Pune Institute of Computer Technology
Dhankawadi, Pune

A MINI PROJECT REPORT
ON

Internship Search Web Application

SUBMITTED BY

Student Name: Rishi Rewadkar	**Roll No:31449**
Student Name: Nikita Renuse	**Roll No:31448**
Student Name: Yash Rode	**Roll No:31450**
Student Name: Sanam Palsule	**Roll No:31451**

Under the guidance of
Prof. A. A. Jewalikar

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year 2019-20

DEPARTMENT OF COMPUTER ENGINEERING

Pune Institute of Computer Technology

Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the mini project entitled

"Internship Search Web Application"

Submitted by

Rishi Rewadkar	Roll No. 31449
Nikita Renuse	Roll No. 31448
Yash Rode	Roll No. 31450
Sanam Palsule	Roll No. 31451

is a bonafide work carried out by students under the guidance of Prof. A. A. Jewalikar and is submitted towards the partial fulfilment of the requirement of Third Year Computer Engineering Semester II of Savitribai Phule Pune University.

Prof. A. A. Jewalikar
Internal Guide

Prof. M.S.Takalikar
Head
Department of Computer Engineering

Place:
Date:

ACKNOWLEDGEMENT

I sincerely thank our WTL Coordinator Prof. A. A. Jewalikar and Head of Department Prof. M.S.Takalikar for their support. I also sincerely convey my gratitude to my guide Prof. A. A. Jewalikar, Department of Computer Engineering for her constant support, providing all the help, motivation and encouragement from beginning till end.

Contents

1 Project Idea and Functional Requirements

1.1 Project Idea

All engineering students want to gain industrial experience and gain hands-on experience to work on live projects during the semester . Industry people are not always able to find the right candidates with required skill sets for the internship . This application creates a platform for students and industry personnel to interact with industry people can directly demand for the candidates with required skill sets for their project and interested candidates with skill sets can apply for internship .Student information like name , CGPA , Grades etc are filled by college that way industry people can make fair judgement about which candidate to hire for internship as the information is validated by College. This way we can filter students according to CGPA ,Branch ,or Year as per the requirement of the project . These shortlisted students are interviewed and further selection process is notified through this platform making sure not much of time is wasted and deserving candidates get the opportunity.benefiting both student and industry people for finding the right match.

1.2 Functional Requirements

1.2.1 STUDENT

STUDENT LOGIN : Student Authentication is provided and valid students enter the student home page. JWT(JSON WEB TOKEN) are validated from the browser cache to check if a user has already logged in . After the token is expired the user is logged out of the system and needs to login again .

STUDENT HOME PAGE: Student Information is displayed provided by Admin . The dashboard contains options to view Internship which he/she is eligible to apply for,to upload/update Resume and to get notification on being shortlisted after applying for internship.

1.2.2 COMPANY

Company Login: Company Authentication is provided and valid users enter the Company home page. JWT(JSON WEB TOKEN) are validated from the browser cache to check if a user has already logged in . After the token is expired the user is logged out of the system and needs to login again .

Company Home Page: Company Home page displays the information provided by the admin and also the dashboard contains options to Create a new Internship and set criteria to apply for internship,shortlisted students based on resume for internship and send messages to shortlisted students.

Company Register: New Company needs to register if it's not already known to the organisation . Admin check for the background based on the information provided during registration . Admin can add a company to its database creating a new company providing username and password .

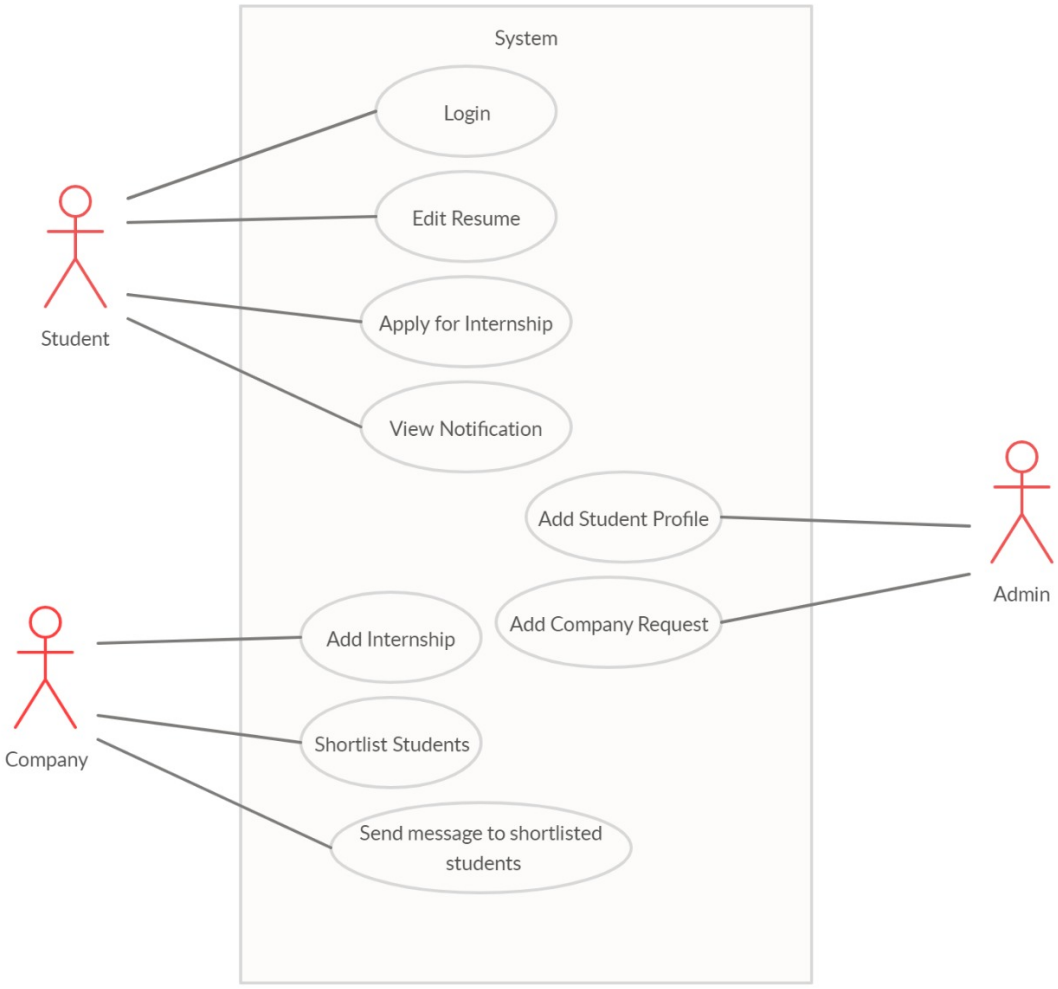
1.2.3 ADMIN

Admin Login: Admin Authentication is provided and valid users enter the Admin home page. JWT(JSON WEB TOKEN) are validated from the browser cache to check if a user has already logged in . After the token is expired the user is logged out of the system and needs to login again .

