

“TP CELL MANAGEMENT SYSTEM”

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Computer Science and Engineering



Project Mentor:

Mr. Ankit Kumar

Designation : Assistant Prof.

Submitted By :

Abhishek Khandelwal

(16ESKCS009)

Department of Computer Science and Engineering
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2019-2020

**Swami Keshvanand Institute of Technology,
Management & Gramothan, Jaipur
Department of Computer Science and Engineering**

CERTIFICATE

This is to certify that Mr Abhishek Khandelwal, a student of B.Tech(Computer Science & Engineering) eight semester has submitted his Project Report entitled "TP Cell Management System" under my guidance.

Mentor

Mr. Ankit Kumar

Designation: Assistant Prof.

Signature

Coordinator

Dr. Neha Janu

Designation: Associate Prof.

Signature

DECLARATION

We hereby declare that the report of the project entitled "TP Cell Management System" is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of "Mr. Ankit Kumar" (Dept. of Computer Science and Engineering) and coordination of "Mrs. Neha Janu" (Dept.of Computer Science and Engineering). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Computer Science and Engineering. It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members

Abhishek Khandelwal(16ESKCS009)

Signature

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization. We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Mr. Ankit Kumar .He has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Dr. Neha Janu for his co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Prof. Dr. Mukesh Gupta, HOD, Department of Computer Science and Engineering, for facilitating, motivating and supporting us during each phase of development of the project. Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Team Members:

Abhishek Khandelwal(16ESKCS009)

Contents

1		2
1.1	Problem Statement and Objective	2
1.2	Introduction to Project	2
1.3	Proposed Logic / Algorithm / Business Plan / Solution / Device	3
1.4	Scope of the Project	3
2		4
2.1	Overall Description	4
2.1.1	Product Perspective	4
2.1.1.1	System Interfaces	4
2.1.1.2	User Interfaces	5
2.1.1.3	Hardware Interfaces	5
2.1.1.4	Software Interfaces	5
2.1.1.5	Project Functions	5
2.1.1.6	Constraints	6
2.1.1.7	Assumption and Dependencies	6
3		7
3.1	System Architecture	7

3.2	High Level Design Diagrams	8
3.2.1	Use Case Diagram	8
3.2.2	Activity Diagram	9
3.2.3	Data-Flow Diagram	10
3.2.4	Class Diagram	10
3.2.5	Flow Chart Diagram	11
4		12
4.1	Introduction to Waterfall Framework	12
5		16
5.1	Functionality Testing	16
6		18
7		26
7.1	CONCLUSION	26
8		27
References		27
9		29

List of Figures

3.1	Use Case Diagram	8
3.2	Activity Diagram	9
3.3	Data Flow Diagram	10
3.4	Class Diagram	10
3.5	Flow Chart Diagram	11
4.1	WaterFall model	13
6.1	Admin:	18
6.2	Student Login Page	19
6.3	Resume Builder	19
6.4	Resume Builder 2	20
6.5	Upload Resume	20
6.6	Company Profile	21
6.7	Student Resume	21
6.8	Admin Table	22
6.9	Student Table	22
6.10	Back-End 3	23
6.11	Front-End 1	23
6.12	Front-End 2	24
6.13	Front-End 3	24

6.14	Back-End 1	25
6.15	Back-End 2	25

Chapter 1

Project Chapter

1.1 Problem Statement and Objective

The TimesJobs survey says that Dissatisfaction with job is on the rise and employees blame poor work life balance for this state of affairs, with a whopping 80 of them looking for change.

TP Cell Management System application matches the skill set with company requirements.

The objective of this project is developing an online application for the Training and Placement Department of the college. The System is an online application that can be accessed throughout the organization and outside as well with proper login provided. This system can be used as an application by the TPO of the college to manage the student information with regards to placement and their skills.

1.2 Introduction to Project

An online application for the Training and Placement Department of the college. The System is an online application that can be accessed throughout the organization and outside as well with proper login provided. This system can

be used as an application by the TPO of the college to manage the student information with regards to placement and their skills.

1.3 Proposed Logic / Algorithm / Business Plan / Solution / Device

It is used to provide proper guidance with mentors in college and help increase skills. It is a voting system which will give an interpretation of the skills particular technology and find creative students in their different types of areas. This online portal provides all requirements and skill sets for companies and updates regularly by TP Cell. Students logging should be able to upload their information in the form of a CV.

1.4 Scope of the Project

It will be useful for the students for applying for jobs in companies based on their skill set. It will be useful for all branch students. It will be secure and only college students will be able to apply for the job.

