Web Lab Assignment

Team:

* Sai Sampath Kumar Raigiri - 32
* Anvesh Mandadi - 22
* Chandra Shekar - 30

SERVER:

File : WWW

#!/usr/bin/env node

/\*\*

 \* Module dependencies.

 \*/

var app = require('../app');

var debug = require('debug')('server:server');

var http = require('http');

/\*\*

 \* Get port from environment and store in Express.

 \*/

var port = normalizePort(process.env.PORT || '3000');

app.set('port', port);

/\*\*

 \* Create HTTP server.

 \*/

var server = http.createServer(app);

/\*\*

 \* Listen on provided port, on all network interfaces.

 \*/

server.listen(port);

server.on('error', onError);

server.on('listening', onListening);

/\*\*

 \* Normalize a port into a number, string, or false.

 \*/

function normalizePort(val) {

  var port = parseInt(val, 10);

  if (isNaN(port)) {

    // named pipe

    return val;

  }

  if (port >= 0) {

    // port number

    return port;

  }

  return false;

}

/\*\*

 \* Event listener for HTTP server "error" event.

 \*/

function onError(error) {

  if (error.syscall !== 'listen') {

    throw error;

  }

  var bind = typeof port === 'string'

    ? 'Pipe ' + port

    : 'Port ' + port;

  // handle specific listen errors with friendly messages

  switch (error.code) {

    case 'EACCES':

      console.error(bind + ' requires elevated privileges');

      process.exit(1);

      break;

    case 'EADDRINUSE':

      console.error(bind + ' is already in use');

      process.exit(1);

      break;

    default:

      throw error;

  }

}

/\*\*

 \* Event listener for HTTP server "listening" event.

 \*/

function onListening() {

  var addr = server.address();

  var bind = typeof addr === 'string'

    ? 'pipe ' + addr

    : 'port ' + addr.port;

  debug('Listening on ' + bind);

}

File: INDEX.JS

module.exports = function (app, db) {

    let student\_details = db.model('student\_details');

    let profile\_details = db.model('profile\_details');

    let course\_details = db.model('course\_details');

    app.get('/course/search',(req,res)=>{

        let search\_by = 'professorName';

        let query = {};

        query[search\_by] = { $regex: 'Lee', $options: 'i' };

        course\_details.find(query).exec((err, course) => {

            if (!err) {

                res.send({

                    result: "Success",

                    data: course

                });

            } else {

                res.status(400).send({

                    result: "Failure",

                    message: "Error in fetching students list",

                    error: err.message

                });

            }

        });

    });

    app.get('/friends/search',(req,res)=>{

        profile\_details.find().exec((err, profile) => {

            if (!err) {

                res.send({

                    result: "Success",

                    data: profile

                });

            } else {

                res.status(400).send({

                    result: "Failure",

                    message: "Error in fetching Profile list",

                    error: err.message

                });

            }

        });

    });

    //api to search student details

    app.get('/signin',(req,res)=>{

        let search\_text = req.query.searchtext;

        let search\_by = 'name';

        let query = {};

        console.log(search\_text);

        query[search\_by] = { $regex:search\_text, $options: 'i' };

        student\_details.find(query).exec((err, students) => {

            if (!err) {

                console.log(students);

                res.send({

                    result: "Success",

                    data: students

                });

            } else {

                res.status(400).send({

                    result: "Failure",

                    message: "Error in fetching students list",

                    error: err.message

                });

            }

        });

    });

    //api to create student details

    app.post('/signup/register',(req,res) => {

        let student = req.body;

        let stud\_details = new student\_details({

            name: student.name ,

            email: student.email,

            password: student.password,

            created\_on:new Date(),

            updated\_on:new Date()

        });

        stud\_details.save((err, student) => {

            if (!err) {

                res.send({

                    result: "Success",

                    message: "Details saved successfully"

                });

            } else {

                res.status(400).send({

                    result: "Failure",

                    message: "Error in creating student",

                    error: err.message

                });

            }

        })

    });

    app.post('/courseDetails/save',(req,res) => {

        let course = req.body;

        let co\_details = new course\_details({

            courseName:course.courseName,

            professorName:course.professorName,

            credits:course.credits,

            description:course.description

        });

        co\_details.save((err, course) => {

            if (!err) {

                res.send({

                    result: "Success",

                    message: "Details saved successfully"

                });

            } else {

                res.status(400).send({

                    result: "Failure",

                    message: "Error in creating Course Details",

                    error: err.message

                });

            }

        })

    });

    app.post('/profile/Fillup',(req,res) => {

        let profile = req.body;

        let prof\_details = new profile\_details({

            userId: profile.userId,

            name: profile.name,

            email: profile.email,

            contact: profile.contact,

            age: profile.age,

            dob: profile.dob,

            major: profile.major,

            yearJoined: profile.yearJoined,

            department: profile.department,

            university: profile.university,

            address: profile.address,

            image: profile.image,

            created\_on:new Date(),

            updated\_on:new Date()

        });

        prof\_details.save((err, profile) => {

            if (!err) {

                res.send({

                    result: "Success",

                    message: "Details saved successfully"

                });

            } else {

                res.status(400).send({

                    result: "Failure",

                    message: "Error in creating student",

                    error: err.message

                });

            }

        })

    });

};

FILE: COURSE\_DETAILS.JS

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

//schema for student form details

const course\_details = new Schema({

    courseName: String ,

    professorName: String,

    credits: String,

    description: String,

});

module.exports =  mongoose.model('course\_details', course\_details);