Admin Home Page: Admin home pages contain options to view and accept or decline new company requests allowing new company to be part of the system . Admin create Student users and provide them with username and password along with academic performance . Act as an intermediate person between a student and company .

2 Design

2.1 Use Case Diagram



2.2 Database Schema

COMPANY SCHEMA:

```
var schema = new Schema({
companyemail : {type:String, require:true},
companyname:{type:String,require:true},
companyurl:{type:String,require:true},
companycontact:{type:String,require:true},
 password:{type:String, require:true},
});
```

INTERNSHIP SCHEMA:

```
var schema = new Schema({
companyname:{type:String,require:true},
aboutinternship:{type:String,require:true},
durationofinternship:{type:String,require:true},
startinternship:{type:Date,required:true},
applybefore:{type:Date,required:true},
internshipemail:{type:String,required:true},
FE:{type:Boolean,required:false},
SE:{type:Boolean,required:false},
TE:{type:Boolean,required:false},
BE:{type:Boolean,required:false},
COMP:{type:Boolean,required:false},
IT:{type:Boolean,required:false},
ENTC:{type:Boolean,required:false},
CGPA:{type:Number,required:true}});
```

STUDENT SCHEMA:

```
var schema = new Schema({
studentname:{type:String,require:true},
studentid:{type:String,require:true},
studentCGPA:{type:String,require:true},
password:{type:String, require:true},
FE:{type:Boolean,required:false},
SE:{type:Boolean,required:false},
TE:{type:Boolean,required:false},
BE:{type:Boolean,required:false},
COMP:{type:Boolean,required:false},
IT:{type:Boolean,required:false},
ENTC:{type:Boolean,required:false}});
```

3 SOURCE CODE AND SCREENSHOTS

3.1 Angular 8

Companyhome.component.ts

```
import { Component, OnInit } from '@angular/core';
import { MyserviceService } from '../../myservice.service';
import { Router ,ActivatedRoute } from '@angular/router';
import { NgForm,FormGroup,FormControl, Validators } from '@angular/forms';
@Component({
selector: 'app-companyhome',
templateUrl: './companyhome.component.html',
styleUrls: ['./companyhome.component.css']
})
export class CompanyhomeComponent implements OnInit {

successmessage='';
companyname='';
uploadedinternship;
studentlist;
listofapplied;
arrayofstu;
myIForm:FormGroup;
finalmessage:FormGroup;
constructor(private _myservice:MyserviceService,
private _router: Router,private _activatedRoute:ActivatedRoute) {

this.myIForm = new FormGroup({
companyname:new FormControl(null,Validators.required),
aboutinternship:new FormControl(null,Validators.required),
durationofinternship:new FormControl(null,Validators.required),
startinternship:new FormControl(null,Validators.required),
applybefore:new FormControl(null,Validators.required),
internshipemail: new FormControl(null,Validators.email),
FE: new FormControl(false,Validators.required),
SE: new FormControl(false,Validators.required),
TE: new FormControl(false,Validators.required),
BE: new FormControl(false,Validators.required),
COMP: new FormControl(false,Validators.required),
ENTC: new FormControl(false,Validators.required),
IT: new FormControl(false,Validators.required),
});

this.finalmessage=new FormGroup({
msg:new FormControl(null,Validators.required)
});

this._myservice.getCompanyName()
.subscribe(
data => this.companyname= data.toString(),
error => {this._router.navigate(['../companylogin'])});
```

```
}

ngOnInit() {
}
addnew()
{
document.getElementById("one").style.display="block";
document.getElementById("two").style.display="none";
document.getElementById("three").style.display="none";
}
viewuploaded()
{
document.getElementById("one").style.display="none";
document.getElementById("two").style.display="block";
document.getElementById("three").style.display="none";
console.log("REady");

this._myservice.getinternship()
.subscribe(
data => this.uploadedinternship=data,
error => {this._router.navigate(['../Company'])});
}

shortlist()
{
document.getElementById("one").style.display="none";
document.getElementById("two").style.display="none";
document.getElementById("three").style.display="block";
this._myservice.shortlist(this.companyname).subscribe(
data=> {this.arrayofstu=data;},error => console.log("Unable TO Fetch"));

}
logout(){
localStorage.removeItem('token');
this._router.navigate(['../Companylogin']);
}
addinternship(){
console.log(this.myIForm.value);
this._myservice.addinternship(this.myIForm.value).subscribe(
data=> this.successmessage='Registration Success',
error=> this.successmessage='Some Error'
);
}
deleteinternship(obj){
console.log(obj);
this._myservice.deleteinternship(obj).subscribe(
data=> this.successmessage='Successfully Deleted',
error=> this.successmessage='Some Error'
);
this.viewuploaded();
}
```

```
sendstudentmessage(){
console.log(this.finalmessage.value);
var obj={"companyname":this.companyname,"msg":this.finalmessage.value.msg};
this._myservice.sendmsg(obj).subscribe(
data=> this.successmessage='Successfully Deleted',
error=> this.successmessage='Some Error'
);
}

selected(studentid){
var studentobj={"studentid":studentid};
this._myservice.selectedstudent(studentobj).subscribe(
data=>this.successmessage="DOne Happy",
error => this.successmessage="Not Happy"
)
}
reject(studentid)
{
var studentobj={"studentid":studentid};
this._myservice.rejectedstudent(studentobj).subscribe(
data=>this.successmessage="Done Happy",
error => this.successmessage="Not Happy"
)
this.shortlist();
}
}
```