Chapter 2

Software Requirement Specification

2.1 Overall Description

The system to be developed here is an application. It is a centralized system. It is client system with a centralized database server. All local clients are connected to the centralized server via the Internet. There is a two way communication between different clients and servers. This application is used by both student and admin(TP Cell).

2.1.1 Product Perspective

2.1.1.1 System Interfaces

This application interacts with the through MUI. The interface is simple, easy to handle and self-explanatory.

Once opened, the user will easily come into the flow with the application and easily uses all interfaces properly.

However the basic interface available in our application is

Register panel

Login panel

Admin panel

Student Resume

Resume Upload

Resume Builder

Company profile

2.1.1.2 User Interfaces

Minimum requirements will be as follows: A Desktop mobile Interface Internet connection

2.1.1.3 Hardware Interfaces

Minimum requirements will be as follows: Desktop mobile System

2.1.1.4 Software Interfaces

Minimum requirements will be as follows: Web Browser(Chrome, Google Chrome, Mozilla Firefox, Microsoft Edge)
Internet connection

2.1.1.5 Project Functions

1. The application can be used for both admin(TP Cell) and students.
2. The user can build a resume on the application.
3. The users already student of college can only login.
4. The administrator can download the student resume in any format.
5. The administrator can update the status of the vacancies about job.

2.1.1.6 Constraints

The users can directly communicate to the authority using the forms which provide them the feature to enter the details of their item which has been lost and not present in the existing list. This will send a mail to the admin through which he can check for the respective complaints and actions to be done by them.

2.1.1.7 Assumption and Dependencies

1. There should be an internet connection.
2. The administrator should have a member of TP Cell
3. The student should have a member of TP Cell
4. The user should have a valid Gmail Id.
5. The student should have a valid Gmail Id.

Chapter 3

SYSTEM DESIGN SPECIFICATION

3.1 System Architecture

Front-End: Angular 9

1. adminHeader
2. Authentication
3. Company Profile
4. Resume Upload
5. List of resumes
6. Resume Builder
7. Student Model