FILE: PROFILE\_DETAIL.JS

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

//schema for profile form details

const profile\_details = new Schema({

    userId: String,

    name: String,

    email: String,

    contact: String,

    age: Number,

    university: String,

    address: String,

    dob: String,

    image: String,

    yearJoined: String,

    department: String,

    major: String,

    created\_on:Date,

    updated\_on:Date

});

module.exports =  mongoose.model('profile\_details', profile\_details);

FILE: STUDENT\_DETAILS.JS

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

//schema for student details

const student\_details = new Schema({

    name: String,

    email: String,

    password: String,

    created\_on:Date,

    updated\_on:Date

});

module.exports =  mongoose.model('student\_details', student\_details);

FILE: APP.JS

let createError = require('http-errors'),

    express = require('express'),

    path = require('path'),

    cookieParser = require('cookie-parser'),

    logger = require('morgan');

const db = require('mongoose'),

    cors = require('cors'),

    config = require('./config'),

    db\_string = 'mongodb://'+config.db\_user+':'+config.db\_pswd+'@ds331558.mlab.com:31558/web\_lab1';

//initializing schema

require('./models/student\_details');

require('./models/profile\_details');

require('./models/course\_details');

//connection for DB

var db\_promise = db.connect(db\_string,{ useNewUrlParser: true });

db\_promise.then((data) => {

    console.log("Database connection is successfull !");

}).catch(reason => {

    console.log("Database connection is unsuccessfull ! : "+reason.message);

})

var app = express();

// view engine setup

app.set('views', path.join(\_\_dirname, 'views'));

app.set('view engine', 'ejs');

app.use(logger('dev'));

app.use(express.json());

app.use(express.urlencoded({ extended: false }));

app.use(cookieParser());

app.use(express.static(path.join(\_\_dirname, 'public')));

app.use(cors());

// Rest APIs

require('./controllers/')(app, db);

// catch 404 and forward to error handler

app.use(function(req, res, next) {

  next(createError(404));

});

// error handler

app.use(function(err, req, res, next) {

  // set locals, only providing error in development

  res.locals.message = err.message;

  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page

  res.status(err.status || 500);

  res.render('error');

});

module.exports = app;

FILE: CONFIG.JS

var config = {

    "db\_user":"test",

    "db\_pswd":"test123"

}

module.exports = config;

CLIENT: (ANGULAR)

App-routing.component.ts

import { NgModule, Component } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

import {SignInComponent} from './sign-in/sign-in.component'

import {SignupComponent} from './signup/signup.component'

import { OnlineCoursesComponent } from './online-courses/online-courses.component';

import { AnnouncementsComponent } from './announcements/announcements.component';

import { VideoLessonsComponent } from './video-lessons/video-lessons.component';

import { DiscoverBooksComponent } from './discover-books/discover-books.component';

import { ProfileComponent } from './profile/profile.component';

import { FriendsComponent } from './friends/friends.component';

import { CoursesComponent } from './courses/courses.component';

import { EditProfileComponent } from './edit-profile/edit-profile.component';

import { CourseEditComponent } from './course-edit/course-edit.component';

import { CalenderComponent } from './calender/calender.component';

const routes: Routes = [

  { path: 'SignUp', component: SignupComponent},

  { path: 'SignIn', component: SignInComponent},

  { path: 'Home', component: CoursesComponent},

  { path: 'Friends', component: FriendsComponent},

  { path: 'DiscoverBooks', component: DiscoverBooksComponent},

  { path: 'VideoLessons', component: VideoLessonsComponent},

  { path: 'Announcements', component: AnnouncementsComponent},

  { path: 'OnlineCourses', component: OnlineCoursesComponent},

  { path: 'Profile', component:ProfileComponent},

  { path: 'editProfile', component:EditProfileComponent},

  { path: 'edit\_course', component: CourseEditComponent},

  { path: 'calender', component: CalenderComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes, { enableTracing: true })],

  exports: [RouterModule]

})

export class AppRoutingModule { }

export const routingComponents = [SignInComponent, SignupComponent, OnlineCoursesComponent, ProfileComponent,VideoLessonsComponent, FriendsComponent, DiscoverBooksComponent,CoursesComponent, AnnouncementsComponent,EditProfileComponent, CourseEditComponent, CalenderComponent,]

app-component.css

body{

    margin: 0;

}

nav ul{

    position: sticky;

    top:0;

    left:0;

    list-style-type: none;

    margin: 0;

    padding: 0;

    overflow: hidden;

    background-color: gray;

    /\* border-color: black;

    border: 5px;

    border-style: solid; \*/

}

nav ul li{

    display: inline;

    float: left;

}

nav ul li.right{

    float: right;

}

nav ul li a{

    display: block;

    text-decoration: none;

    color: black;

    padding: 14px 16px;

    text-align: center;

}

nav ul li a.active{

    background-color: darkcyan;

}

nav ul li a:hover:not(.active){

background-color: brown;

}

@media screen and (max-width: 600px) {

    nav ul li.right{float: none}

    nav ul li {float: none;}

}

header

{

    background-color: black;

    color: white;

    padding: 20px;

    text-align: center;

    font-style: italic;

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

    font-size: 40px;

}

.sticky {

  position: fixed;

  top: 0;

  width: 100%;

}

Appcomponent.specs.ts

import { TestBed, async } from '@angular/core/testing';

import { RouterTestingModule } from '@angular/router/testing';

import { AppComponent } from './app.component';

describe('AppComponent', () => {

  beforeEach(async(() => {

    TestBed.configureTestingModule({

      imports: [

        RouterTestingModule

      ],

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  }));

  it('should create the app', () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.debugElement.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'StudentWorld'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.debugElement.componentInstance;

    expect(app.title).toEqual('StudentWorld');

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.debugElement.nativeElement;

    expect(compiled.querySelector('.content span').textContent).toContain('StudentWorld app is running!');

  });

});