companyhomecomponent.html

```
<div class="jumbotron container" align="center">
<div class="row">
<div class="col-sm-7">
<h1>Company Name: {{companyname}}</h1>
</div>
<div class="col-sm-5">
<h1>COMPANY LOGO</h1>
</div>
</div>
</div>
<hr>

<div class="container">
<h1>OFFER INTERNSHIP</h1>
<button type="button" class="btn btn-outline-secondary" style="margin-right: 30px;"
(click)="addnew()">ADD NEW Internship</button>
<button type="button" class="btn btn-outline-secondary" style="margin-right: 30px;"
(click)="viewuploaded()">View Uploaded Internship</button>
<button type="button" class="btn btn-outline-secondary" (click)="shortlist()"
style="margin-right: 30px;">Short-List Students</button>
```

```
<button type="button" class="btn btn-outline-secondary"
(click)="logout()">LOGOUT</button>
<hr>
</div>
<div class="container" id="one">
<form [formGroup]="myIForm">
<div class="form-group">
<label for="description">About Internship:</label>
<textarea class="form-control" rows="7" formControlName="aboutinternship"
id="comment" name="text" placeholder="Description About the Internship"></textarea>
</div>
<p>OR</p>

<div class="custom-file mb-3">
<input type="file" class="custom-file-input" id="customFile" name="filename">
<label class="custom-file-label" for="customFile">About Internship Upload File</label>
</div>

<div class="form-group">
<label for="sel1">Duration of Internship :</label>
<select class="form-control" formControlName="durationofinternship" id="sel1">
<option>1 Month</option>
<option>2 Month</option>
<option>3 Month</option>
</select>
</div>

<div class="form-group row">
<label for="example-date-input" class="col-2 col-form-label">Starting from :</label>
<div class="col-3">
<input class="form-control" formControlName="startinternship" type="date" id="date-
input">
</div>
</div>

<div class="form-group row">
<label for="example-datetime-local-input" class="col-2 col-form-label">Apply
Before</label>
<div class="col-3">
<input class="form-control" formControlName="applybefore" type="date" id="date-
input">
</div>
</div>

<label for="demo">For further query add your email address here :</label>
<div class="input-group mb-3">
<input type="text" class="form-control" formControlName="internshipemail"
placeholder="Email Address" id="demo" name="email">
<div class="input-group-append">
<span class="input-group-text">@gmail.com</span>
</div>
```

```
</div>
<!-- hidden form passing company name -->
<input type="hidden" [(ngModel)]="companyname" id="cn"
formControlName="companyname">

<!-- pass?? -->

<hr>
<h2> This Internship Is for :</h2>
<div style="padding: 10px;">
<h4>Student In : </h4>
<div class="form-check">
<label class="form-check-label" for="check1">
<input type="checkbox" class="form-check-input" formControlName="FE" id="check1"
name="option1" value="present">First Year Engineering
</label>
</div>
<div class="form-check">
<label class="form-check-label" for="check1">
<input type="checkbox" class="form-check-input" id="check1" formControlName="SE"
name="option1" value="present">Second Year Engineering
</label>
</div>
<div class="form-check">
<label class="form-check-label" for="check1">
<input type="checkbox" class="form-check-input" id="check1" formControlName="TE"
name="option1" value="present">Third Year Engineering
</label>
</div>
<div class="form-check">
<label class="form-check-label" for="check1">
<input type="checkbox" class="form-check-input" id="check1" name="option1"
formControlName="BE" value="present">Final Year Engineering
</label>
</div>
</div>


<div style="padding: 10px;">
<h4>Department : </h4>
<div class="form-check">
<label class="form-check-label" for="check1">
<input type="checkbox" class="form-check-input" id="check1"
formControlName="COMP" name="option1" value="present">Computer Science
</label>
</div>
<div class="form-check">
<label class="form-check-label" for="check1">
<input type="checkbox" class="form-check-input" id="check1" name="option1"
formControlName="IT" value="present">Information Technology
</label>
```

```
</div>
<div class="form-check">
<label class="form-check-label" for="check1">
<input type="checkbox" class="form-check-input" id="check1" name="option1"
formControlName="ENTC" value="present">Electronics and Telecommunication
Engineering
</label>
</div>
</div>

<div style="padding: 10px;">
<h4>Academic grade minimum :</h4>
<div class="form-group row">
<label for="example-number-input" class="col-2 col-form-label">CGPA:</label>
<div class="col-2">
<input class="form-control" type="number" value="7" min="0" max="10"
formControlName="CGPA" step="0.1" id="example-number-input">
</div>
</div>
</div>
<button type="submit" style="width: 40%;margin-right: 50px;margin-left:
30px;"class="btn btn-success" (click)="addinternship()">POST</button>
<button type="submit" style="width: 40%;margin-left: 50px;"class="btn btn-dark">RESET
all Inputs</button>
<hr>
</form>
</div>
<div class="container" id="two" style="display: none;">
<h2>UPLOADED INTERNSHIP</h2>
<div id="uploaded" *ngFor ="let upint of uploadedinternship">
<p>Description:{{upint.aboutinternship}}</p>
<p>Duration:{{upint.durationofinternship}}</p>
<p>Start: {{upint.startinternship}} _ APPLY_BEFORE:{{upint.applybefore}}</p>
<p>Email Address: {{upint.internshipemail}}</p>
<button type="button" class="btn btn-outline-danger"
(click)="deleteinternship(upint)">REMOVE UPLODED INTERNSHIP</button>
</div>
</div>
<div class="container" id="three" style="display: none;" >
<h2>APPLIED CANDIDATES</h2>
<div class="card" *ngFor="let stu of arrayofstu">
<div class="card-header">
<p>Name OF Student: {{stu.studentname}} </p>
<p>STUDENT ID:{{stu.studentid}} YEAR:__</p>
</div>
<div class="card-body">
<p>SCGPA :{{stu.studentCGPA}}</p>
<p>VIEW RESUME :</p>
</div>
<div class="card-footer">
```

```
<button type="button" class="btn btn-outline-danger"
(click)="reject(stu.studentid)">REJECT/REMOVE</button>
<button type="button" class="btn btn-outline-success" style="float: right;"
(click)="selected(stu.studentid)">SHORTLIST</button>
</div>
</div>
<hr>
<h2>Send Message To Shortlisted Students</h2>
<form [formGroup]="finalmessage">
<div class="form-group">
<label for="comment">for further process:</label>
<input class="form-control" formControlName="msg" id="comment" name="text"
placeholder="WHAT NEXT ???">
</div>
<button type="submit" class="btn btn-primary" (click)="sendstudentmessage()">SEND
MESSAGE</button>
</form>
<hr>
</div>
```

companylogincomponent.ts

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { NgForm,FormGroup,FormControl, Validators } from '@angular/forms';
import {MyserviceService} from '../../myservice.service';
import { Router, ActivatedRoute } from '@angular/router';

@Component({
selector: 'app-companylogin',
templateUrl: './companylogin.component.html',
styleUrls: ['./companylogin.component.css']
})
export class CompanyloginComponent implements OnInit {
loginForm:FormGroup;
constructor(private _myservice:MyserviceService,private _router:Router,private
_activatedRoute:ActivatedRoute) {
this.loginForm=new FormGroup({
companyemail: new FormControl(null,Validators.required),
password:new FormControl(null,Validators.required)
});
}

ngOnInit() {
}
Clogin(){
console.log(this.loginForm.value);
this._myservice.login(this.loginForm.value).subscribe(
data=>{ console.log(data);
localStorage.setItem('token',data.toString());
this._router.navigate(['/company']);
},
```

```
error =>{ console.log("INVALID CRED");});
}
movetoregister()
{
this._router.navigate(['../newCompany'],{relativeTo:this._activatedRoute});
}

