ProConnect

Teammates:

Pavan Kumar Jonnadula – 16324822 – pvjb6c@umsystem.edu

Sri Nikhitha Boddapati – 16322565 – sb4dz@umsystem.edu

Vamsi Alapaty – 18230326 – vanny@umsystem.edu

Sukumar Bodapati – 16326105 - sb5zh@umsystem.edu

Programming Tools:

VSCode, GitHub.

Github Link: https://github.com/JVSPAVAN/WebMobile-2022Spring/tree/main/Project/Source%20Code

Video Link:

https://github.com/JVSPAVAN/WebMobile-2022Spring/tree/main/Project/Video

Introduction:

This Proconnect website acts as a centralized application of all the UMKC websites. Whenever any student wants to check their coursework, they use Canvas website and when they want to enroll for the courses, they have to login to umsystems. So, its like the student need to visit multiple websites for various information. As all the programs are available in a single application, it will save time and be more convenient for the user.

Background/Related work:

For this application, we have analyzed how canvas works like how it stores the data and update the data. For umsystem applications, we have analyzed all the screens how it works.

Define your proposed idea:

We have various applications for students at UMKC, such as canvas for course work, umsystem for personal and financial information, and teams for texting. Navigating through the many applications and logging into them is tough. It takes a long time. Instead, if we have a single application with all the functionality, users may easily navigate across the pages of the application, saving time and improving accessibility.

Methodology/Approach:

The user must login only once in this project application that we are constructing. The user may then see the modules and choose one based on his needs, such as if he wants to access his course work, he can click on the course work module, and all his course work schedule will appear. For the Front end, we have used angular framework which is a trending technology. Since it is a single page application it helps to load the application quickly. For backend we have used nodejs. In the backend, packages like b-crypt are used which helps to keep the user details confidential. MongoDB is chosen as the database since it is a no-sql database used to query large datasets like university data/student data.

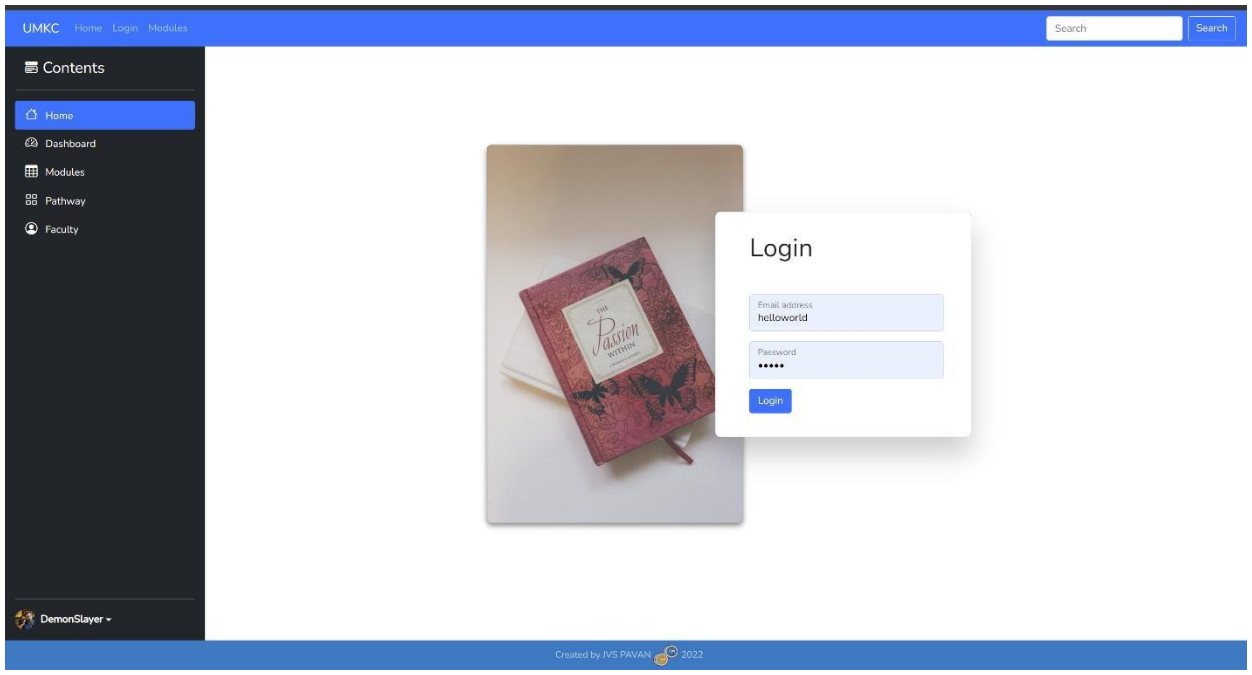
Features:

- Official UMKC info integration:
  - After logging into the application user will be navigated into home page which acts as a UMKC information screen
- Single sign-on:
  - This application that constructed will help the users to navigate through different features using Single sign-on.
- Redirection of features:
  - We can navigate through multiple screens in a single navbar.
- User signing in using JWT authentication:
  - This JWT user cryptographic algorithm which helps the student for secure login.
- User confidentiality using b-crypt o It is used for encrypting the user credentials.
- Usage of cloud database (MongoDB atlas):
  - As MongoDB Atlas is a fully managed cloud database, makes students to login at any place at any time.