Backend: Node.js, Express, MySQL, 1. UserDAO

2. Server
3. MySQL
4. Routes
5. Utils

3.2 High Level Design Diagrams

3.2.1 Use Case Diagram

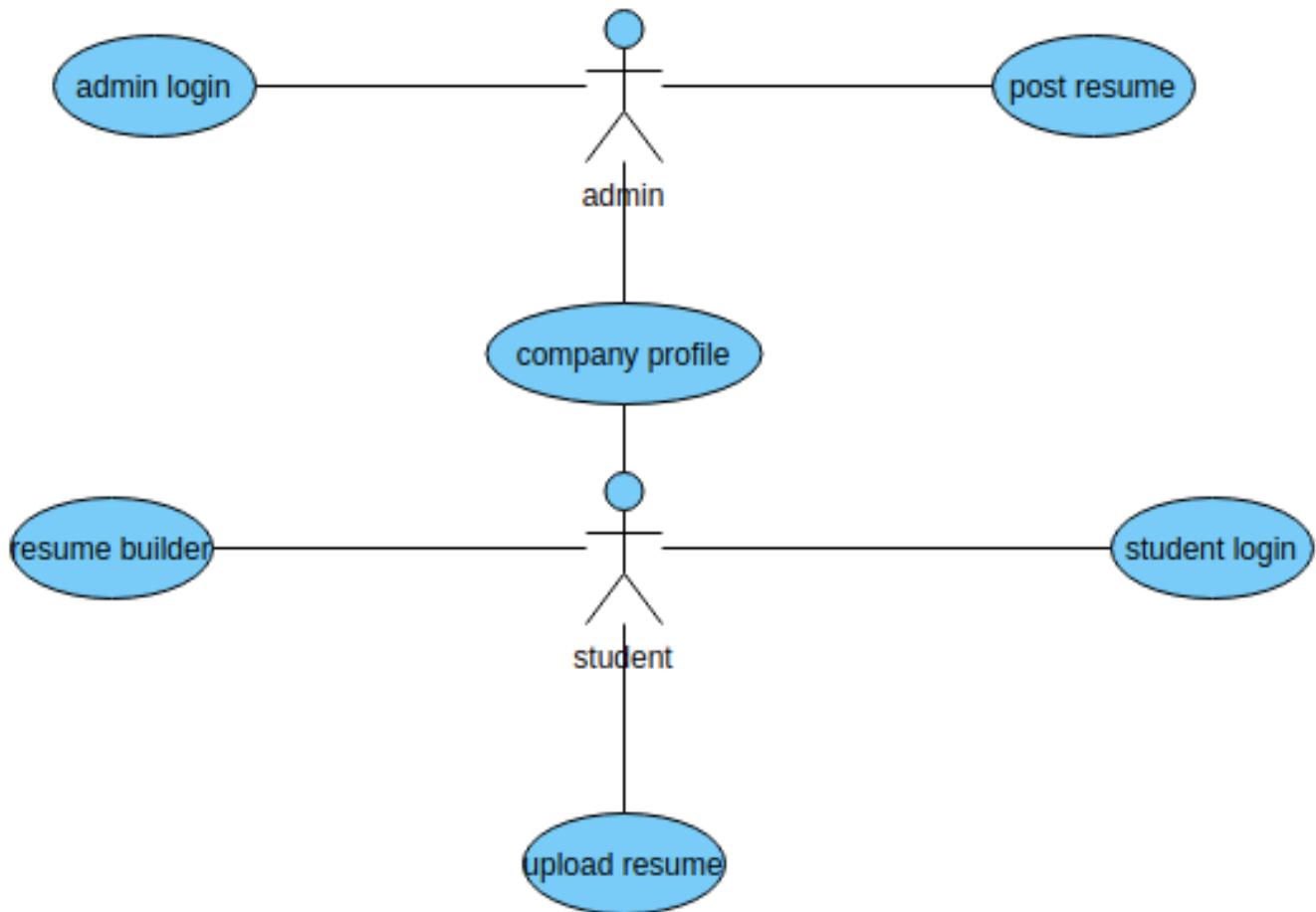


Figure 3.1: Use Case Diagram

This diagram describes a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

In this use case diagram the user here can be referred as a student build resume by resume builder. The admin can post the resume to the company and send acknowledgement to the student.