App.component.ts

import { Component, OnInit } from '@angular/core';

import { Router } from '@angular/router';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit {

  title = 'Student World';

  constructor(private router: Router) { }

  ngOnInit(){

    this.router.navigate(['/SignUp']);

  }

}

App.module.ts

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { NgbModule } from '@ng-bootstrap/ng-bootstrap';

import { HttpClientModule, HttpClient } from '@angular/common/http';

import { AppRoutingModule, routingComponents } from './app-routing.module';

import { AppComponent } from './app.component';

import { FooterComponent } from './footer/footer.component';

import { HomeComponent } from './home/home.component'

import { FormsModule } from '@angular/forms';

@NgModule({

  declarations: [

    AppComponent,

    routingComponents,

    FooterComponent,

    HomeComponent,

  ],

  imports: [

    NgbModule,

    BrowserModule,

    AppRoutingModule,

    FormsModule,

    HttpClientModule,

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Announcement.component.ts

import { Component, OnInit } from '@angular/core';

@Component({

  selector: 'app-announcements',

  templateUrl: './announcements.component.html',

  styleUrls: ['./announcements.component.css']

})

export class AnnouncementsComponent implements OnInit {

  constructor() { }

  ngOnInit() {

  }

}

Announcement.component.html

<app-home></app-home>

<div class="card-deck">

    <div class="card" style="margin: 40px">

      <img src="https://info.umkc.edu/unews/wp-content/uploads/2019/09/Performer-standing-on-stage-in-theatre-620x400.jpg" class="card-img-top" alt="...">

      <a href="https://info.umkc.edu/unews/the-hangups-of-center-stage/"><div class="card-body">

        <h5 class="card-title">The hangups of center stage</h5>

        <p class="card-text">As you are taking careful steps onto stage, the curtains open, revealing a bunch of curious faces. Then, all excitement fades, replaced with existential dread. At its worst, communication...</p>

        <p class="card-text"><small class="text-muted">October 4, 2019</small></p>

      </div></a>

    </div>

    <div class="card" style="margin: 40px">

      <img src="https://ridekc.org/assets/uploads/blog-images/\_large/buspartner2019\_banner.jpg" class="card-img-top" alt="...">

      <a href="https://ridekc.org/blog/business-partner-2019"><div class="card-body">

        <h5 class="card-title">KC’s Scott Smith and Michael Debacker honored with Business Partner Award</h5>

        <p class="card-text">The 2019 winners of the Spirit of Rosa Parks Business Partner Award are Scott Smith & Michael DeBacker, Co-Chairs of the KC Chamber’s BIG 5 Transportation Initiative. </p>

        <p class="card-text"><small class="text-muted">September 06, 2019</small></p>

      </div></a>

    </div>

    <div class="card" style="margin: 40px">

      <img src="https://ridekc.org/assets/uploads/blog-images/\_homepage/rp\_schools\_sq.jpg" class="card-img-top" alt="...">

      <a href="https://ridekc.org/blog/superintendents-community-partner-award">

      <div class="card-body">

        <h5 class="card-title">Superintendents Honored with Community Partner Award</h5>

        <p class="card-text">The Community Partner Award was presented to a trifecta of outstanding school superintendents.</p>

        <p class="card-text"><small class="text-muted">September 05, 2019</small></p>

      </div></a>

    </div>

</div>

<div class="card-deck">

    <div class="card" style="margin: 40px">

      <img src="https://info.umkc.edu/unews/wp-content/uploads/2019/10/Annotation-2019-10-06-152605-620x400.png" class="card-img-top" alt="...">

      <a href="https://info.umkc.edu/unews/concert-review-dorian-electra/"><div class="card-body">

        <h5 class="card-title">Concert Review: Dorian Electra</h5>

        <p class="card-text">It’s Saturday night in Kansas City, and all the most glittery, most online, most tight-pants-wearing young people are in one spot: the Stray Cat Film Center for Dorian Electra’s “Flamboyant” Tour.

            The venue was intimate – a sold out show at an art house “microcinema,” with only a few hundred spots available. </p>

        <p class="card-text"><small class="text-muted">October 8, 2019</small></p>

      </div></a>

    </div>

    <div class="card" style="margin: 40px" style="width: 360px">

      <img src="https://info.umkc.edu/unews/wp-content/uploads/2019/10/Joker-620x400.jpg" class="card-img-top" alt="...">

      <a href="https://info.umkc.edu/unews/joker-review-the-clown-prince-reigns/"><div class="card-body" style="width: 360px">

        <h5 class="card-title">‘Joker’ review: The clown prince reigns</h5>

        <p class="card-text">“Joker” tries to be a lot of things: an “anti-comic book movie,” a loving tribute to the early films of Martin Scorsese, a scathing indictment on the modern world and its treatment of the disenfranchised. At its core, it attempts to be a cautionary tale.....</p>

        <p class="card-text"><small class="text-muted">October 7, 2019</small></p>

      </div></a>

    </div>

    <div class="card" style="margin: 40px">

      <img src="https://info.umkc.edu/unews/wp-content/uploads/2019/10/KC-GUN-STORY-620x400.jpg" class="card-img-top" alt="...">

      <a href="https://info.umkc.edu/unews/the-future-of-gun-legislation-in-kansas-city-and-missouri/">

      <div class="card-body">

        <h5 class="card-title">The future of gun legislation in Kansas City and Missouri</h5>

        <p class="card-text">All it took was one man with a gun for many Kansas Citians to stay home on First Friday last week, paralyzed with fear. With 112 gun related homicides reported in Kansas City in 2019 according to the Kansas City Star, this fear seems...</p>