Front-end:

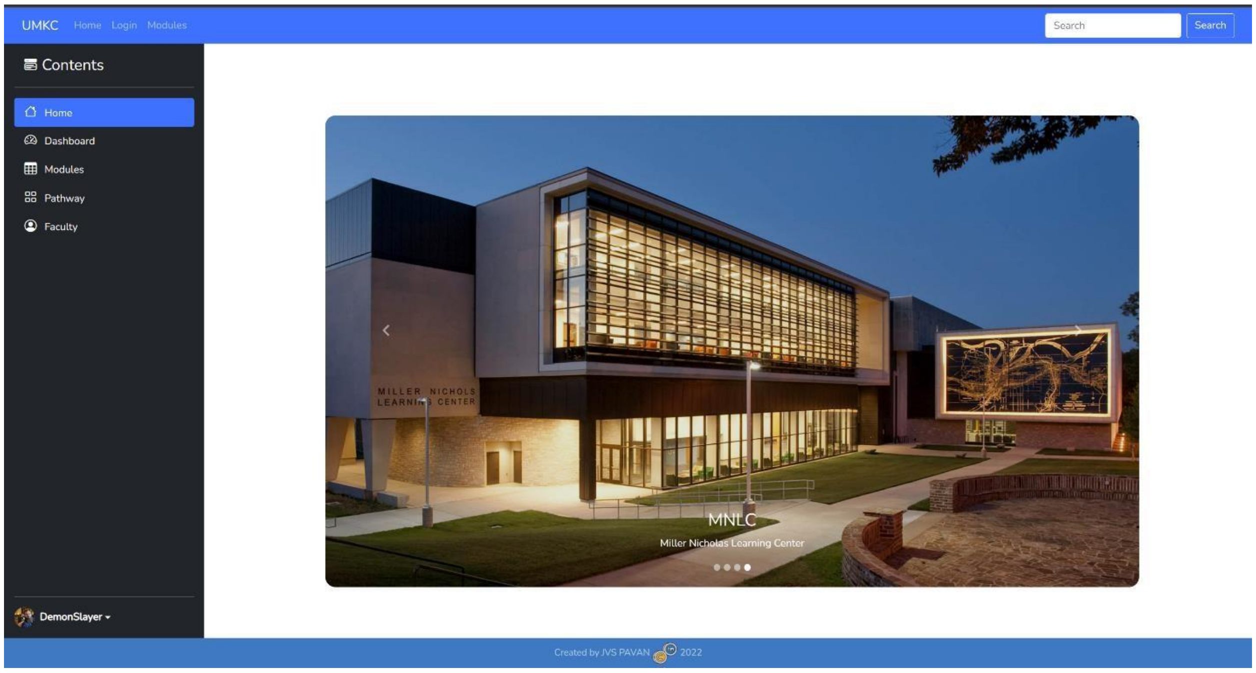
Login Screen:

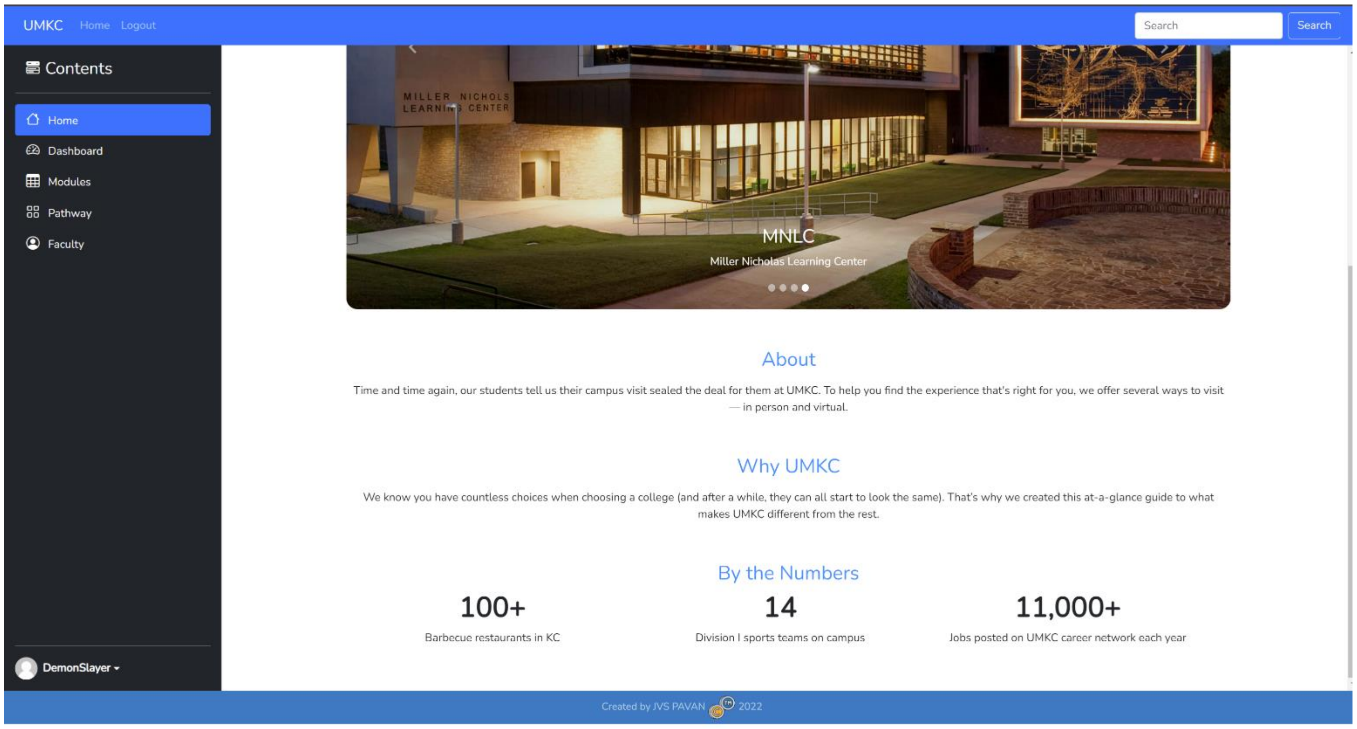
Under login screen a reactive form is used which helps the student to login. After entering the credentials when login button is clicked an Api call is made to the backend server to validate the user credentials. Here, b-crypt package is used in the backed server to decrypt the encrypted password in the database and validates with the data entered in the login form.



Home Screen:

This screen provides the information about the university. A pictorial representation of university is displayed in an image carousal. Most of the information is displayed under the paragraphs below. This acts as a information screen for the user to know more about the UMKC university.





About

Time and time again, our students tell us their campus visit sealed the deal for them at UMKC. To help you find the experience that's right for you, we offer several ways to visit — in person and virtual.

Why UMKC

We know you have countless choices when choosing a college (and after a while, they can all start to look the same). That's why we created this at-a-glance guide to what makes UMKC different from the rest.