}
```

companylogincomponent.html

```
<div class="container" style="padding-top: 200px;">
<h2>COMPANY LOGIN</h2>
<form [formGroup]="loginForm" >
<div class="form-group">
<label for="uname">Username:</label>
<input type="text" formControlName="companyemail"class="form-control"
id="username" placeholder="Enter username" name="username" >
<span class="help-block" >Please enter a valid username !!!</span>
</div>
<div class="form-group">
<label for="pwd">Password:</label>
<input type="password" formControlName="password" class="form-control" id="pwd"
placeholder="Enter password" name="pswd" >
<span class="help-block" >Please enter a valid Password !!!</span>
</div>
<button type="submit" (click)="Clogin()" class="btn btn-primary">Submit</button>
<button (click)="movetoregister()" style="float: right;" class="btn btn-primary">Sign Up/
New User Registeration</button>
</form>
</div>
```

Admincomponent.html

```
<div class="jumbotron" align="center">
<h1 style="font-family: 'Times New Roman', Times, serif;"><b>Admin Control
Panel</b></h1>
</div>
<div class="container">
<button type="button" class="btn btn-outline-secondary" style="margin-right: 30px;"
(click)="viewrequest()">View Request For NEW Registration </button>
<button type="button" class="btn btn-outline-secondary" style="margin-right:
30px;"(click)="viewexisting()">View Existing Companys</button>
<button type="button" class="btn btn-outline-secondary">UPLOAD RESULT</button>
</div>
<hr>
<div class="container" id="one">
<div class="card">
<div class="card-header">
<h3>Name OF Company: _____</h3>
</div>
```

```
<div class="card-body">
<p>Company URL:_______</p>
<p>Company Email ID:____</p>
<p>Company Contact Number: _____</p>
</div>
<div class="card-footer">
<button type="button" class="btn btn-outline-success" style="margin-right:
30px;">ACCEPT REQUEST</button>
<button type="button" class="btn btn-outline-danger" style="float: right;">REJECT
REQUEST</button>
</div>
</div>
</div>
<div class="container" id="two" style="display: none;">
<div class="card">
<div class="card-header">
<h3>Name OF Company: _____</h3>
</div>
<div class="card-body">
<p>Company URL:_______</p>
<p>Company Email ID:____</p>
<p>Company Contact Number: _____</p>
<p>PASSWORD: _____</p>
</div>
<div class="card-footer">
<button type="button" class="btn btn-outline-success" style="margin-right: 30px;">EDIT/
UPDATE</button>
<button type="button" class="btn btn-outline-danger" style="float: right;">DELETE
COMPANY</button>
</div>
</div>
</div>
```

studentcomponent.html

```
<div class="jumbotron container" align="center">
<div class="row">
<div class="col-sm-8">
<h2>Student Name: {{dval.studentname}}</h2>
<h4>ID: {{dval.studentid}} Department: {{dval.studentdepartment }}Year:
{{dval.studentyear}} </h4>
<h4>Current Avg CGPA:{{dval.studentCGPA}} </h4>
</div>
<div class="col-sm-4">
<h1>Profile Picture</h1>
</div>
</div>
</div>
<div class="container">
<h1>INTERNSHIP</h1>
<button type="button" class="btn btn-outline-secondary" style="margin-right: 30px;"
(click)="viewinternship()">Search for available/upcoming Internships</button>
```

```
<button type="button" class="btn btn-outline-secondary" style="margin-right: 30px;"
(click)="viewnotification()">Notifications</button>
<button type="button" class="btn btn-outline-secondary" style="margin-right: 30px;"
(click)="editresume()">Edit Resume</button>
<button type="button" class="btn btn-outline-secondary" style="margin-right: 30px;"
(click)="logout()">Log Out</button>
</div>
<div class="container" id="one" >
<hr>
<!-- Template for COMPANY INTERNSHIP VIEW -->
<div class="card" *ngFor ="let myint of myinternship">
<div class="card-header">
<div class="row">
<div class="col-sm-8">Name OF Company: {{myint.companyname}} </div>
<div class="col-sm-4">IMAGE</div>
</div>
</div>
<div class="card-body">
<h3>About Internship:</h3>
<p>{{myint.aboutinternship}}</p>
</div>
<div class="card-footer">
<p>Start Date:{{myint.startinternship}} Apply Before:{{myint.applybefore}} <button
type="button" class="btn btn-outline-success" style="float: right;"
(click)="applyforinternship(myint.companyname)">Apply for Internship</button></p>
</div>
</div>
</div>
<div id="two" class="container" style="display: none;">
<hr>
<div class="card" *ngFor="let noti of notifications">
<div class="card-header"><h3>COMPANY NAME:{{noti.companyname}}</h3></div>
<div class="card-body">
<h4>Message</h4>
<p>{{noti.msg}}</p>
</div>

</div>
</div>

<div id="three" class="container" style="display: none;">
<hr>
<h2>UPLOAD RESUME</h2>
<form >
<h3>Resume:</h3>
<div class="custom-file mb-3">
<input type="file" class="custom-file-input" id="customFile" name="filename"
(change)="selectfile($event)">
<label class="custom-file-label" for="customFile">Choose file</label>
</div>
```

```
<button type="submit" class="btn btn-primary" (click)="uploadresume()">Confirm
UPLOAD</button>
</form>
<hr>
<h3>View Uploaded Resume</h3>
<button type="submit" class="btn btn-primary" (click)="viewresume()">View
Resume</button>

</div>
```

studentcomponent.ts

```
import { Component, OnInit } from '@angular/core';
import { MyserviceService } from '../../myservice.service';
import { Router ,ActivatedRoute } from '@angular/router';
import { NgForm,FormGroup,FormControl, Validators } from '@angular/forms';
@Component({
selector: 'app-studenthome',
templateUrl: './studenthome.component.html',
styleUrls: ['./studenthome.component.css']
})
export class StudenthomeComponent implements OnInit {

dval;
myinternship;
notifications;
resume;
constructor(private _myservice:MyserviceService,private _router: Router,private
_activatedRoute:ActivatedRoute)
{
this._myservice.getstudentid()
.subscribe(
data => {this.dval=data;console.log("VALUE FOUND");console.log(data) },
error => {this._router.navigate(['../studentlogin'])});
}

ngOnInit() {
}
viewnotification()
{
document.getElementById("two").style.display="block";
document.getElementById("one").style.display="none";
document.getElementById("three").style.display="none";
var obj={"studentid":this.dval.studentid};

this._myservice.getnotification(obj).subscribe(
data => { this.notifications=data,console.log(data)},
error => {console.log("Nahi mila")});
}
viewinternship()
{
```