        <p class="card-text"><small class="text-muted">October 8, 2019</small></p>

      </div></a>

    </div>

</div>

<div class="card-deck">

    <div class="card" style="margin: 40px">

      <img src="https://info.umkc.edu/unews/wp-content/uploads/2019/10/College-Movies-620x400.png" class="card-img-top" alt="...">

      <a href="https://info.umkc.edu/unews/college-is-it-just-like-the-movies/">

        <div class="card-body">

        <h5 class="card-title">College: is it just like the movies?</h5>

        <p class="card-text">You’ve probably seen movies set around a college campus, and they’ve likely set some sort of expectation for you of what college life is like. But the question is, are these movies exactly relatable?</p>

        <p class="card-text"><small class="text-muted">October 8, 2019</small></p>

      </div></a>

    </div>

    <div class="card" style="margin: 40px">

      <img src="https://info.umkc.edu/unews/wp-content/uploads/2019/10/GTY\_Missouri\_State\_Capitol\_Building\_ml\_160309\_16x9\_1600-620x400.jpg" class="card-img-top" alt="...">

      <a href="https://info.umkc.edu/unews/does-missouri-care-about-college-funding-for-higher-education-hangs-in-the-balance/">

      <div class="card-body">

        <h5 class="card-title">Does Missouri care about college? Funding for higher education hangs in the balance</h5>

        <p class="card-text">The state of higher-education funding in Missouri is worrisome.

            Each year, Missouri legislators decide how much of the state budget to allocate for higher education. While there are constraints that dictate how much the state can spend, primarily that only one-third of the total state budget is discretionary, funding has consistently lagged behind national standards, as seen in the State Higher Education Finance (SHEF) report....</p>

        <p class="card-text"><small class="text-muted">October 7, 2019</small></p>

      </div></a>

    </div>

    <div class="card" style="margin: 40px">

      <img src="https://info.umkc.edu/unews/wp-content/uploads/2019/10/Schomers\_Jousting-620x400.jpg" class="card-img-top" alt="...">

      <a href="https://info.umkc.edu/unews/kansas-city-volleyball-falls-in-uphill-battle-against-grand-canyon-rebounds-with-victory-over-chicago-state/"><div class="card-body">

        <h5 class="card-title">Kansas City Volleyball falls in uphill battle against Grand Canyon, rebounds with victory over Chicago State</h5>

        <p class="card-text">Kansas City Volleyball battled hard in its third conference game against Grand Canyon University last week. With the Sept. 23-29 WAC Offensive Player of the Week Ty Lukes combating the WAC Defensive Player of the Week Klaire Mitchell, the Roos came up short, losing three sets to one.</p>

        <p class="card-text"><small class="text-muted">October 6, 2019</small></p>

      </div></a>

    </div>

</div>

<app-footer></app-footer>

Calender.component.html:

<app-home></app-home>

<iframe src="https://calendar.google.com/calendar/embed?src=rsaisampathkumar%40gmail.com&ctz=America%2FChicago" style="border: 50; margin-left: 100px" width="1200" height="900" frameborder="50" scrolling="no"></iframe>

<app-footer></app-footer>

Calender.component.ts:

import { Component, OnInit } from '@angular/core';

@Component({

  selector: 'app-calender',

  templateUrl: './calender.component.html',

  styleUrls: ['./calender.component.css']

})

export class CalenderComponent implements OnInit {

  constructor() { }

  ngOnInit() {

  }

}

CourseEdit.component.html

<app-home></app-home>

<div class="card" >

    <br>

  <form (ngSubmit)="onRegister()">

            <div class="col-sm-6">

                <div class="card">

                    <div class="card-body">

                        <div class="row">

                            <div class="col-md-6">

                                <label>Course Name:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="courseName" name="name" id="name" placeholder="Please Enter Your Name">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Professor Name:</label>

                            </div>

                            <div class="col-md-6">

                                <input  [(ngModel)]="professorName"name="pname" id="pname" placeholder="Please Enter Your Professor Name">

                            </div>

                            </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Credits:</label>

                            </div>

                            <div class="col-md-6">

                                <input  [(ngModel)]="credits" name="credits" id="credits" placeholder="Please Enter the Number of Credits">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Description:</label>

                            </div>

                            <div class="col-md-6">

                                <input  [(ngModel)]="description" name="description" id="description" placeholder="Please Enter Course Description">

                            </div>

                        </div>

                  </div>

                </div>

              <button style="height: 100%; " routerLink="/edit\_course">Save</button>

            </div>

  </form>

</div>

<app-footer></app-footer>

courseEdit.component.html

import { Component, OnInit } from '@angular/core';

import {Observable} from 'rxjs';

import {HttpClient} from '@angular/common/http';

@Component({

  selector: 'app-course-edit',

  templateUrl: './course-edit.component.html',

  styleUrls: ['./course-edit.component.css']

})

export class CourseEditComponent implements OnInit {

  public courseName: string;

  public professorName: string;

  public credits: string;

  public description: string;

  public url: string

  message: Observable<any>

  constructor(private http: HttpClient) { }

  onRegister() {

    console.log('hello');

    this.url = 'http://127.0.0.1:3000/courseDetails/save'

    this.http.post(this.url, {

      courseName: this.courseName ,

      professorName: this.professorName,

      credits: this.credits,

      description: this.description,

    })

      .subscribe(

        (res: any) => {

          this.message = res.message;

          alert(this.message);

          console.log(this.message);

        }

      );

