
        pdff = await cloudinary.uploader.upload(` ${path.resolve(__dirname,
      "../uploads")}/${fileInArray[i][0]}`,{ pages: true });
    }

    let topicRegDoc = new TopicRegDoc({
      name: req.body.name,
      date:req.body.date,
      pdf : pdff.secure_url,
      cloudinary_id_pdf: pdff.public_id,
    });

    await topicRegDoc.save();
    res.json(topicRegDoc);
  } catch (err) {
    console.log(err);
  }
});

router.get("/", async (req, res) => {
  try {
    let topicRegDoc = await TopicRegDoc.find();
    res.json(topicRegDoc);
  } catch (err) {
    console.log(err);
  }
});

router.delete("/:id", async(req,res)=>{

  try {
    const id=req.params.id;
    let topicRegDoc = await TopicRegDoc.findByIdAndDelete(id);
    res.json(topicRegDoc);
  }
```

```

    } catch (err) {
      console.log(err)
    }
  })

module.exports = router;

```

- iv. Final document submission management & Chat management – IT20133818
Perera M.P.M.

Backend

4) final doc and chat management

```

const mongoose = require('mongoose');

const finalDocSchema = new mongoose.Schema({
  name: {
    type: String,
  },
  pdf: {
    type: String,
  },
  cloudinary_id_pdf: {
    type: String,
  },
  date: {
    type: Date,
    default: Date.now
  }
});

module.exports = mongoose.model("FinalDoc", finalDocSchema);

```

```

const router = require("express").Router();
const cloudinary = require("../utils/cloudinary");

```

```
const multer = require('multer')
const uuid=require('uuid').v4;
const path = require('path');
const FinalDoc = require("../models/finalDoc");

const files=[];
const fileInArray=[]

const storage=multer.diskStorage({
  destination:(req,file,cb)=>{
    cb(null,"uploads")
  },
  filename:(req,file,cb)=>{
    let filePath=[];
    console.log("MULTER ENTRY ",file.originalname)
    console.log("files",req.files)

    const ext = path.extname(file.originalname);
    const id = uuid();
    filePath = `${id}${ext}`;
    fileInArray.push([(filePath)])
    console.log("IN ARRAY ",filePath)
    files.push(fileInArray)
    console.log("PUSHED MAIN ARRAY", fileInArray)
    cb(null,filePath)
    console.log("current length",files.length)
  }
})

const upload=multer({
  fileFilter: (req, file, cb) => {
    if (file.mimetype == "image/png" || file.mimetype == "image/jpg" ||
    file.mimetype == "image/jpeg" || file.mimetype == "video/mp4" || file.mimetype ==
    "application/pdf") {
      cb(null, true);
    } else {
      cb(null, false);
      return cb(new Error('Only .png, .jpg, .jpeg .mp4 and .pdf format
allowed!'));
    }
  },
  storage:storage,
})
```

```
router.post("/", upload.array('uploaded_Image', 10), async (req, res) => {
  try {

    console.log(req.files.length)
    console.log("Files", fileInArray)
    let pdff;

    for(let i=0;i<fileInArray.length;i++){
      let fileext = fileInArray[i][0].split('.')[1];
      console.log(path.resolve(__dirname, "../uploads"))
      if(fileext=="pdf")
        pdff = await cloudinary.uploader.upload(` ${path.resolve(__dirname,
      "../uploads")}/${fileInArray[i][0]}`,{ pages: true });
    }

    let finalDoc = new FinalDoc({
      name: req.body.name,
      date:req.body.date,
      pdf : pdff.secure_url,
      cloudinary_id_pdf: pdff.public_id,
    });

    await finalDoc.save();
    res.json(finalDoc);
  } catch (err) {
    console.log(err);
  }
});

router.get("/", async (req, res) => {
  try {
    let finalDoc = await FinalDoc.find();
    res.json(finalDoc);
  } catch (err) {
    console.log(err);
  }
});

router.delete("/:id", async (req, res) =>{
  try {
    const id=req.params.id;
    let finalDoc = await FinalDoc.findByIdAndDelete(id);
  }
});
```