```
document.getElementById("two").style.display="none";
document.getElementById("one").style.display="block";
document.getElementById("three").style.display="none";
var obj={"FE":this.dval.FE,"SE":this.dval.SE,"TE":this.dval.TE,"BE":this.dval.BE};
this._myservice.getallinternship(obj)
.subscribe(
data => {this.myinternship=data},
error => {console.log("Nahi mila");});
}
applyforinternship(companyname){
var obj={"companyname":companyname,"s_id":this.dval._id};
console.log("CHECK1");
this._myservice.appliedinternship(obj).subscribe(
data => {console.log("Successfull done");},
error => {console.log("Nahi mila");});
}
logout(){
localStorage.removeItem('token');
this._router.navigate(['../studentlogin']);
}
editresume(){
document.getElementById("two").style.display="none";
document.getElementById("one").style.display="none";
document.getElementById("three").style.display="block";
}
selectfile(event)
{
if(event.target.files.length>0){
const file=event.target.files[0];
this.resume=file;
console.log("got something");
}
}
uploadresume(){
var obj={'file':this.resume}
this._myservice.uploadresume(obj).subscribe(
data => {console.log("Successfull done");},
error => {console.log("Nahi mila");});
}
}
```

3.2 NodeJS Code

Users.js

```
var express = require('express');
var router = express.Router();
var Company= require('../models/company');
var jwt = require('jsonwebtoken');
var Internship = require('../models/internship');
var Student= require('../models/student');
var Appliedinternship = require('../models/appliedinternship');
var multer = require('multer');
var companyname='';

var decodedToken='';

router.post('/addinternship',function(req,res,next){
var internship = new Internship({
aboutinternship: req.body.aboutinternship,
companyname: req.body.companyname ,
```

```
durationofinternship: req.body.durationofinternship,
startinternship: req.body.startinternship,
internshipemail: req.body.internshipemail,
applybefore: req.body.applybefore,
FE: req.body.FE,
SE: req.body.SE,
TE: req.body.TE,
BE: req.body.BE,
COMP: req.body.COMP,
IT: req.body.IT,
ENTC: req.body.ENTC,
// CGPA: req.body.CGPA
});
let promise=internship.save();
promise.then(function(doc){
return res.status(201).json(doc);
})

promise.catch(function(err){
return res.status(501).json({message:'Error While Adding Internship'})
})

});

router.post('/studentapplied',function(req,res,next){
```

```
Student.findOne({_id:req.body.s_id},function(err,founddata){
if(err)
{
console.log("Phir le aya");
}
else
{
var appliedinternship = new Appliedinternship({
companyname:req.body.companyname,
studentname:founddata.studentname,
studentCGPA:founddata.studentCGPA,
studentid:founddata.studentid,
selected:false,
msg:'nothing yet'
});
let promise=appliedinternship.save();
promise.then(function(doc){
return res.status(201).json(doc);
})
promise.catch(function(err){
return res.status(501).json({message:'Error While Applying'});
})
}
})
```

```
});



router.post('/newCompany',function(req,res,next){
var company = new Company({
companyemail: req.body.companyemail,
companyname: req.body.companyname,
companyurl: req.body.companyurl,
companycontact: req.body.companycontact,
password:Company.hashPassword(req.body.password)
});
let promise=company.save();

promise.then(function(doc){
return res.status(201).json(doc);
})
promise.catch(function(err){
return res.status(501).json({message:'Error regesitering user'})
})
});
router.post('/studentlogin',function(req,res,next){
let promise=Student.findOne({studentid:req.body.studentid}).exec();
promise.then(function(doc){
if(doc){
if(doc.password==req.body.password){
```

```
let token=jwt.sign({studentid:doc.studentid},'secret',{expiresIn:'1h'});
return res.status(200).json(token);
}else{
return res.status(501).json({message:"Invalid Credentials"});
}
}
else
{
return res.status(501).json({message:"User StudentId is not registered"})
}

});
promise.catch(function(err){
return res.status(501).json({message:"some internal Error"});
});
});

var SFE,SSE,STE,SBE;
router.get('/studentinternship',filterval,function(req,res,next){
Internship.find({FE:SFE,SE:SSE,TE:STE,BE:SBE},function(err,founddata){
if(err)
{
console.log("Error While Retriving ");
res.status(500).send();
}else
```

```
{
if(founddata.length==0)
{
console.log("no internships");
}
console.log(founddata);
res.send(founddata);
}
});
});



function filterval (req,res,next){
SFE=req.query.SFE;
SSE=req.query.SSE;
STE=req.query.STE;
SBE=req.query.SBE;
next();
}


var compname;
router.get('/shortlist',findcid,function(req,res,next){


Appliedinternship.find({companyname:compname},function(err,founddata){
if(err)
```

```
{
console.log("Error While Retriving ");
res.status(500).send();
}else
{
if(founddata.length==0)
{
console.log("nothing to show");
}
console.log(founddata);
return res.send(founddata);
}
});

});
function findcid(req,res,next){
compname=req.query.companyname;
next();
}

router.post('/Companylogin',function(req,res,next){
let promise=Company.findOne({companyemail:req.body.companyemail}).exec();
promise.then(function(doc){

if(doc){
```

```
if(doc.isValid(req.body.password)){
let token=jwt.sign({companyname:doc.companyname},'secret',{expiresIn:'1h'});
return res.status(200).json(token);
}else{
return res.status(501).json({message:"Invalid Credentials"});
}
}
else
{
return res.status(501).json({message:"User email is not registered"})
}
});
promise.catch(function(err){
return res.status(501).json({message:"some internal Error"});
});
});
router.get('/companyname', verifyToken, function(req,res,next){
companyname =decodedToken.companyname;
return res.status(200).json(decodedToken.companyname);
})
router.get('/studentid', verifyToken, function(req,res,next){
sid =decodedToken.studentid;
Student.findOne({studentid:sid},function(err,founddata){
if(err)
{
```

```
console.log("Error While Retriving ");
res.status(500).send();
}else
{
if(founddata.length==0)
{
console.log("nothing to show");
}
else{
var responseobject=founddata;
}
console.log("FOUND DATA");
console.log(founddata);
res.send(founddata);
}
});
})
function verifyToken(req,res,next){
let token = req.query.token;

jwt.verify(token,'secret', function(err, tokendata){
if(err){
return res.status(400).json({message:' Unauthorized request'});
}
if(tokendata){
```

```
decodedToken = tokendata;
next();
}
})
}
router.get('/uploadedinternship',verifyToken,function(req,res,next){
cn =decodedToken.companyname;
Internship.find({companyname:cn},function(err,founddata){
if(err)
{
console.log("Error While Retriving ");
res.status(500).send();
}else
{
if(founddata.length==0)
{
console.log("nothing to show");
}
else{
var responseobject=founddata;
}
console.log("FOUND DATA");
console.log(founddata);
res.send(founddata);
}
```

```
});
})



router.post('/deleteinternship',function(req,res,next){
console.log("FOUND ROUTER");
console.log(req.body);
var id=req.body._id;

Internship.findOneAndRemove({_id:id},function(err){
if(err)
{
console.log("Error while deleting");
return res.status(500).send();
}
console.log("I DID IT");
})
});

router.post('/selectedstudent',function(req,res,next){
Appliedinternship.findOneAndUpdate({studentid:req.body.studentid},{$set:{sele
cted:true}},function(err,doc){
if(err)
{
console.log("Problem occures");
```