  }

  ngOnInit() {

  }

}

Courses.html

<app-home></app-home>

<div class="card-deck">

    <div \*ngFor="let student of result" class="card" style="margin: 15px">

      <div class="card-body">

        <h5 class="card-title">{{student.courseName}}</h5>

        <p><b>Professor:</b> {{student.professorName}}</p>

        <p class="card-text">{{student.description}}</p>

      </div>

    </div>

</div>

<app-footer></app-footer>

DiscoverBooks.component.html

<app-home></app-home>

<div class="card-deck">

  <div class="card" style="margin: 40px">

    <img src="https://image.slidesharecdn.com/advancese-170424194409/95/advance-software-engineering-1-638.jpg?cb=1493063160" class="card-img-top" alt="...">

    <a href="http://cs-courses.mines.edu/csci370/"><div class="card-body">

      <h5 class="card-title">Advance Software Enginnering</h5>

      <p class="card-text">Advanced Software Engineering, also known as field session, is a capstone course in which students apply their coursework knowledge to a challenging applied problem in computer science. Students are expected to gather requirements, create a detailed design, then implement and document their solution. Team work and communication skills are emphasized throughout the session.  </p>

      <p class="card-text"><small class="text-muted">October 4, 2019</small></p>

    </div></a>

  </div>

  <div class="card" style="margin: 40px">

    <img src="https://image.shutterstock.com/image-photo/web-development-coding-programming-internet-260nw-1141706672.jpg" class="card-img-top" alt="...">

    <a href="https://www.techopedia.com/definition/23898/web-programming"><div class="card-body">

      <h5 class="card-title">Web Programming</h5>

      <p class="card-text">Web programming refers to the writing, markup and coding involved in Web development, which includes Web content, Web client and server scripting and network security. The most common languages used for Web programming are XML, HTML, JavaScript, Perl 5 and PHP. Web programming is different from just programming, which requires interdisciplinary knowledge on the application area, client and server scripting, and database technology. </p>

      <p class="card-text"><small class="text-muted">September 06, 2019</small></p>

    </div></a>

  </div>

  <div class="card" style="margin: 40px">

    <img src="http://new.whatmobile.net/wp-content/uploads/2019/06/2-1-1.jpg" class="card-img-top" alt="...">

    <a href="https://www.python.org/about/gettingstarted/"><div class="card-body">

      <h5 class="card-title">Python Programming</h5>

      <p class="card-text">nstalling Python is generally easy, and nowadays many Linux and UNIX distributions include a recent Python. Even some Windows computers (notably those from HP) now come with Python already installed. If you do need to install Python and aren't confident about the task you can find a few notes on the BeginnersGuide/Download wiki page, but installation is unremarkable on most platforms. </p>

      <p class="card-text"><small class="text-muted">September 06, 2019</small></p>

    </div></a>

  </div>

</div>

<app-footer></app-footer>

EditProfile.component.html

<app-home></app-home>

<div class="card" >

    <br>

        <form (ngSubmit)="onProfile()" class="row" style="margin-left: 40px">

            <div class="col-sm-4 text-center">

                <div class="card">

                    <div class="card-body">

                      <img src="https://scontent.fmkc1-1.fna.fbcdn.net/v/t1.0-1/p240x240/29063485\_1659424430807667\_4337560930995883609\_n.jpg?\_nc\_cat=105&\_nc\_oc=AQlazz4tbl4rS5O5Mc-tdKPzzWz5j8pINw6XjzN3sL8kply2Rg2H5k3MMLd0bZJXbLpnFX6FpdA\_DLzfmN\_dd1zo&\_nc\_ht=scontent.fmkc1-1.fna&oh=fb52eaac1cef2ae4faf99eff8660a1c2&oe=5E2D9010" alt="">

                      <input [(ngModel)]="image" type="type" name="image" id="image" placeholder="Please enter your Image Link">

                    </div>

                </div>

            </div>

            <div class="col-sm-6">

                <div class="card">

                    <div class="profile-head emp-profile">

                        <h5>

                        <input [(ngModel)]="userId" type="text" name="userId" id="userId" placeholder="Enter Used Id">

                        </h5>

                        <h6>

                        <input [(ngModel)]="department" type="text" name="department" id="department" placeholder="Enter Department">

                        </h6>

                        <hr>

                    <div class="card-body">

                        <div class="row">

                            <div class="col-md-6">

                                <label>Name:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="name" type="text" name="name" id="name" placeholder="Please Enter Your Name">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Email ID:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="text" type="email" name="email" id="email" placeholder="Please Enter Your email">

                            </div>

                            </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Address:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="address" type="Address" name="address" id="address" placeholder="Please Enter Your Address">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Contact:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="contact" type="tel" name="contact" id="contact" placeholder="Please Enter Your Contact">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Age:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="age" type="number" name="age" id="age" placeholder="Please Enter Your Age">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Date Of Birth:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="dob" type="date" name="dob" id="dob" placeholder="Please Enter Your Date Of birth">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>University:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="university" type="text" name="university" id="university" placeholder="Please Enter Your University Name">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Major:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="major" type="text" name="major" id="major" placeholder="Please Enter Your Major">

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-md-6">

                                <label>Year Joined:</label>

                            </div>

                            <div class="col-md-6">

                                <input [(ngModel)]="yearJoined" type="year" name="yearJoined" id="yearJoined" placeholder="Please Enter Year Joined">

                            </div>

                        </div>

                    </div>

                  </div>

                </div>

            </div>

            <button type="submit" style="height: 100%;" routerLink="e">Save</button>

        </form>

    <br>

</div>

<app-footer></app-footer>

EditProfile.component.ts

import { Component, OnInit } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { Observable } from 'rxjs';

@Component({

  selector: 'app-edit-profile',

  templateUrl: './edit-profile.component.html',

  styleUrls: ['./edit-profile.component.css']

