**STUDENT ATTENDANCE MANAGEMENT SYSTEM**

**PROJECT REPORT**

**Submitted to :**

**Dr. NAGARAJU M K. (Sr. Professor)**

in partial fulfillment for the award

of the

**B. Tech**

degree in

**Computer Science and Engineering**

**School of computer science and engineering**

We hereby declare that the project entitled “**STUDENT** **ATTENDANCE MANAGEMENT SYSTEM**”submitted by us tothe School of Computer Science and Engineering, VIT University, Vellore in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out

by us under the supervision of Dr.**NAGARAJU M** we further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

**School of Computer Science and Engineering**

**CERTIFICATE**

The project report entitled “**STUDENT ATTENDANCE** **MANAGEMENT SYSTEM**” is prepared and submitted by ourgroup members . It has been found satisfactory in terms of scope, quality and presentation as partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering in VIT University, India.

**Guide**

**PROF.NAGARAAJU M.**

ACKNOWLEDGEMENT

I would like to express my gratitude to all those who have helped me in the successful completion of this project. Without their support, I would not have been able to achieve the goal of the project successfully.

I would like to take this opportunity to thank my guide, DR.NAGARAJU M, for his constant support, guidance and mentorship without which it would have been really difficult to complete the project on time.

I would like to thank our Dean, Dr. Arunkumar T., who provided us with the facilities required and conducive conditions for the project.

Finally, I would like to express my sincere gratitude to VIT University, which provided me with a platform to hone my skills over a period of four years.

**Abstract:**

Attendance Management System is software developed for daily student attendance in schools, colleges and institutes. If facilitates to access the attendance information of a particular student in a particular class. The information is sorted by the operators, which will be provided by the teacher for a particular class. This system will also help in evaluating attendance eligibility criteria of a student.

The intention of developing Attendance Management System is to computerized the tradition way of taking attendance. Another purpose for developing this software is to generate the desired reports automatically at the end of the session or in the between of the session as they require.

This project is basically a desktop application which means self contained software runs on the system on which it has been installed under the user control and it will work for a particular institute or college only.

**TABLE OF**

**KEYWORDS**

**ER-DIAGRAMS ..............................................................**

DATABASE TABLE ..........................................................................

SHCEMA DIAGRAM ........................................................................

DEPLOYEMENT PLANS……………………………………………………………………

SOURCE CODE………………………………………………………………………………

FRONT END……………………………………………………………………………………

BACKEND……………………………………………………………………………………..

IMPLEMENTATION …………………………………………………………………….

**TABLE:**

**ER-DIAGRAM:**

**DEPLOYEMENT PLANS:**

Attendance

system

|  |  |  |  |
| --- | --- | --- | --- |
| Database | Backend | Frontend |  |
| (Node.js) |  |
|  |  |  |

**PROJECT DIVIDED INTO 3 MAIN PARTS:**

*  **The database**
*  **Backend(nodejs)**
*  **Frontend(HTML,css and javascript)**

**DEPLOYEMENT PLANS:**

**2**

**CONNECTION OF SERVER:**

*  Server name: attendance17.database.windows.net
*  User name:technophilic
*  Server name:Attendance

**3**

**WORKING PRINCIPLE:**

User interacts with the frontend (or 'the website') this creates an event (say fetch attendance for a particular class) and the frontend sends a query to the backend as a response to the created event

Backend receives the request from the frontend and parses its data to understand what the request is

Then it sends a SQL query to the database (something like 'SELECT \* FROM ATTENDANCE WHERE NAME="VINEETH" ')

Database responds with an appropriate response (this is the table which we see in terminal)

This response is sent to backend, now we format it properly apply filters according to the request and send a JSON object to the frontend

Front-end reads the data from response and displays it in the website

**4**

**SERVER CODE:**

**FRONT END:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="apple-touch-icon" sizes="76x76" href="../public/img/apple-icon.png">

<link rel="icon" type="image/png" href="../public/img/favicon.png"> <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" /> <title>Login Page - Now Ui Kit by Creative Tim</title>

<meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0, shrink-to-fit=no' name='viewport' />

<!-- Fonts and icons -->

<link

href="https://fonts.googleapis.com/css?family=Montserrat:400,700,200" rel="stylesheet" />

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/latest/css/font-awesome.min.css" />

<!-- CSS Files -->

<link href="../public/css/bootstrap.min.css" rel="stylesheet" /> <link href="../public/css/now-ui-kit.css?v=1.1.0" rel="stylesheet" /> <!-- CSS Just for demo purpose, don't include it in your project --> <link href="../public/css/demo.css" rel="stylesheet" />

</head>

<body class="login-page sidebar-collapse">

<div class="page-header" filter-color="orange">

<div class="page-header-image" style="background-image:url(../public/img/login.jpg)"></div>