```
    res.json(finalDoc);
} catch (err) {
  console.log(err)
}

})

module.exports = router;
```

Server.js

```
const express = require("express");

const mongoose = require("mongoose");
const passport = require("passport");
const dotenv = require("dotenv");
const cors = require("cors");
const http = require("http");
const { Server } = require("socket.io");

const server = http.createServer();

const io = new Server(server, {
  cors: {
    origin: "http://localhost:3000",
    methods: ["GET", "POST"],
  },
});

io.on("connection", (socket) => {
  console.log(`User Connected: ${socket.id}`);

  socket.on("join_room", (data) => {
    socket.join(data);
    console.log(`User with ID: ${socket.id} joined room: ${data}`);
  });

  socket.on("send_message", (data) => {
    socket.to(data.room).emit("receive_message", data);
  });

  socket.on("disconnect", () => {
```

```
        console.log("User Disconnected", socket.id);
    });
});
server.listen(3001, () => {
    console.log("SERVER RUNNING");
});

const usersRouter = require("./routes/api/users");
const userProfiles = require("./routes/userProfiles");
const topicRoutes = require("./routes/topics");
const groupRoutes = require("./routes/groups");
const panelRoutes = require("./routes/panels");

dotenv.config();
const config = require('config');
const app = express();
// Body parser middleware
app.use(
    express.urlencoded({
        extended: false
    })
);
app.use(express.json());
app.use(cors());
// DB Config
const db = config.get('mongoURI');
// Connect to MongoDB
mongoose
    .connect(
        db, {
            useNewUrlParser: true,
            useUnifiedTopology: true
        }
    )
    .then(() => console.log("MongoDB successfully connected"))
    .catch(err => console.log(err));
// Passport middleware
app.use(passport.initialize());
// Passport config
require("./config/passport")(passport);
// Routes
app.use("/api/users", usersRouter);
app.use(userProfiles);
app.use(topicRoutes);
```

```

app.use(groupRoutes);
app.use(panelRoutes);
app.use("/marksSupervisor", require("./routes/marksSupervisor"));
app.use("/marksPannel", require("./routes/marksPannel"));
app.use("/templates", require("./routes/templates"));
app.use("/topicRegDoc", require("./routes/topicRegDoc"));
app.use("/finalDoc", require("./routes/finalDoc"));

const port = process.env.PORT || 5000;
app.listen(port, () => console.log(`Server up and running on port ${port} !`));

```

4. Screenshots of the MongoDB Schemas

i. User management and document management – IT20133672 C.D. Adhiketty

The screenshot shows the MongoDB Compass application interface. The left sidebar shows the database structure with 'ResearchProjectManagementToolDatabase' selected, containing collections like 'finaldocs', 'groups', 'markspannels', 'marksupervisors', 'panels', 'templates', 'topicregdocs', 'topics', and 'users'. The main area is titled 'ResearchProjectManagementToolDatabase.users' and shows 7 documents and 1 index. A specific document is selected, displaying its details:

```

_id: ObjectId("629b7bf43655191760257d41")
name: "Charitha Desamanya Adhiketty"
email: "1720133672@srkay.sril.lk"
password: "$2a$10$wRkqg$inhtJhd3k4y.01.aa/xi4z05EMIda5y47hlo3IH1u488a"
userCategory: "Student"
regNo: "1720133672"
nic: "200009962428"
pho: "0773825119"
date: 2022-06-01T00:00:00+00:00
__v: 0

```