})

export class EditProfileComponent implements OnInit {

  public name : string;

  public email : string;

  public userId : string;

  public department : string;

  public contact : string;

  public address : string;

  public age: string;

  public dob : string;

  public university : string;

  public major : string;

  public yearJoined : string;

  public image: String;

  url: string;

  message:Observable<any>

  constructor(private http: HttpClient) { }

  onProfile(){

    this.url = "http://127.0.0.1:3000/profile/Fillup"

    this.http.post(this.url,{

      name: this.name,

      email: this.email,

      userId: this.userId,

      contact: this.contact,

      age: this.email,

      dob: this.dob,

      major: this.major,

      department: this.department,

      university: this.university,

      address: this.address,

      image: this.image,

      yearJoined: this.yearJoined,

    })

    .subscribe(

      (res:any)=>{

        this.message = res.message;

        alert(this.message);

        console.log(this.message);

      }

    )

  }

  ngOnInit() {

  }

}

Footer.Component.html

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<div class="footer">

    <div class="row">

            <div class="column">

                <p class="title">

                    About Me

                </p>

                <p>

                    We are of three graduate student from University of Missouri Kansas City worked together to design this project for the academic purpose.

                </p>

            </div>

            <div class="column">

                <p class="title">

                    News Letter

                </p>

                <p>

                </p>

                <input type="EMAIL" placeholder="Email Address"><button >></button>

            </div>

            <div class="column">

                <p class="title">

                    Follow Me

                </p>

                <a href="#" class="fa fa-facebook"></a>

                <a href="#" class="fa fa-twitter"></a>

                <a href="#" class="fa fa-google"></a>

                <a href="#" class="fa fa-linkedin"></a>

                <a href="#" class="fa fa-youtube"></a>

                <a href="#" class="fa fa-instagram"></a>

            </div>

    </div>

</div>

Friends.component.ts

import { Component, OnInit } from '@angular/core';

import {Observable} from "rxjs";

import {HttpClient} from "@angular/common/http";

@Component({

  selector: 'app-friends',

  templateUrl: './friends.component.html',

  styleUrls: ['./friends.component.css']

})

export class FriendsComponent implements OnInit {

  url:string;

  result:Observable<any>;

  constructor(private http: HttpClient) { }

  ngOnInit() {

    this.url = 'http://127.0.0.1:3000/friends/search'

    this.http.get(this.url).subscribe((res: any) => {

      this.result = res.data;

      console.log(this.result);

    });

  }

}

Friends.html

<app-home></app-home>

<div class="card-deck">

  <div \*ngFor="let profile of result" class="card" style="margin: 40px">

    <div class="card-body">

      <h5 class="card-title">{{profile.userId}}</h5>

      <p class="card-text"> {{profile.name}}</p>

      <p class="card-text">{{profile.email}}</p>

      <p class="card-text">{{profile.contact}}</p>

      <p class="card-text">{{profile.age}}</p>

      <p class="card-text">{{profile.university}}</p>

      <p class="card-text">{{profile.address}}</p>

    </div>

  </div>

</div>

<app-footer></app-footer>

[Home.component.html](http://Home.component.html)

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<header>

    {{title}}

</header>

<nav id="navbar">

    <ul>

    <li><a class="active" routerLink="/Home">Home</a></li>

    <li><a routerLink="/Friends" >Friends</a></li>

    <li><a routerLink="/Announcements">Announcements</a></li>

    <li><a routerLink="/VideoLessons">Video Lessons</a></li>

    <li><a routerLink="/DiscoverBooks">Discover Books</a></li>

    <li><a routerLink="/calender">Calender</a></li>

    <li class="right"><a routerLink="/SignIn">LogOut</a></li>

    <li class="right"><a routerLink="/Profile">Profile</a></li>

    <li class="right"><a routerLink="/edit\_course">Edit Course</a></li>

    </ul>

</nav>

Online-coureses.html

<app-home></app-home>

<div class="card-deck">

        <div class="card" style="margin: 40px">

          <img src="..." class="card-img-top" alt="...">

          <div class="card-body">

            <h5 class="card-title">Card title</h5>

            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

            <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

          </div>

        </div>

        <div class="card" style="margin: 40px">

          <img src="..." class="card-img-top" alt="...">

          <div class="card-body">

            <h5 class="card-title">Card title</h5>

            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

            <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

          </div>

        </div>

        <div class="card" style="margin: 40px">

          <img src="..." class="card-img-top" alt="...">

          <div class="card-body">

            <h5 class="card-title">Card title</h5>

            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

            <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

          </div>

        </div>

</div>

<div class="card-deck">

        <div class="card" style="margin: 40px">

          <img src="..." class="card-img-top" alt="...">

          <div class="card-body">

            <h5 class="card-title">Card title</h5>

            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

            <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

          </div>

        </div>

        <div class="card" style="margin: 40px">

          <img src="..." class="card-img-top" alt="...">

          <div class="card-body">

            <h5 class="card-title">Card title</h5>

            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

            <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

          </div>

        </div>

        <div class="card" style="margin: 40px">

          <img src="..." class="card-img-top" alt="...">

          <div class="card-body">

            <h5 class="card-title">Card title</h5>

            <p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

            <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

          </div>

        </div>

    </div>

    <app-footer></app-footer>

Profile.component.html

<app-home></app-home>

<div class="card" >

    <br>

        <div class="row" style="margin-left: 40px">

            <div class="col-sm-4 text-center">

                <div class="card">

                    <div class="card-body">

                      <img src="https://scontent.fmkc1-1.fna.fbcdn.net/v/t1.0-1/p240x240/29063485\_1659424430807667\_4337560930995883609\_n.jpg?\_nc\_cat=105&\_nc\_oc=AQlazz4tbl4rS5O5Mc-tdKPzzWz5j8pINw6XjzN3sL8kply2Rg2H5k3MMLd0bZJXbLpnFX6FpdA\_DLzfmN\_dd1zo&\_nc\_ht=scontent.fmkc1-1.fna&oh=fb52eaac1cef2ae4faf99eff8660a1c2&oe=5E2D9010" alt="">