```
}

else{

console.log(doc);

}

})

});



router.post('/rejectedstudent',function(req,res,next){

Appliedinternship.findOneAndDelete({studentid:req.body.studentid},function(er
r,doc){

if(err)

{

console.log("Problem occures");

}

else{

console.log(doc);

}

})

});



router.post('/studentmsg',function(req,res,next){

Appliedinternship.findOneAndUpdate({companyname:req.body.companyname,selected
:true},{$set:{msg:req.body.msg}},function(err,doc){
```

```
if(err)

{

console.log("problem has encountered");

}

})

})


router.post('/getmsg',function(req,res,next){

Appliedinternship.find({studentid:req.body.studentid,selected:true,msg:{$ne:n
ull}},function(err,doc)

{

if(err)

{

console.log("Wrong");

}

if(res)

{

return res.send(doc);

}

})

})



const storage = multer.diskStorage({

destination:(req,file,callBack)=>{
```

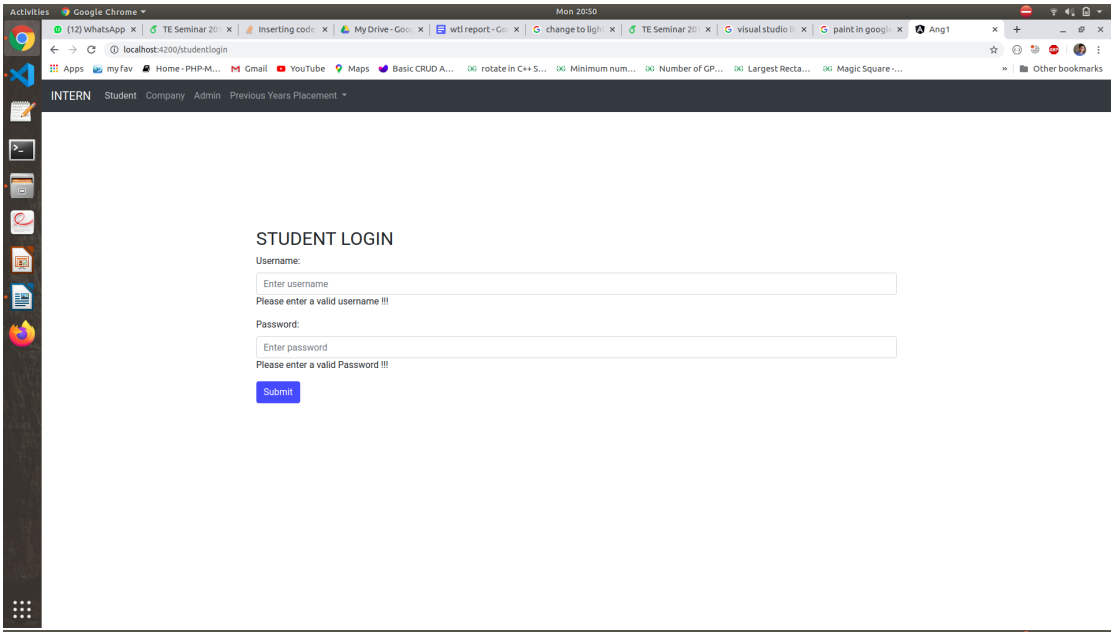
```
callBack(null,'uploads')

},

filename:(req,file,callBack)=>{

callBack(null,`${file.originalname}`)

}

})

var upload = multer({storage:storage});

router.post('/uploadresume',upload.single('file'),(req,res,next)=>{

const file=req.file

console.lof(file.filename);

if(!file){

console.log("NAHI HAI FILE")

}

res.send(file)

})


module.exports = router;
```

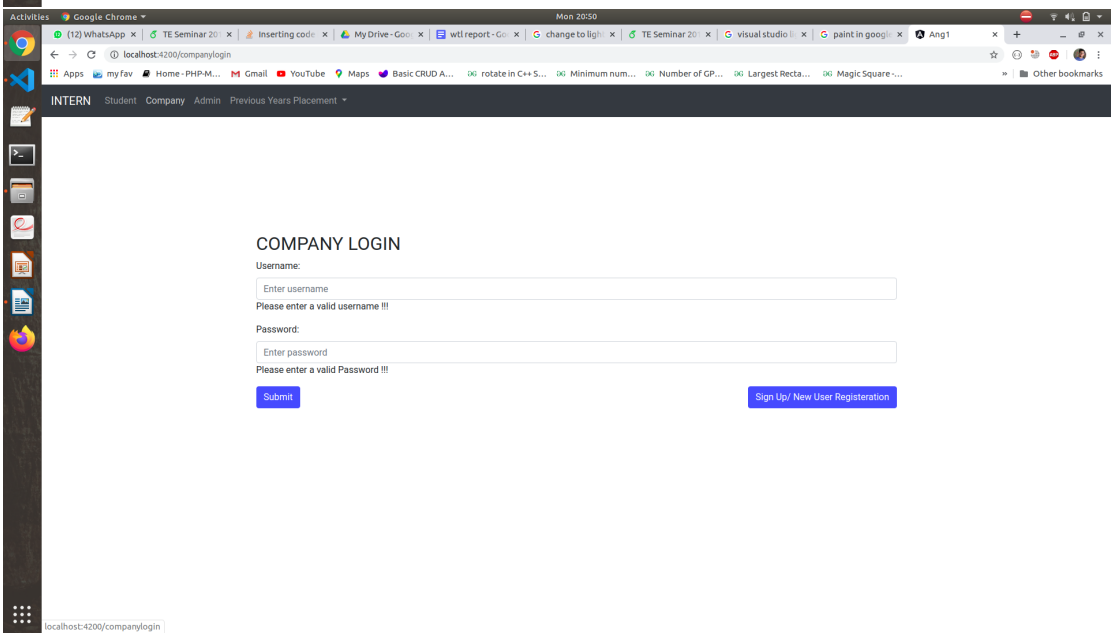
app.js

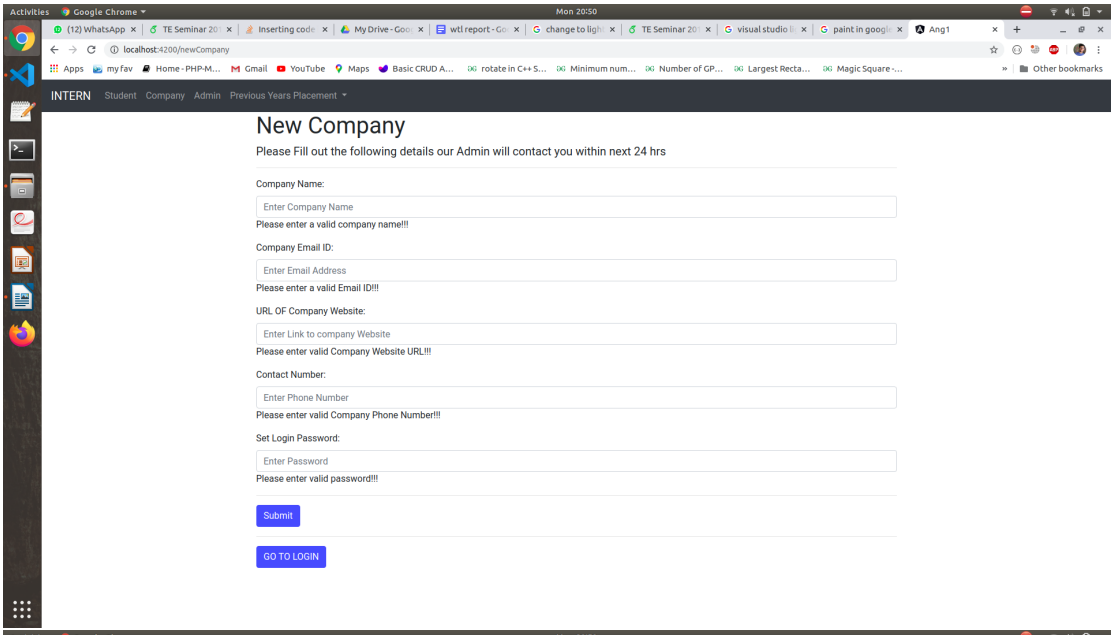
```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');