By the Numbers

100+	14	11,000+
Barbecue restaurants in KC	Division I sports teams on campus	Jobs posted on UMKC career network each year

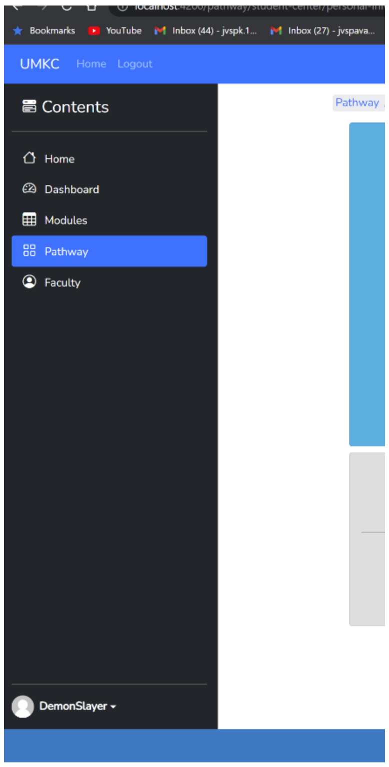
Side Navbar:

This side navbar contains the following contents:

- Home
- Dashboard
- Modules
- Pathway
- Faculty

These contents change dynamically based on the user login.

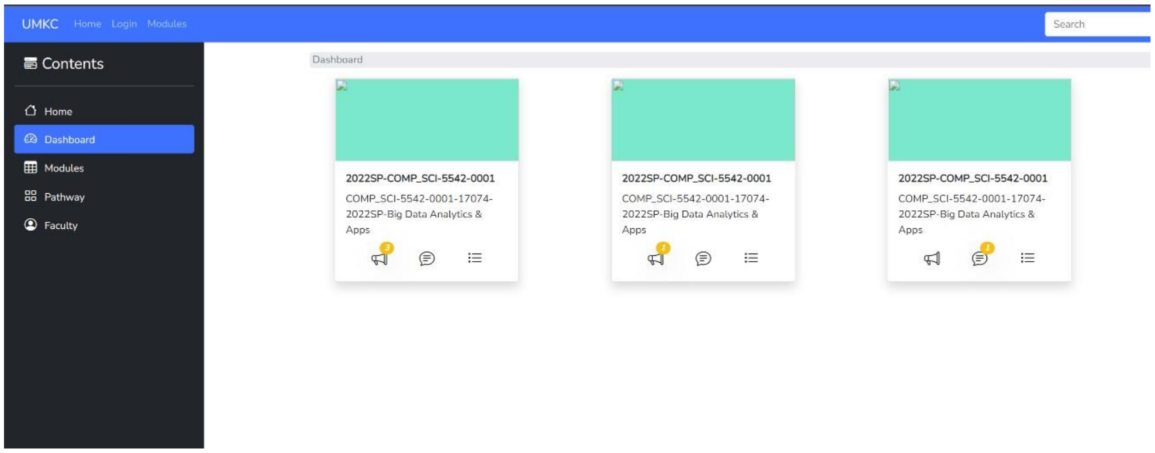
When the user is logged out the sidenav is displayed as below.



Dashboard:

Dashboard is used to display the enrolled courses of the student in the form of interactive cards. Inside cards it displays the notification badges for announcements/assignments.

- When the student selects the enrolled course, it will navigate to course home page. On the course homepage we have a side navbar which is used to navigate through the course components.
- We have different course components like announcements, discussions, grades.



2022SP-COMP_SCI-5542-0001

COMP_SCI-5542-0001-17074-2022SP-Big Data Analytics & Apps

2022SP-COMP_SCI-5542-0001

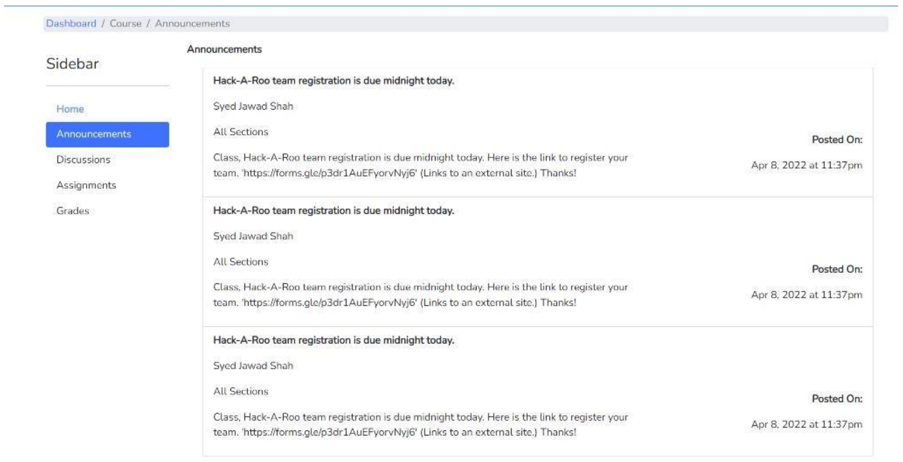
COMP_SCI-5542-0001-17074-2022SP-Big Data Analytics & Apps

2022SP-COMP_SCI-5542-0001

COMP_SCI-5542-0001-17074-2022SP-Big Data Analytics & Apps

Announcements:

The announcements of the selected course will be displayed in a list format.



Announcements

Hack-A-Roo team registration is due midnight today.

Syed Jawad Shah

All Sections

Class, Hack-A-Roo team registration is due midnight today. Here is the link to register your team. 'https://forms.gle/p3dr1AuEFyorvNyj6' (Links to an external site.) Thanks!

Posted On: Apr 8, 2022 at 11:37pm

Hack-A-Roo team registration is due midnight today.

Syed Jawad Shah

All Sections

Class, Hack-A-Roo team registration is due midnight today. Here is the link to register your team. 'https://forms.gle/p3dr1AuEFyorvNyj6' (Links to an external site.) Thanks!

Posted On: Apr 8, 2022 at 11:37pm

Hack-A-Roo team registration is due midnight today.