```

_id: ObjectId("629b7bf43655191760257d44")
name: "Dhanusha Adhiketty"
email: "staff987615@gmail.sril.lk"
userCategory: "Supervisor"
regNo: "1720133672"
nic: "1999099602438"
pho: "0773221876"
password: "$2a$10$wRkqg$inhtJhd3k4y.01.aa/xi4z05EMIda5y47hlo3IH1u488a"
date: 2022-06-04T15:36:20.359+00:00
__v: 0

```

The bottom status bar shows system information: 82°F Rain showers, 11:20 PM, 6/4/2022, and a battery level of 20%.

MongoDB Compass - project.h4eh7.mongodb.net/ResearchProjectManagementToolDatabase.markssupervisors

Connect View Collection Help

Local

3 DBS 8 COLLECTIONS

FAVORITE

HOSTS
project-shard-00-00.h4eh7...
project-shard-00-02.h4eh7...
project-shard-00-01.h4eh7...

CLUSTER
Replica Set (atlas-7vspj0-s...
3 Nodes

EDITION
MongoDB 5.0.8 Enterprise

Filter your data

ResearchProjectManagementT...
finaldocs
groups
markspannels
markssupervisors
panels
templates
topicregdocs
topics
users

admin
local

MONGOSH

Type here to search

Documents

ResearchProjectManagementToolDatabase.markssupervisors

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 1 of 1

DOCUMENTS 1 TOTAL SIZE 226B AVG. SIZE 226B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

_id: ObjectId("629b8ee23655191760257d63")
name: "MarkingScheme_Document_2021"
description: ""
pdf: "https://res.cloudinary.com/dpmipuvk/image/upload/v1654361825/nhyraawfizhsmtliahs"
cloudinary_id_pdf: "nhyraawfizhsmtliahs"
_v: 0

82°F Rain showers 11:20 PM 6/4/2022

MongoDB Compass - project.h4eh7.mongodb.net/ResearchProjectManagementToolDatabase.markspannels

Connect View Collection Help

Local

3 DBS 8 COLLECTIONS

FAVORITE

HOSTS
project-shard-00-00.h4eh7...
project-shard-00-02.h4eh7...
project-shard-00-01.h4eh7...

CLUSTER
Replica Set (atlas-7vspj0-s...
3 Nodes

EDITION
MongoDB 5.0.8 Enterprise

Filter your data

ResearchProjectManagementT...
finaldocs
groups
markspannels
markssupervisors
panels
templates
topicregdocs
topics
users

admin
local

MONGOSH

Type here to search

Documents

ResearchProjectManagementToolDatabase.markspannels

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 1 of 1

DOCUMENTS 1 TOTAL SIZE 231B AVG. SIZE 231B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

_id: ObjectId("629b8ff043655191760257d65")
name: "markingscheme_presentation_2021"
description: ""
pdf: "https://res.cloudinary.com/dpmipuvk/image/upload/v1654361859/k3a7mb5b..."
cloudinary_id_pdf: "k3a7mb5bprfjx03cg"
_v: 0

82°F Rain showers 11:20 PM 6/4/2022

ResearchProjectManagementToolDatabase.templates

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } **OPTIONS** **FIND** **RESET** **...**

ADD DATA **VIEW** **...**

Displaying documents 1 - 3 of 3

_id	name	description	pdf	cloudinary_id_pdf	_v
629b8cc73655191760257d59	"Project_Presentation_Guidelines"	"Please Refer the guidelines documentation for project presentation"	https://res.cloudinary.com/dmipuvkg/image/upload/v1654361286/xy88mpo...	ky88mpodgmogrjhxo1	0
629b8dd33655191760257d5b	"TopicRegistrationTemplate_2022"	"Student Groups compulsory to use this template"	https://res.cloudinary.com/dmipuvkg/image/upload/v1654361386/lolcw#85...	lolcw#85fb49ebwf7yh4	0
629b8dd33655191760257d5d	"Final_Report_Template_2022"	"Student Groups should use this template to submit the final report"	https://res.cloudinary.com/dmipuvkg/image/upload/v1654361442/ofhddbm...	ofhddbmmyjofsqmuad	0

ii. Student group management and panel management – IT20133122 R.A.E. Wijenayake

ResearchProjectManagementToolDatabase.groups

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } **OPTIONS** **FIND** **RESET** **...**

ADD DATA **VIEW** **...**

Displaying documents 1 - 4 of 4

_id	l_id	member1	member2	member3	member4	_v	g_id	panel
629b9318da99a48548b9b4e	IT2014785	IT20147856 Fernando P.H.	IT20157823 Kahanawala A.T	IT20254812 Gunarathne P.P.T	IT20147856 Perera Y.M.N	0	2022_Feb_IT20147856	Panel 01