3.2.2 Activity Diagram

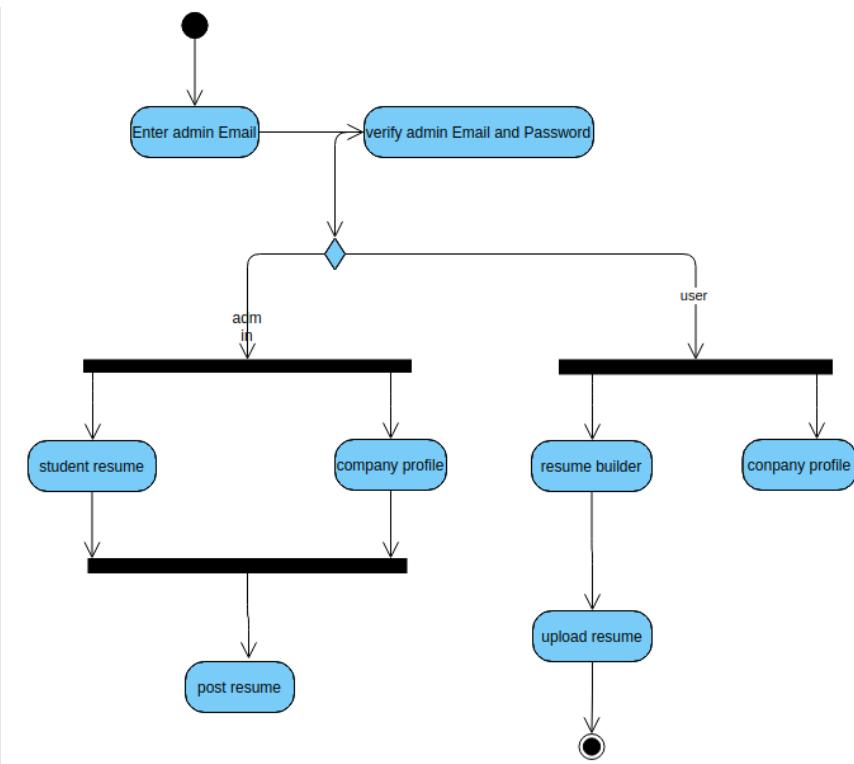


Figure 3.2: Activity Diagram

This diagram describes the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

3.2.3 Data-Flow Diagram

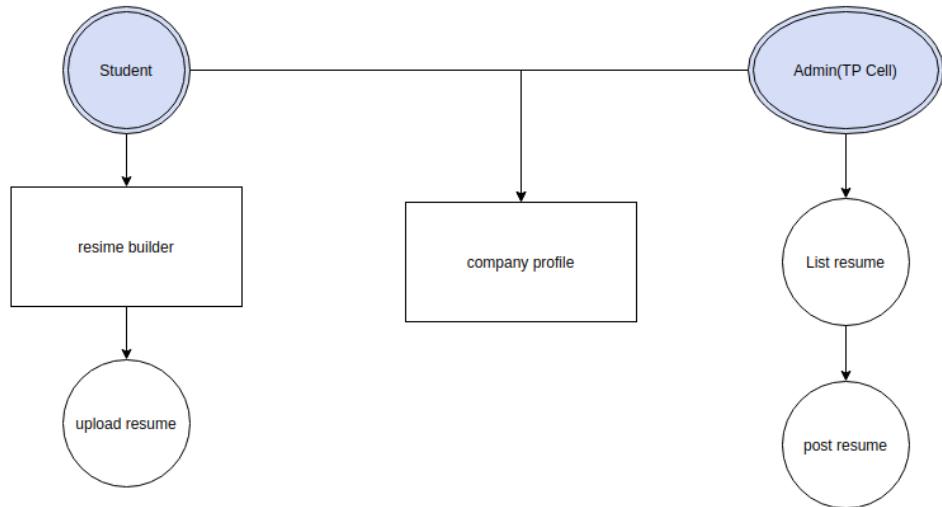


Figure 3.3: Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

3.2.4 Class Diagram

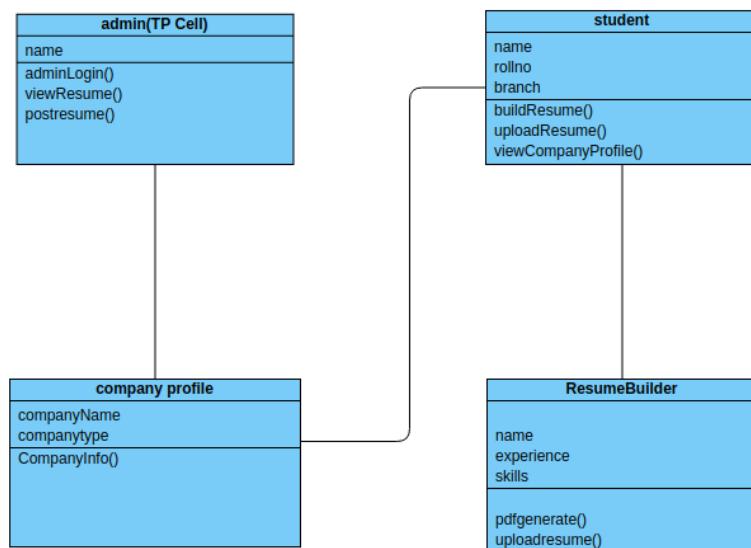


Figure 3.4: Class Diagram

Classes in TP Cell management System - Admin(TP Cell), student, company profile, resume builder
Admin-

Attributes: name, email

Operations: admin login, view resume, post resume

User(student)-

Attributes: company name, company type

Operations: company info

3.2.5 Flow Chart Diagram

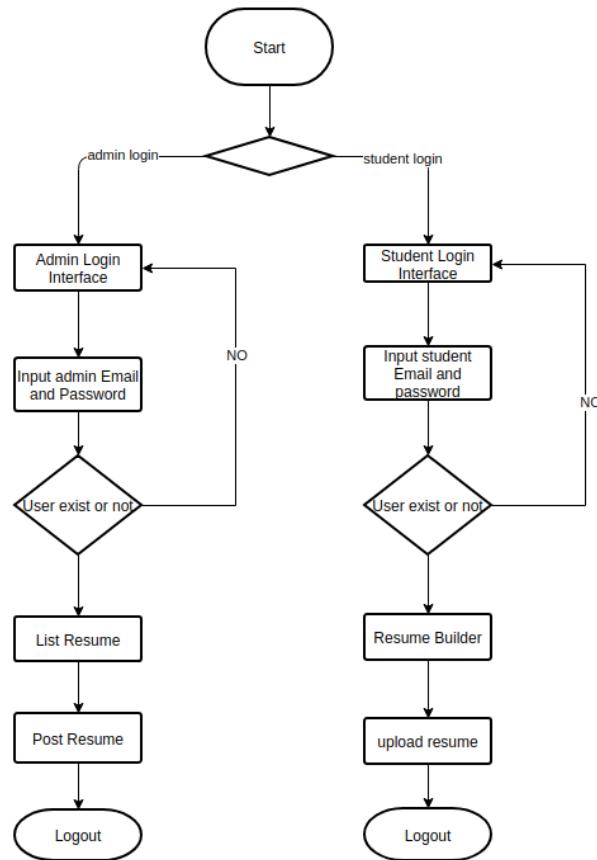


Figure 3.5: Flow Chart Diagram

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

Chapter 4

METHODOLOGY AND TEAM

4.1 Introduction to Waterfall Framework

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as an input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.