Syed Jawad Shah

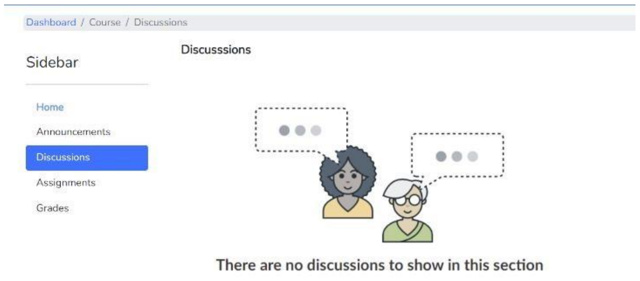
All Sections

Class, Hack-A-Roo team registration is due midnight today. Here is the link to register your team. 'https://forms.gle/p3dr1AuEFyorvNyj6' (Links to an external site.) Thanks!

Posted On: Apr 8, 2022 at 11:37pm

Discussions:

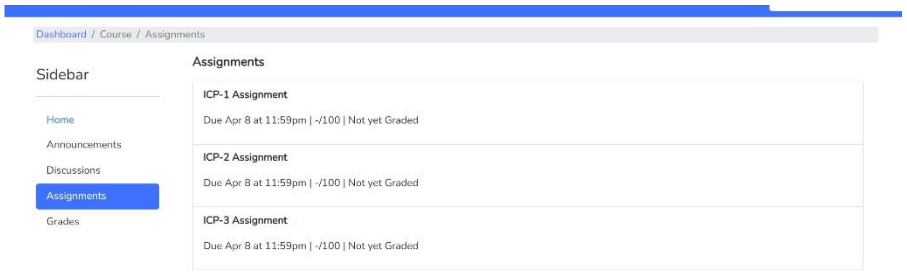
Discussions of the selected course will be displayed in a list format if there are any.



There are no discussions to show in this section

Assignments:

Assignments of the selected course will be displayed in a list format.


Dashboard / Course / Assignments

Sidebar: Home, Announcements, Discussions, Assignments, Grades

Assignments

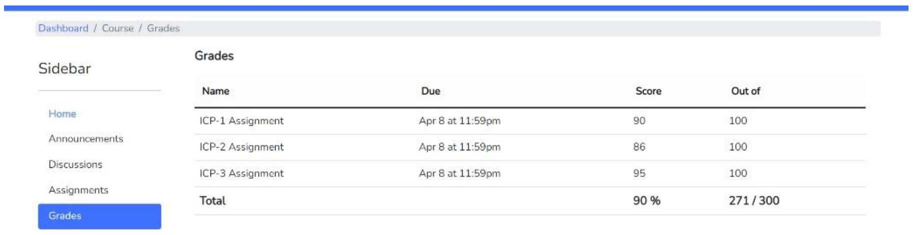
ICP-1 Assignment
Due Apr 8 at 11:59pm | -/100 | Not yet Graded

ICP-2 Assignment
Due Apr 8 at 11:59pm | -/100 | Not yet Graded

ICP-3 Assignment
Due Apr 8 at 11:59pm | -/100 | Not yet Graded

Grades:

Grades of the selected course will be displayed in a table format. The entire scores will be stored in database. A logic has been used to calculate the totals and percentages.


Dashboard / Course / Grades

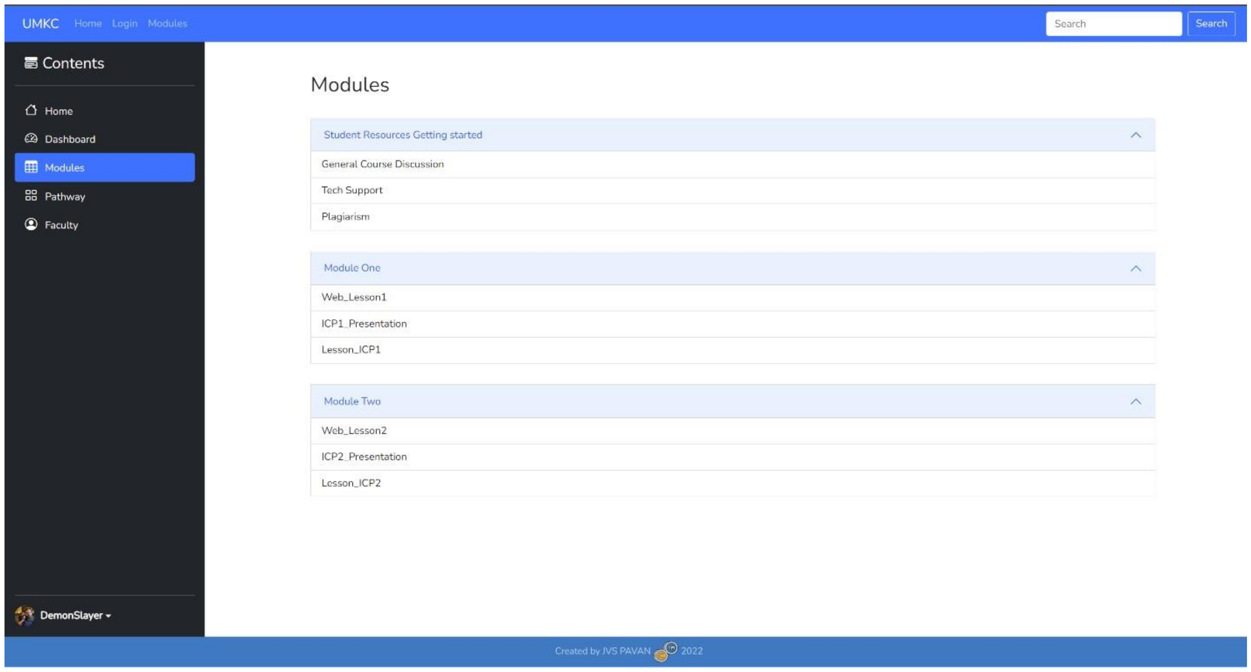
Sidebar: Home, Announcements, Discussions, Assignments, Grades

Grades

Name	Due	Score	Out of
ICP-1 Assignment	Apr 8 at 11:59pm	90	100
ICP-2 Assignment	Apr 8 at 11:59pm	86	100
ICP-3 Assignment	Apr 8 at 11:59pm	95	100
Total		90 %	271 / 300

Modules Screen:

The course contents are displayed using an accordion list. The data will be acquired from the backend API. If admin updates any course contents, it is auto updated in the UI screen.