                    </div>

                </div>

            </div>

            <div class="col-sm-6">

                <div class="card">

                    <div class="profile-head emp-profile">

                        <h5>

                        RSaiSampathKumar

                        </h5>

                        <h6>

                        Computer Science Student

                        </h6>

                        <hr>

                    <div class="card-body">

                        <div class="row">

                            <div class="col-md-6">

                                <label>Name:</label>

                                </div>

                                    <div class="col-md-6">

                                        <p>Raigiri Sai Sampath Kumar</p>

                                    </div>

                        </div>

                                <div class="row">

                                        <div class="col-md-6">

                                            <label>Email:</label>

                                        </div>

                                        <div class="col-md-6">

                                            <p>SaiSampathKumar@gmail.com</p>

                                        </div>

                                    </div>

                                    <div class="row">

                                            <div class="col-md-6">

                                                <label>Age: </label>

                                            </div>

                                            <div class="col-md-6">

                                                <p>24</p>

                                            </div>

                                        </div>

                                        <div class="row">

                                                <div class="col-md-6">

                                                    <label>Address:</label>

                                                </div>

                                                <div class="col-md-6">

                                                    <p>5056 Baltimore Avenue</p>

                                                </div>

                                            </div>

                                            <div class="row">

                                                    <div class="col-md-6">

                                                        <label>Contact</label>

                                                    </div>

                                                    <div class="col-md-6">

                                                        <p>9178888888</p>

                                                    </div>

                                                </div>

                                                <div class="row">

                                                    <div class="col-md-6">

                                                        <label>University:</label>

                                                    </div>

                                                    <div class="col-md-6">

                                                        <p>UMKC</p>

                                                    </div>

                                                </div>

                                                <div class="row">

                                                    <div class="col-md-6">

                                                        <label>Year Joinde:</label>

                                                    </div>

                                                    <div class="col-md-6">

                                                        <p>Fall 2018</p>

                                                    </div>

                                                </div>

                    </div>

                  </div>

                </div>

            </div>

            <button style="height: 100%; " routerLink="/editProfile">Edit Profile</button>

        </div>

    <br>

</div>

<app-footer></app-footer>

SignIn.component.html

<body>

    <header>

            <br><br><br><br><br>

        Welcome to Student World

    </header>

    <div class="sign-up-page">

        <img class="signin-logo-img" src="https://signup.com/mobileweb/2.0/images/build/mobile-organizers.png" alt="Sign Up">

        <h1> SIGN IN</h1>

        <form name="sign" (ngSubmit)="onSignIn()">

            <input [(ngModel)]="name" name="name" type="text" class="input-box" placeholder="UserName">

            <!-- <p>{{name}}</p> -->

            <input [(ngModel)]="password" name="Password" type="Password" class="input-box" placeholder="Your Password">

            <button type="submit" class="signup-btn"> Sign In </button>

            <hr>

            <p class="or">OR</p>

            <button type="button" class="facebook-btn"> Login with Facebook </button>

            <p> Do you have an account? <a routerLink="/SignUp">Sign Up</a></p>

        </form>

    </div>

</body>

SignIn.component.ts

import { Component, OnInit } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { Observable } from 'rxjs';

import {Router} from '@angular/router';

@Component({

  selector: 'app-sign-in',

  templateUrl: './sign-in.component.html',

  styleUrls: ['./sign-in.component.css']

})

export class SignInComponent implements OnInit {

  public name: string;

  public pass: string;

  url:string;

  result:Observable<any>;

  constructor(private http: HttpClient,private router: Router) { }

  onSignIn(){

    // if(this.pass != null && this.name != null)

    // {

      console.log(this.name);

      this.url = "http://127.0.0.1:3000/signin?searchtext="+this.name;

      console.log(this.url);

      this.http.get(this.url).subscribe((res:any)=>{

        this.result = res.data;

        if(this.result)

        {

          if(this.pass == res.data.password)

          {

            this.router.navigate(['/Home']);

          }

        }

        console.log(this.result);

      })

    // }

  //   else{

  //     alert("Incorrect Username or Password Please try again Later......!")

  //   }

  }

  ngOnInit() {

  }

  }

SignUp.component.html

<body >

    <header>

        <br><br><br><br><br><br><br>

        Welcome To Student World

    </header>

    <div class="sign-up-page">

        <img class="signin-logo-img" src="https://signup.com/mobileweb/2.0/images/build/mobile-organizers.png" alt="Sign Up">

        <h1> SIGN UP NOW</h1>

        <form (ngSubmit)="onRegister()">

            <input [(ngModel)]="firstName" name="firstName" class="input-box" placeholder="First Name">

            <input [(ngModel)]="lastName" name="lastName" class="input-box" placeholder="Last Name">

            <input [(ngModel)]="email" name="email" type="Email" class="input-box" placeholder="Your Email">

            <input [(ngModel)]="Password" name="Password" type="Password" class="input-box" placeholder="Your Password">

            <input [(ngModel)]="Password1" name="Password1" type="Password" class="input-box" placeholder="Re-enter Password">

            <p><span><input type="checkbox"></span> I agree to the terms and conditions </p>

            <button type="submit" class="signup-btn"> Sign Up </button>

            <hr>

            <p class="or">OR</p>

            <button type="button" class="facebook-btn"> Sign up with Facebook </button>

            <p> Do you have an account? <a routerLink ="/SignIn"> Sign in</a></p>

        </form>

    </div>

</body>

SignUp.component.ts

import { Component, OnInit } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { Observable } from 'rxjs';

import { Router } from '@angular/router';