var app = express();

//add mongoose and connect to db
var mongoose = require('mongoose');
mongoose.connect("mongodb://localhost/INTERN");

//add cors cross origin resource sharing
var cors=require('cors');
app.use(cors({origin:'http://localhost:4200'}));

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

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

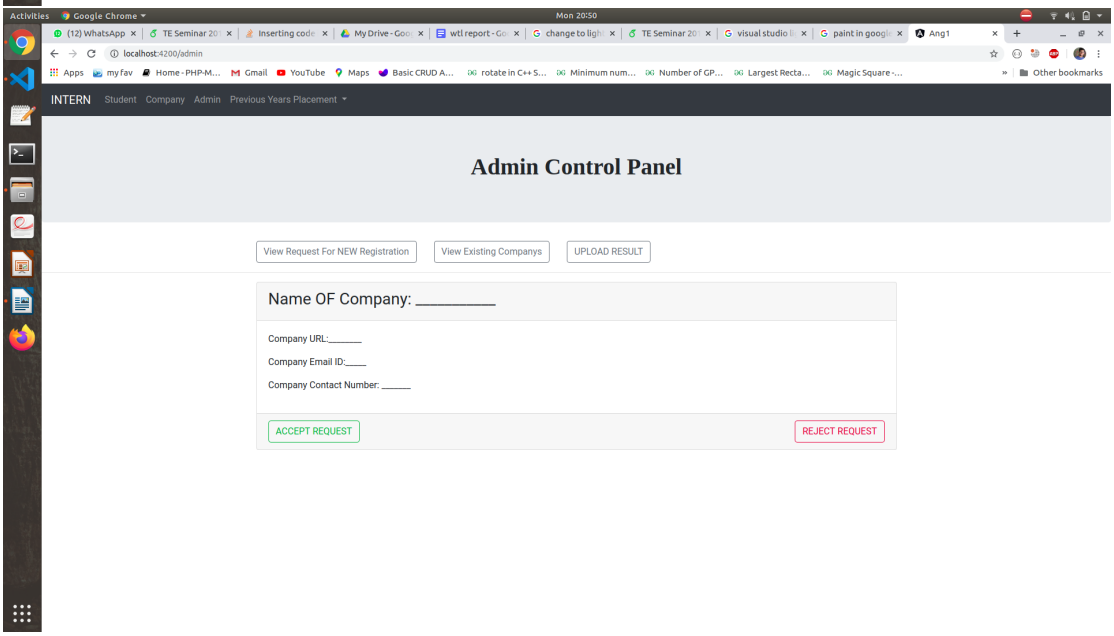
```
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', indexRouter);
app.use('/users', usersRouter);

// 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;
```

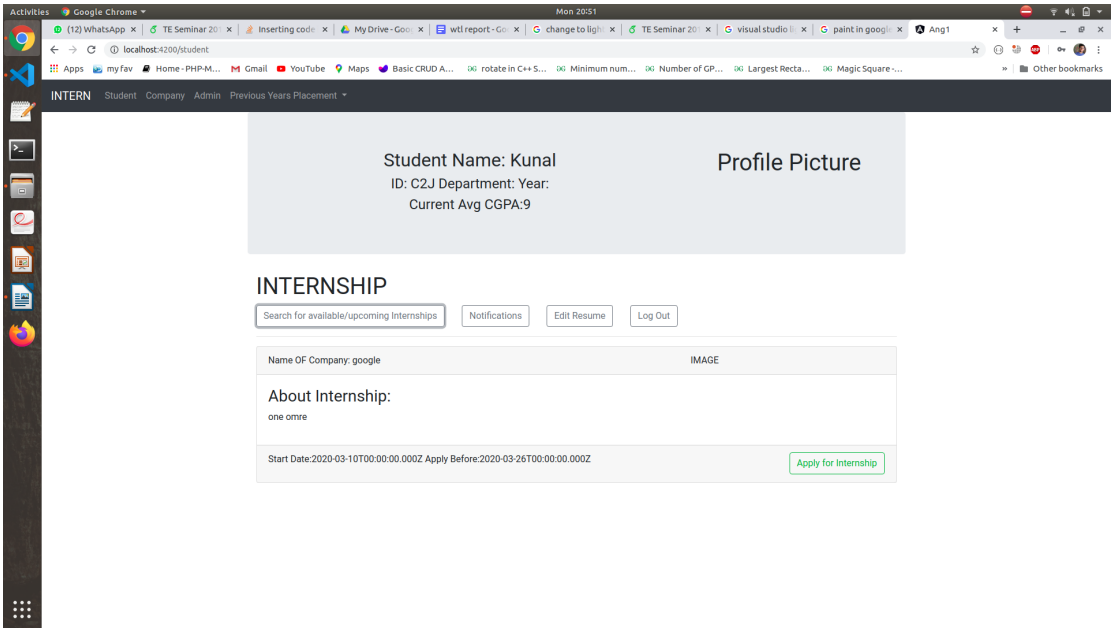
3.3 Screenshots






INTERN
Student
Company
Admin
Previous Years Placement
New Company
Please Fill out the following details our Admin will contact you within next 24 hrs
Company Name:
Enter Company Name
Please enter a valid company name!!!
Company Email ID:
Enter Email Address
Please enter a valid Email ID!!!
URL OF Company Website:
Enter Link to company Website
Please enter valid Company Website URL!!!
Contact Number:
Enter Phone Number
Please enter valid Company Phone Number!!!
Set Login Password:
Enter Password
Please enter valid password!!!
Submit
GO TO LOGIN