Modules

Student Resources Getting started
- General Course Discussion
- Tech Support
- Plagiarism

Module One
- Web_Lesson1
- ICP1_Presentation
- Lesson_ICP1

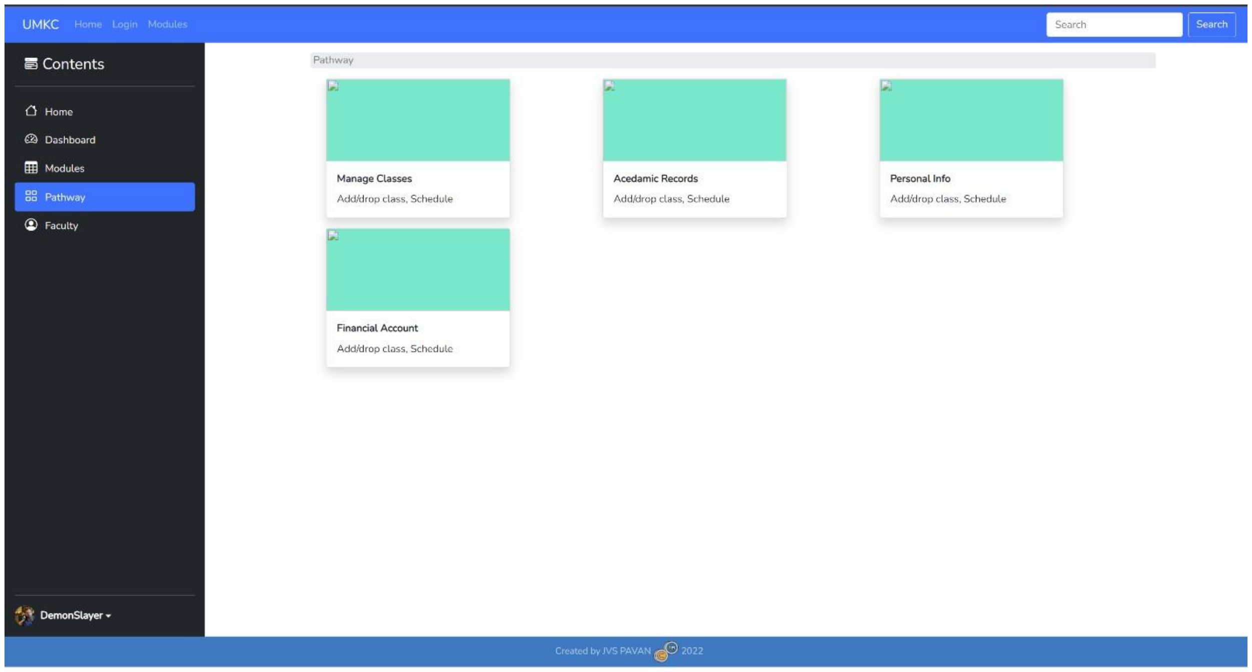
Module Two
- Web_Lesson2
- ICP2_Presentation
- Lesson_ICP2

Created by JVS PAVAN 2022

Pathway:

Four main features of the Pathway integration.

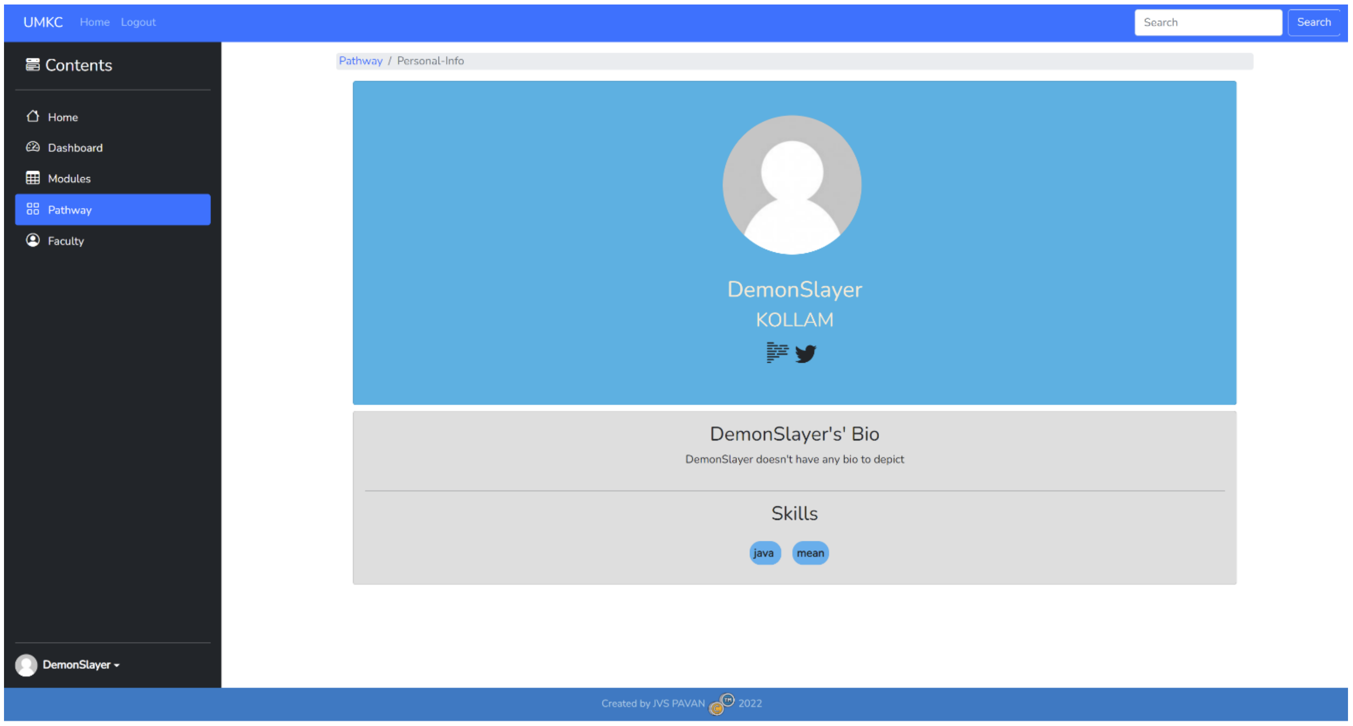
- Manage Classes
- Academic Records
- Personal Information
- Financial account



For Example, on clicking the managed classes the student will be navigated to a screen which displays all the courses available for the semester and course information like Instructor name, seats availability, timings, mode of teaching.