<div class="container">

<div class="col-md-4 content-center">

<div class="card card-login card-plain">

**5**

<form class="form" method="" action="">

<div class="header header-primary text-center"> <h1>Staff Login</h1>

</div>

<div class="content">

<div class="input-group form-group-no-border input-lg">

<span class="input-group-addon">

<i class="now-ui-icons users\_circle-08"></i>

</span>

<input type="text" class="form-control"placeholder="Employee id">

</div>

<div class="input-group form-group-no-border input-lg">

<span class="input-group-addon">

<i class="now-ui-icons objects\_key-25"></i>

</span>

<input type="password" placeholder="Password" class="form control"/>

</div> </div>

<div class="footer text-center">

<a href="/home" class="btn btn-primary btn-round btn-lg btn-block">Login </a>

</div>

<div class="pull-left"> <h6>

<a href="#pablo" class="link">Create Account</a>

</h6> </div>

<div class="pull-right"> <h6>

<a href="#pablo" class="link">Need Help?</a>

</h6> </div>

</form> </div>

</div>

**6**

</div> </div>

</body>

<!-- Core JS Files -->

<script src="../public/js/core/jquery.3.2.1.min.js" type="text/javascript"></script>

<script src="../public/js/core/popper.min.js" type="text/javascript"></script>

<script src="../public/js/core/bootstrap.min.js" type="text/javascript"></script>

<!-- Plugin for Switches, full documentation here: http://www.jque.re/plugins/version3/bootstrap.switch/ --> <script src="../public/js/plugins/bootstrap-switch.js"></script> <!-- Plugin for the Sliders, full documentation here: http://refreshless.com/nouislider/ -->

<script src="../public/js/plugins/nouislider.min.js" type="text/javascript"></script>

<!-- Plugin for the DatePicker, full documentation here: https://github.com/uxsolutions/bootstrap-datepicker --> <script src="../public/js/plugins/bootstrap-datepicker.js" type="text/javascript"></script>

<!-- Control Center for Now Ui Kit: parallax effects, scripts for the example pages etc -->

<script src="../public/js/now-ui-kit.js?v=1.1.0" type="text/javascript"></script> </html>

let express=require('express');

**7**

**JAVA SCRIPT**

let cors=require('cors');

let bodyParser=require('body-parser'); let app=express();

let path=require('path');

let passport=require('passport');

let LocalStrategy=require('passport-local').Strategy; let session=require('expres session');

app.use(cors()); app.use(body Parser.json());

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

app.set('views', path.join(\_\_dirname, 'views')); app.set('view engine', 'ejs'); app.use(session({'secret':'thisislameisthislame?'})); app.use(passport.initialize()); app.use(passport.session());

app.get('/',function (req, res) { res.render('index.ejs');

});

app.get('/attendsel',function (req, res) { res.render('attendance\_sel.ejs');

});

app.get('/attendance',function (req, res) { res.render('attendance.ejs');

});

app.get('/home',function (req, res) { res.render('dashboard.ejs');

}); app.post('/signup',function(req,res){

console.log('post request recieved !');

let username=req.body.username;

});

passport.serializeUser(function(user, done) { done(null, user.id); });

**8**

// used to deserialize the user passport.deserializeUser(function(id, done) {

User.findById(id, function(err, user)

{ done(err, user);

});

});

passport.use('local-login', new LocalStrategy({

// by default, local strategy uses username and password, we will override with email

// usernameField : 'email',

// passwordField : 'password', passReqToCallback : true // allows us to pass

back the entire request to the callback },

function(req, username, password, done) { // callback with email and password from our form

* // find a user whose email is the same as the forms email
* // we are checking to see if the user trying to login already exists

User.findOne({username}, function(err, user) {

* + // if there are any errors, return the error

before anything else if (err)

return done(err);

// if no user is found, return the message if (!user)

return done(null, false, req.flash('loginMessage', 'No user found.')); // req.flash is the way to set flashdata using connect-flash

// if the user is found but the password is

wrong

if (!user.validPassword(password))

**9**

return done(null, false, req.flash('loginMessage', 'Oops! Wrong password.')); // create the loginMessage and save it to session as flashdata

// all is well, return successful user return done(null, user);

});

}

));

app.post('/login', passport.authenticate('local-login', { successRedirect : '/home', // redirect to the secure

profile section

failureRedirect : '/login', // redirect back to the signup page if there is an error

failureFlash : true // allow flash messages }));

let port=process.env.PORT||'3000'; app.listen(port,function () {

if(port==='3000')

console.log('server started at port 3000 !'); Else

console.log('prod server started !');

});

**10**

**HOME SCREEN (USER INTERFACE)**

**DASH BOARD:**

**ACADEMIC INTERFACE:**

**11**

**12**

CONCLUSION

We made this project with aim of making an interface for updating faculty attendance online . We have used node.js,html,css and javascript to frame it.Its full fledged working website just as our college portal-VTOP BETA.

**13**