INTERN
Student
Company
Admin
Previous Years Placement
Admin Control Panel
View Request For NEW Registration
View Existing Companys
UPLOAD RESULT
Name OF Company: _____
Company URL:_____
Company Email ID:____
Company Contact Number: _____
ACCEPT REQUEST
REJECT REQUEST


Activities
Google Chrome
Mon 20:51
localhost:4200/student
INTERN
Student
Company
Admin
Previous Years Placement
Student Name: Kunal
ID: C2J Department: Year:
Current Avg CGPA:9
Profile Picture
INTERNSHIP
Search for available/upcoming Internships
Notifications
Edit Resume
Log Out
Name OF Company: google
IMAGE
About Internship:
one omre
Start Date:2020-03-10T00:00:00.000Z Apply Before:2020-03-26T00:00:00.000Z
Apply for Internship

4 DEPLOYMENT DETAILS

4.1 Building and serving from disk

During development, typically the ng serve command is used to build, watch, and serve the application from local memory, using webpack-dev-server. When ready to deploy, however, use of the ng build command is done to build the app and deploy the build artifacts elsewhere. Both ng build and ng serve clear the output folder before they build the project, but only the ng build command writes the generated build artifacts to the output folder.

Nearing the end of the development process, serving the contents of the output folder from a local web server gives a better idea of how the application will behave when it is deployed to a remote server. Two terminals are required to get the live-reload experience.

On the first terminal, the ng build command is run in watch mode to compile the application to the dist folder. ng build –watch Like the ng serve command, this regenerates output files when source files change.

On the second terminal, a web server is installed (such as lite-server), and run it against the output folder. For example: lite-server – baseDir="dist/project-name"

The server automatically reloads the browser when new files are output.

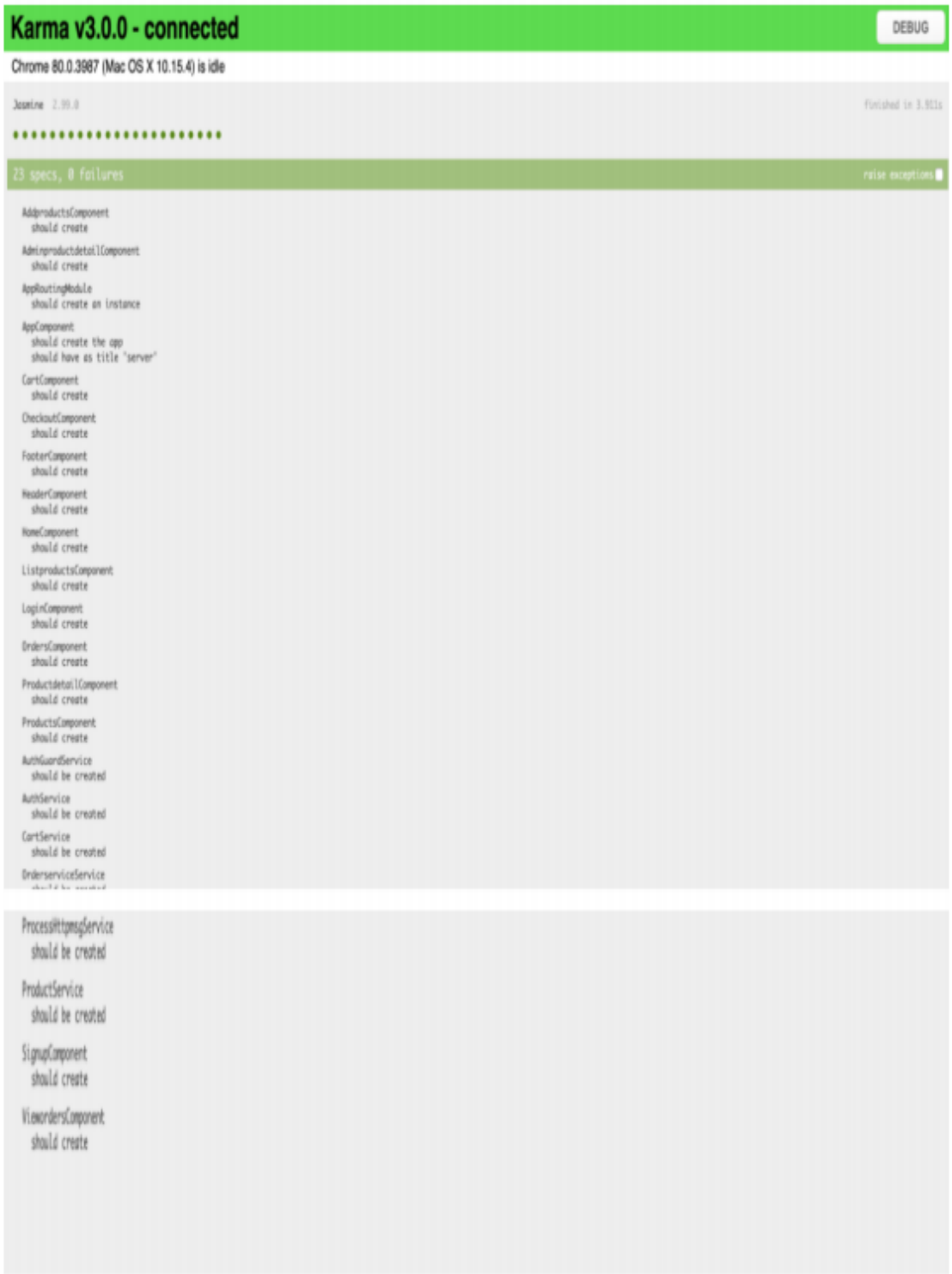
4.2 Deployment to a remote server

A production build is created and the output directory to a web server is copied.For the production build: ng build –prod

• Copy everything within the output folder (dist/ by default) to a folder on the server.

• Configure the server to redirect requests for missing files to index.html

5 TESTING DETAILS



6 CONCLUSION

The project entitled Internship Search Web Application was completed successfully.The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application for connecting people who are looking for internship and people who are looking for interns to hire .This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using HTML,CSS,and management of database using MongoDB. The entire system is secured.Also the project helped us understand about the development phases of a project and software development life cycle(SDLC). We learned how to test different features of a project

FUTURE SCOPE:- There is a scope for further development in our project to a great extent.A number of features can be added to this system in future like adding additional validators to validate users and its resume .Adding filters to only look for best suitable candidates for the internship on basis of resume.

References

[1] https://angular.io/

[2] https://expressjs.com/

[3]] https://mongoosejs.com

[4] https://nodejs.org/en/

[5] https://jwt.io/