Personal Info:

The user details are shown in the personal info screen. Here the details alike the basic details and information Abou the loggedin user is displayed.


UMKC Home Logout | Search | Search

Contents: Home, Dashboard, Modules, Pathway, Faculty

Pathway / Personal-Info

DemonSlayer
KOLLAM

DemonSlayer's' Bio
DemonSlayer doesn't have any bio to depict

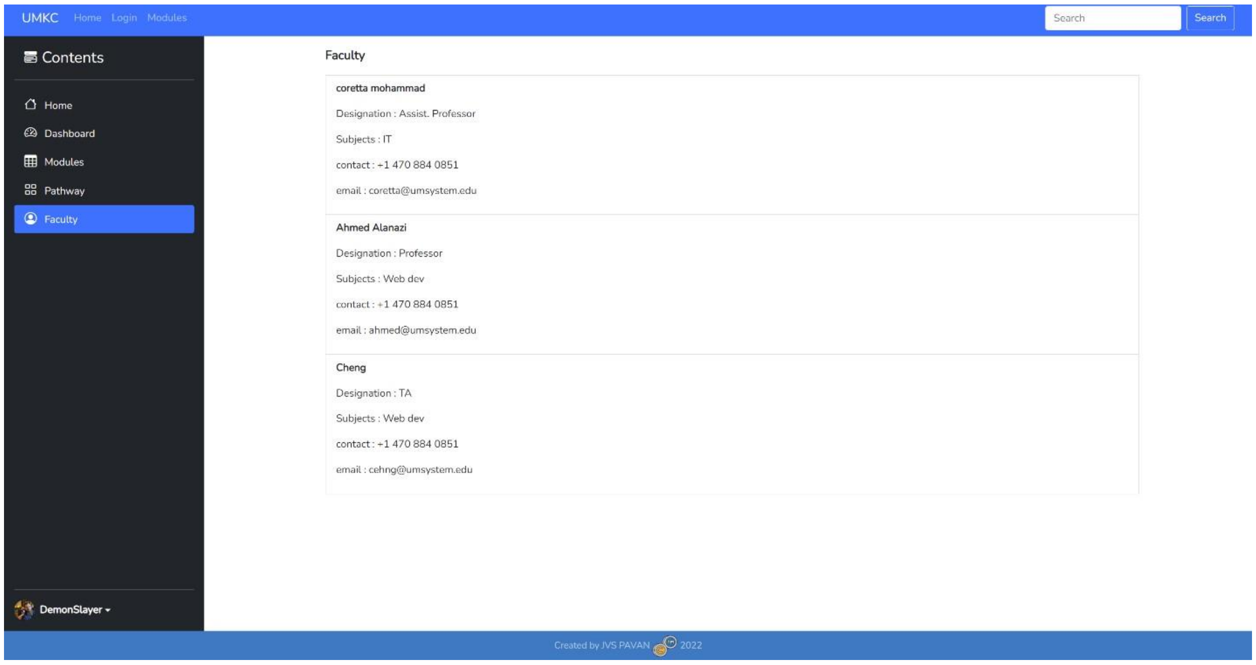
Skills
java mean

DemonSlayer

Created by JVS PAVAN 2022

Faculty:

The basic contact information of the teaching faculty will be displayed in cards.


UMKC Home Login Modules | Search | Search

Contents: Home, Dashboard, Modules, Pathway, Faculty

Faculty

coretta mohammad
Designation : Assist. Professor
Subjects : IT
contact : +1 470 884 0851
email : coretta@umsystem.edu

Ahmed Alanazi
Designation : Professor
Subjects : Web dev
contact : +1 470 884 0851
email : ahmed@umsystem.edu

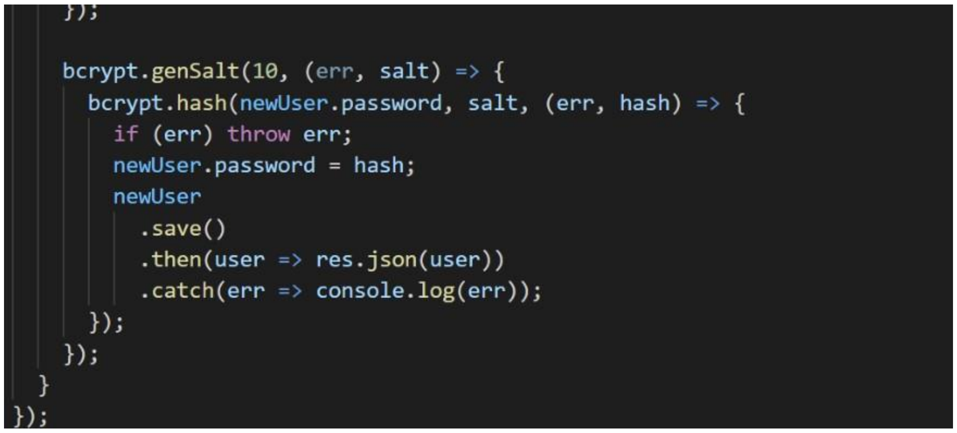
Cheng
Designation : TA
Subjects : Web dev
contact : +1 470 884 0851
email : cehng@umsystem.edu