629b9406dbaa99a48548b9b51	IT20186417	IT20186417 Arachchi S.H.H	IT20127836 Bentharca S.M.P	IT20107482 Gamage W.W.C	IT20254617 Balasooriya L.D	0	2022_Feb_IT20186417	Panel 02
629b94dcdba99a48548b9b54	IT20124867	IT20124867 Punsala S.N.Y	IT20271523 Prabhush S.H.P	IT20145144 Desupura S.H.P	IT20127381 Wijesinghe A.H	0	2022_Feb_IT20124867	Panel 03

MongoDB Compass - project.b4eh7.mongodb.net/ResearchProjectManagementToolDatabase.panels

Local

HOSTS: project-shard-00-00.b4eh7... project-shard-00-02.b4eh7... project-shard-00-01.b4eh7...

CLUSTER: Replica Set (atlas-7vxp0-s... 3 Nodes

EDITION: MongoDB 5.0.8 Enterprise

Filter your data

ResearchProjectManagementT...

- finaldocs
- groups
- markspannels
- marksupervisors
- panels**
- templates
- topicregdocs
- topics
- users

admin

local

MONGOSH

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 4 of 4

DOCUMENTS 4 TOTAL SIZE 589B AVERAGE SIZE 147B INDEXES 1 TOTAL SIZE 36.0KB AVERAGE SIZE 36.0KB

_id	panelNo	pMember01	pMember02	pMember03	_v
ObjectId("629b1672db25b148101d2f34")	"Panel 01"	"Gunasekara A.M.P.P"	"Alagiyawanna A.M.T.P"	"Ranasinghe R.A."	0
ObjectId("629b3ab1c6be0d57f0bd99df")	"Panel 02"	"Dananjaya K.K.D.O"	"Iresha P.H.R"	"Perera N.A.P"	0
ObjectId("629b9661365519176b257d9b")	"Panel 03"	"C.D. Adhiketty"	"A.S. Wijenayake"	"A.D. Samaranayake"	0
ObjectId("629b96b2365519176b257da0")	"Panel 04"	"C.D. Perera"	"C.D. Dissanayake"	"C.D. Jayanayake"	0

82°F Rain showers 11:22 PM 6/4/2022

iii. Topic registration management – IT20135966 Gunasekara A.M.P.P.

MongoDB Compass - project.b4eh7.mongodb.net/ResearchProjectManagementToolDatabase.topics

Local

HOSTS: project-shard-00-00.b4eh7... project-shard-00-02.b4eh7... project-shard-00-01.b4eh7...

CLUSTER: Replica Set (atlas-7vxp0-s... 3 Nodes

EDITION: MongoDB 5.0.8 Enterprise

Filter your data

ResearchProjectManagementT...

- finaldocs
- groups
- markspannels
- marksupervisors
- topics**
- templates
- topicregdocs
- users

admin

local

MONGOSH

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 5 of 5

DOCUMENTS 5 TOTAL SIZE 2.0KB AVERAGE SIZE 410B INDEXES 1 TOTAL SIZE 36.0KB AVERAGE SIZE 36.0KB

_id	status	sName	cName	feedback	gid	area	name	description	_v	comment
ObjectId("629ae20fd25b148101d2eb7")	"Approved"	"Mr. Nihal Thissa"	"Sennil"	"Accepted"	"2022_Feb_IT20135966"	"Artificial Intelligence "	"Objective Bayesian Nets for Integrating Consistent Datasets"	"This paper addresses a data integration problem: given several mutual..."	0	"nice topic"
ObjectId("629e9719dba99a48540b9b5c")	"Pending"	"Ms. Nihal Thissa"	"Sennil"	"Not yet assigned"	"2022_Feb_IT20147856"	"Cloud computing"	"Cloud Cryptography "	"Data in the cloud is needed to be protected and secured from foreign a..."	0	"Well done"
ObjectId("629e977cdba99a48540b9b5f")	"Pending"	"Ms. Nalaka Godahewa"	"Sennil"	"Not yet assigned"	"2022_Feb_IT20147856"	"Cloud computing"	"Cloud Cryptography "	"Data in the cloud is needed to be protected and secured from foreign a..."	0	"Well done"
ObjectId("629e977cdba99a48540b9b5f")	"Pending"	"Ms. Nalaka Godahewa"	"Sennil"	"Not yet assigned"	"2022_Feb_IT20147856"	"Cloud computing"	"Cloud Cryptography "	"Data in the cloud is needed to be protected and secured from foreign a..."	0	"Well done"

82°F Rain showers 11:24 PM 6/4/2022

MongoDB Compass - project.h4eh7.mongodb.net/ResearchProjectManagementToolDatabase.topicregdocs

Local

HOSTS: project-shard-00-00.j4eh7... project-shard-00-02.j4eh7... project-shard-00-01.j4eh7...

CLUSTER: Replica Set (atlas-7vpx0-s...) 3 Nodes

EDITION: MongoDB 5.0.8 Enterprise

Filter your data

ResearchProjectManagementT...

- finaldocs
- groups
- markspannels
- markssupervisors
- panels
- templates
- topicregdocs**
- topics
- users

admin local

MONGOSH