@Component({

  selector: 'app-signup',

  templateUrl: './signup.component.html',

  styleUrls: ['./signup.component.css']

})

export class SignupComponent implements OnInit {

  public firstName : string;

  public lastName : string;

  public Password : string;

  public Password1 : string;

  public email: string;

  url: string;

  message:Observable<any>

  constructor(private http: HttpClient, private router: Router) { }

  onRegister(){

    if(this.Password == this.Password1){

    if(this.lastName != null && this.firstName != null && this.Password != null && this.Password1!=null && this.email!=null)

    {

    this.url = "http://127.0.0.1:3000/signup/register"

    this.http.post(this.url,{

      name: this.firstName +" "+ this.lastName ,

      email: this.email,

      password: this.Password,

    })

    .subscribe(

      (res:any)=>{

        this.message = res.message;

        alert(this.message);

        this.router.navigate(['/Home']);

        console.log(this.message);

      }

    )

    }

    else{

      alert("Please fill the form correctly");

    }

  }

  else{

    alert("Passwords are not matching");

  }

  }

  ngOnInit(){

  }

}

Video-Lesson.component.html

<app-home></app-home>

<div class="card-deck">

    <div class="card" style="margin: 40px">

      <iframe class="card-img-top" width="853" height="355" src="https://www.youtube.com/embed/tyDKR4FG3Yw" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

      <div class="card-body">

        <h5 class="card-title">Discrete Mathematics</h5>

        <p class="card-text">Today we introduce set theory, elements, and how to build sets.</p>

        <p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

      </div>

    </div>

    <div class="card" style="margin: 40px">

      <iframe class="card-img-top" width="853" height="355" src="https://www.youtube.com/embed/DuDz6B4cqVc" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

      <div class="card-body">

        <h5 class="card-title">Data Structures</h5>

        <p class="card-text">Today we’re going to talk about on how we organize the data we use on our devices. You might remember last episode we walked through some sorting algorithms, but skipped over how the information actually got there in the first place! And it is this ability to store and access information in a structured and meaningful way that is crucial to programming. From strings, pointers, and nodes, to heaps, trees, and stacks get ready for an ARRAY of new terminology and concepts.

          </p>

        <p class="card-text"><small class="text-muted">Last updated 17 mins ago</small></p>

      </div>

    </div>

</div>

<div class="card-deck">

    <div class="card" style="margin: 40px">

      <iframe class="card-img-top" width="853" height="355" src="https://www.youtube.com/embed/26QPDBe-NB8" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

      <div class="card-body">

        <h5 class="card-title">Operating Systems</h5>

        <p class="card-text">computers keep getting faster and faster, and by the start of the 1950s they had gotten so fast that it often took longer to manually load programs via punch cards than to actually run them! The solution was the operating system (or OS), which is just a program with special privileges that allows it to run and manage other programs. So today, we’re going to trace the development of operating systems from the Multics and Atlas Supervisor to Unix and MS-DOS, and take at look at how these systems heavily influenced popular OSes like Linux, Windows, MacOS, and Android that we use today.</p>

        <p class="card-text"><small class="text-muted">Last updated 12 mins ago</small></p>

      </div>

    </div>

    <div class="card" style="margin: 40px">

      <iframe class="card-img-top" width="853" height="355" src="https://www.youtube.com/embed/X3paOmcrTjQ" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

      <div class="card-body">

        <h5 class="card-title">Data Science</h5>

        <p class="card-text">This Data Science tutorial video will give you an idea on the life of a Data Scientist, steps involved in Data science project, roles & salary offered to a Data Scientist. Data is everywhere. In fact, the amount of digital data that exists is growing at a rapid rate, doubling every two years, and changing the way we live. Data Science is basically dealing with unstructured and structured data. Data Science is a field that comprises of everything that is related to data cleansing, preparation, and analysis. In simple terms, it is the umbrella of techniques used when trying to extract insights and information from data. Now, let us get started and understand what is Data Science all about.

          </p>

        <p class="card-text"><small class="text-muted">Last updated 17 mins ago</small></p>

      </div>

    </div>

</div>

<div class="card-deck">

    <div class="card" style="margin: 40px">

        <iframe class="card-img-top" width="853" height="355" src="https://www.youtube.com/embed/sB2iQSvrcG0" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

      <div class="card-body">

        <h5 class="card-title">Software Engineering Basics</h5>

        <p class="card-text">In university and colleges, software engineering can be a large part of the learning process. Today, we take a look at just why so much emphasis is placed on software engineering in schools, and what are some of the key learning points in it. Software engineering is all about writing code in a structured and formal way, with the end goal of achieving greater readability and maintainability of code.</p>

        <p class="card-text"><small class="text-muted">Last updated 12 mins ago</small></p>

      </div>

    </div>

    <div class="card" style="margin: 40px">

      <iframe class="card-img-top" width="853" height="355" src="https://www.youtube.com/embed/X3paOmcrTjQ" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

      <div class="card-body">

        <h5 class="card-title">Data Science</h5>

        <p class="card-text">This Data Science tutorial video will give you an idea on the life of a Data Scientist, steps involved in Data science project, roles & salary offered to a Data Scientist. Data is everywhere. In fact, the amount of digital data that exists is growing at a rapid rate, doubling every two years, and changing the way we live. Data Science is basically dealing with unstructured and structured data. Data Science is a field that comprises of everything that is related to data cleansing, preparation, and analysis. In simple terms, it is the umbrella of techniques used when trying to extract insights and information from data. Now, let us get started and understand what is Data Science all about.

          </p>

        <p class="card-text"><small class="text-muted">Last updated 17 mins ago</small></p>

      </div>

    </div>

</div>

<app-footer></app-footer>