DemonSlayer

Created by JVS PAVAN 2022

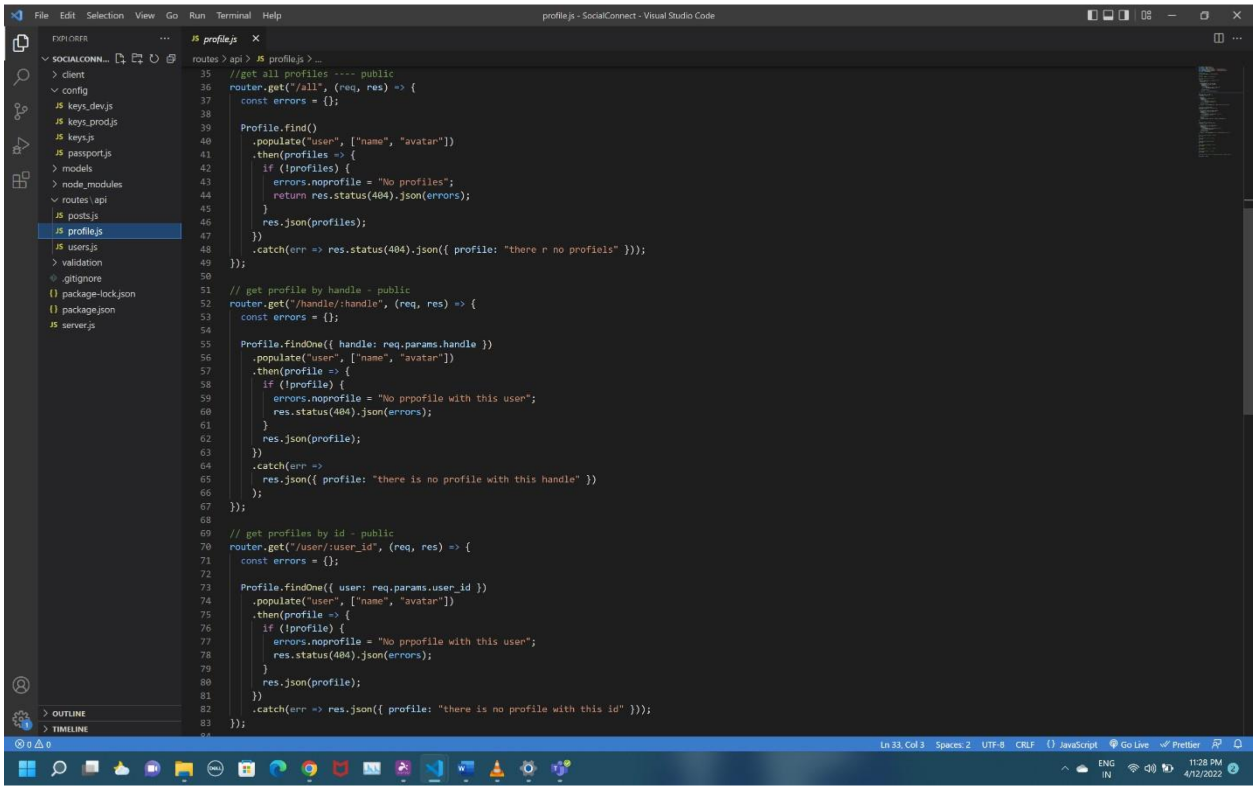
Back-end:

- As part of backend server nodejs application is created. To encrypt or decrypt the user credentials b-crypt package is used. To make the application more secure JWT authentication is integrated.



```
    });

    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))
          .catch(err => console.log(err));
      });
    });
  }
});
```

- Individual files are used to integrate the respective UI modules to the respective backend servers.
- For addition of courses or any information in the application, CRUD (Create, Read, Update and Delete) operation Api calls were implemented.



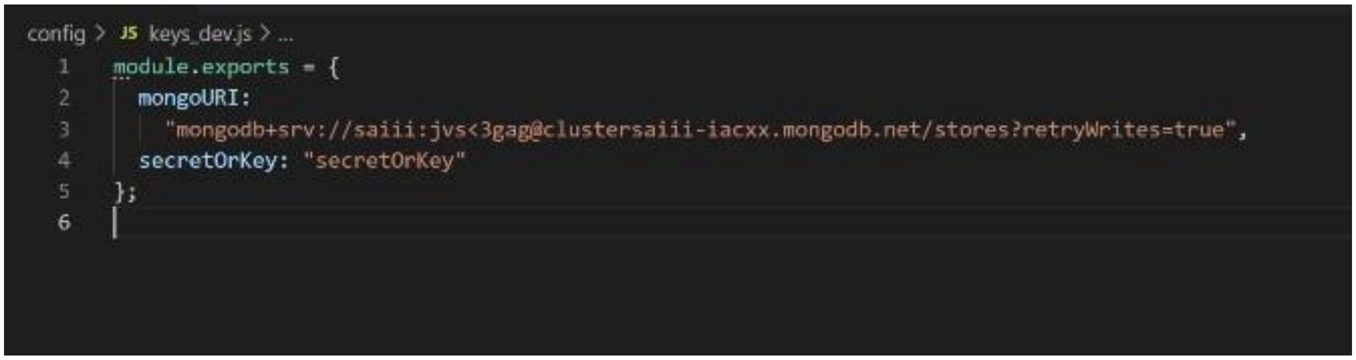
```
//get all profiles ---- public
router.get("/all", (req, res) => {
  const errors = {};

  Profile.find()
    .populate("user", ["name", "avatar"])
    .then(profiles => {
      if (!profiles) {
        errors.noprofile = "No profiles";
        return res.status(404).json(errors);
      }
      res.json(profiles);
    })
    .catch(err => res.status(404).json({ profile: "there r no profiels" }));
});

// get profile by handle - public
router.get("/handle/:handle", (req, res) => {
  const errors = {};

  Profile.findOne({ handle: req.params.handle })
    .populate("user", ["name", "avatar"])
    .then(profile => {
      if (!profile) {
        errors.noprofile = "No prpofile with this user";
        res.status(404).json(errors);
      }
      res.json(profile);
    })
    .catch(err =>
      res.json({ profile: "there is no profile with this handle" })
    );
});

// get profiles by id - public
router.get("/user/:user_id", (req, res) => {
  const errors = {};

  Profile.findOne({ user: req.params.user_id })
    .populate("user", ["name", "avatar"])
    .then(profile => {
      if (!profile) {
        errors.noprofile = "No prpofile with this user";
        res.status(404).json(errors);
      }
      res.json(profile);
    })
    .catch(err => res.json({ profile: "there is no profile with this id" }));
});
```

- Model classes are created for each feature that are implemented in the project.
- All the API(Get, Put, Post) calls will return the response in the Json object.