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

OPTIONS FIND RESET

Displaying documents 1 - 9 of 9

_id	name	date	pdf	cloudinary_id_pdf	_v
ObjectId("629a3545538c9e39c43be7ad")	"t20113672"	null	"https://res.cloudinary.com/dpmipuvk/image/upload/v1654273348/uqvqgqvv..."	"uvvqgqvvdyyd1d7ongem"	0
ObjectId("629a35800538c9e39c43be7b1")	"awan"	null	"https://res.cloudinary.com/dpmipuvk/image/upload/v1654273407/uvrvbupd..."	"uvrvbupdeltgyqznoik"	0
ObjectId("629a35c0538c9e39c43be7b6")	"oko"	2022-06-03T16:24:32.566+00:00	"https://res.cloudinary.com/dpmipuvk/image/upload/v1654273471/sknval6z..."	"sknval6z4qx1st5fbee1"	0
ObjectId("629a3815538c9e39c43be7c2")	"jji"	2022-06-03T16:34:29.187+00:00	"https://res.cloudinary.com/dpmipuvk/image/upload/v1654274068/zfdny14k..."	"zfdny14kgihkempzloxi8"	0
ObjectId("629a4182de537c49fc661a94")	"cr2m7nm5deapsuzpeet"	2022-06-03T17:14:42.649+00:00	"https://res.cloudinary.com/dpmipuvk/image/upload/v1654276481/cr2m7nn..."	"cr2m7nn5deapsuzpeet"	0
ObjectId("629b2994fe5af157e086c1b9")	"e7akzogzo03dabb1pliq"	2022-06-04T00:44:52.669+00:00	"https://res.cloudinary.com/dpmipuvk/image/upload/v1654335891/e7akzog..."	"e7akzogzo03dabb1pliq"	0

TOTAL SIZE 1.7KB AVG. SIZE 197B DOCUMENTS 9 INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

82°F Rain showers 11:24 PM 6/4/2022

iv. Final document submission management & Chat management – IT20133818 Perera M.P.M.

MongoDB Compass - project.h4eh7.mongodb.net/ResearchProjectManagementToolDatabase.finaldocs

Local

HOSTS: project-shard-00-00.j4eh7... project-shard-00-02.j4eh7... project-shard-00-01.j4eh7...

CLUSTER: Replica Set (atlas-7vpx0-s...) 3 Nodes

EDITION: MongoDB 5.0.8 Enterprise

Filter your data

ResearchProjectManagementT...

- finaldocs
- groups
- markspannels
- markssupervisors
- panels
- templates
- topicregdocs
- topics
- users

admin local

MONGOSH

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

OPTIONS FIND RESET

Displaying documents 1 - 2 of 2

_id	name	date	pdf	cloudinary_id_pdf	_v
ObjectId("629a3545538c9e39c43be7ad")	"t20113672"	null	"https://res.cloudinary.com/dpmipuvk/image/upload/v1654273348/uqvqgqv..."	"uvvqgqvvdyyd1d7ongem"	0
ObjectId("629b2994fe5af157e086c1b9")	"e7akzogzo03dabb1pliq"	2022-06-04T00:44:52.669+00:00	"https://res.cloudinary.com/dpmipuvk/image/upload/v1654335891/e7akzog..."	"e7akzogzo03dabb1pliq"	0

TOTAL SIZE 403B AVG. SIZE 202B DOCUMENTS 2 INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

82°F Rain showers 11:25 PM 6/4/2022

Marking Rubric

Date	Weight	Marks (out of 5)			
Description		5	5	5	5
1. SRS Document -	10				
N/A	5				
N/A	5				
2. Frontend Development	25				
a. Viva	10				
b. demo					
1. Implemented all the components [2]					
2. UI/UX [08]					
3. Components [5]	15				
3. Backend Development	25				
a. Viva	10				
b. Demo					
1. Routes [5]					
2. DAL [5]					
3. DAO [5]	15				
4. Apps are hosted	10				
a. Frontend	5				
b. Backend	5				
5. Unit Tests	10				
a. Frontend	5				
b. Backend	5				
6. Test Cases	5				
7. Technical Report					
a. Screenshots of the UI [1]					
b. Screenshots of the MongoDB [2]					
c. Description [2]	5				

Total	100				
Comments					

Mark out of 5

- 0.0 - Section is not done
- 0.1 - There is something but entirely wrong or not relevant
- 1.0 - Relevant but wrong
- 2.0 - Slightly correct
- 3.0 - Average
- 4.0 - Good - Slight errors
- 5.0 - WOW (No errors)

Report Template.

1. Component Description (Member wise)
2. Screenshots of the Frontend (Member wise)
3. Screenshots/ Code Snippets of the Backend (Member wise)
4. Screenshots of the MongoDB Schemas (Member wise)
5. Proof of Hosting (Backend and Frontend)

When you write the report, create topics as given in the report template and subtopics from the member's SLIIT Student ID number. Please refer to the marking rubric as well.