Database:

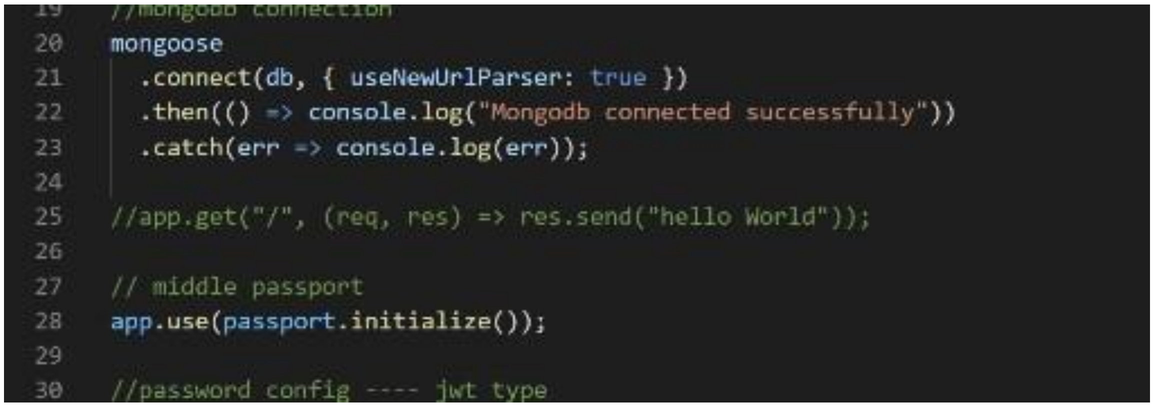
For the database part, used MongoDb Atlas which is a cloud-based storage database. Created multiple tables for the modules like student table which contains student personal information, courses selected. Similarly, created tables for courses, grades, faculty information.

MongoDB integration is as follows:



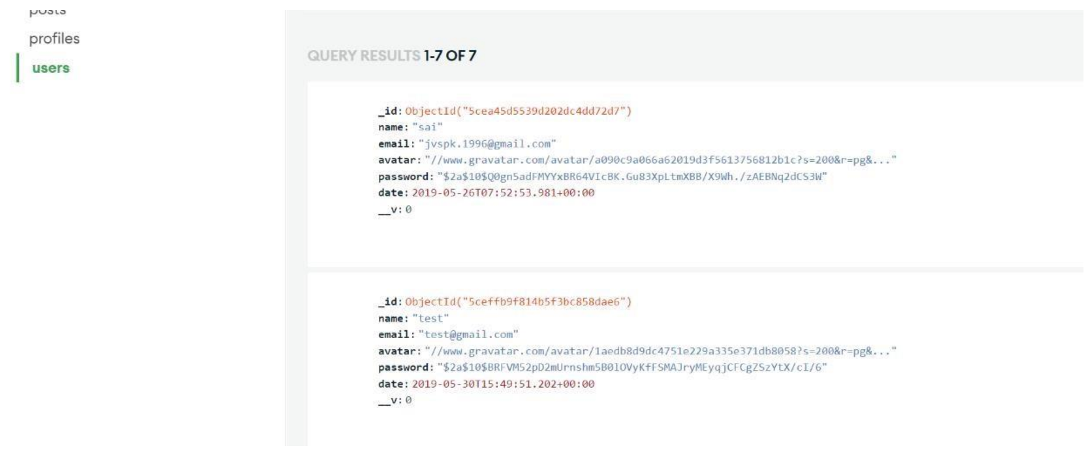
```
module.exports = {
  mongoURI:
    "mongodb+srv://saiii:jvs<3gag@clustersaiii-iacxx.mongodb.net/stores?retryWrites=true",
  secretOrKey: "secretOrKey"
};

```

Using Mangoose as follows:



```
//mongodb connection
mongoose
  .connect(db, { useNewUrlParser: true })
  .then(() => console.log("Mongodb connected successfully"))
  .catch(err => console.log(err));

//app.get("/", (req, res) => res.send("hello World"));

// middle passport
app.use(passport.initialize());

//password config ---- jwt type
```

User Data structure is as follows:



```
_id: ObjectId("5cea45d5539d202dc4dd72d7")
name: "sai"
email: "jvspk.1996@gmail.com"
avatar: "//www.gravatar.com/avatar/a090c9a066a62019d3f5613756812b1c?s=200&r=pg&..."
password: "$2a$10$Q0gn5adFMYYxBR64VIcBK.Gu83XpLtmXBB/X9Wh./zAEBNq2dCS3W"
date: 2019-05-26T07:52:53.981+00:00
__v: 0

_id: ObjectId("5ceffb9f814b5f3bc858dae6")
name: "test"
email: "test@gmail.com"
avatar: "//www.gravatar.com/avatar/1aedb8d9dc4751e229a335e371db8058?s=200&r=pg&..."
password: "$2a$10$BRFVM52pD2mUrnshm5B01OVyKfFSMAJryMEyqjCFCgZSzYtX/cI/6"
date: 2019-05-30T15:49:51.202+00:00
__v: 0
```

Modules and Work sharing:

Names	Modules
Jonnadula Pavan Kumar	Login Screen and Backend
Vamsi Alapaty	Pathway which includes four internal screens
Srinikhitha Boddapati	Home, Canvas and Faculty screens
Sukumar Bodapati	MongoDB integration and Modules screen

Blockage:

- Time is the only constraint for the creativity

Conclusion:

This mean stack application helps the user to access the major features of Multiple UMKC websites like umsystem/Canvas, umkc connect, Pathway.

Future Work:

- Hand shake integration