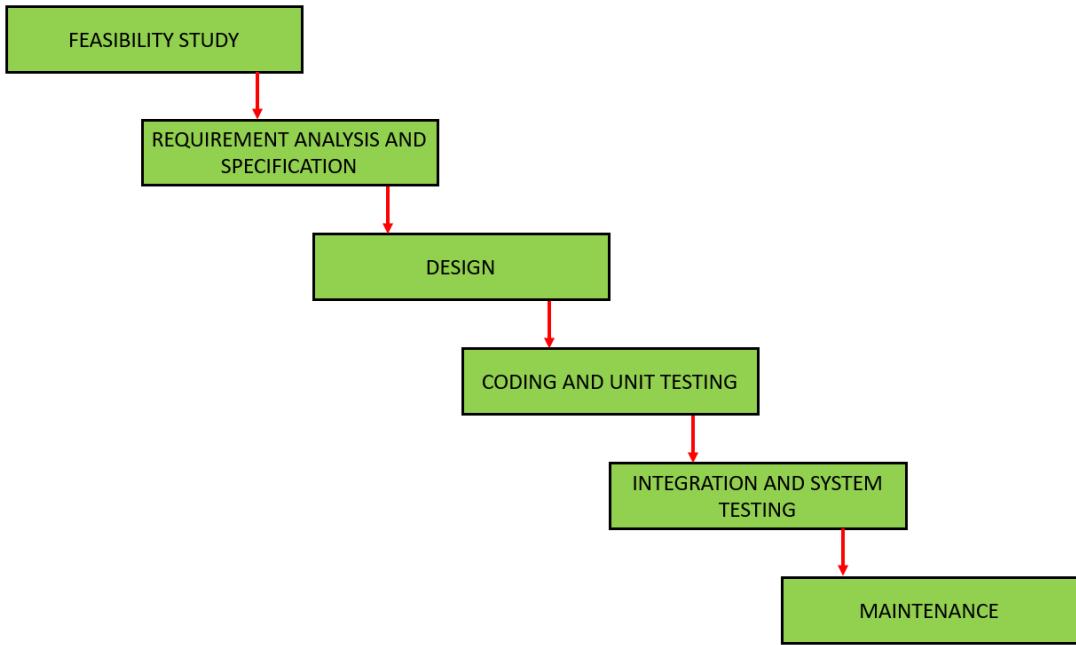


Figure 4.1: WaterFall model

The sequential phases in Waterfall model are-

1. **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
2. **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
3. **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
4. **Integration and Testing:** All the units developed in the imple-

mentation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployment of system:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
6. **Maintenance:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Waterfall Model Pros Cons

Advantage The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantage The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

Chapter 5

System Testing

The designed system has been testing through following test parameters.

5.1 Functionality Testing

In testing the functionality of the web sites the following features were tested:

1. Links
 - (a) Internal Links: All internal links of the website were checked by clicking each link individually and providing the appropriate input to reach the other links within.
 - (b) External Links: Till now no external links are provided on our website but for future enhancement we will provide the links to the candidate's actual profile available online and link up with the elections updates online etc.
 - (c) Broken Links : Broken links are those links which do not divert the page to specific page or any page at all. By testing

the links on our website, there was no link found on clicking which we did not find any page.

2. Forms

- (a) Error message for wrong input : Error messages have been displayed as and when we enter the wrong details (eg. Dates), and when we do not enter any details in the mandatory fields. For example: when we enter wrong password we get error message for acknowledging us that we have entered it wrong and when we do not enter the username and/or password we get the messages displaying the respective errors.
- (b) Optional and Mandatory fields : All the mandatory fields have been marked with a red asterisk (*) and apart from that there is a display of error messages when we do not enter the mandatory fields. For example: As the first name is a compulsory field in all our forms so when we do not enter that in our form and submit the form we get an error message asking for us to enter details in that particular field.

3. Database Testing is done on the database connectivity.

Chapter 6

PROJECT SCREENSHOTS

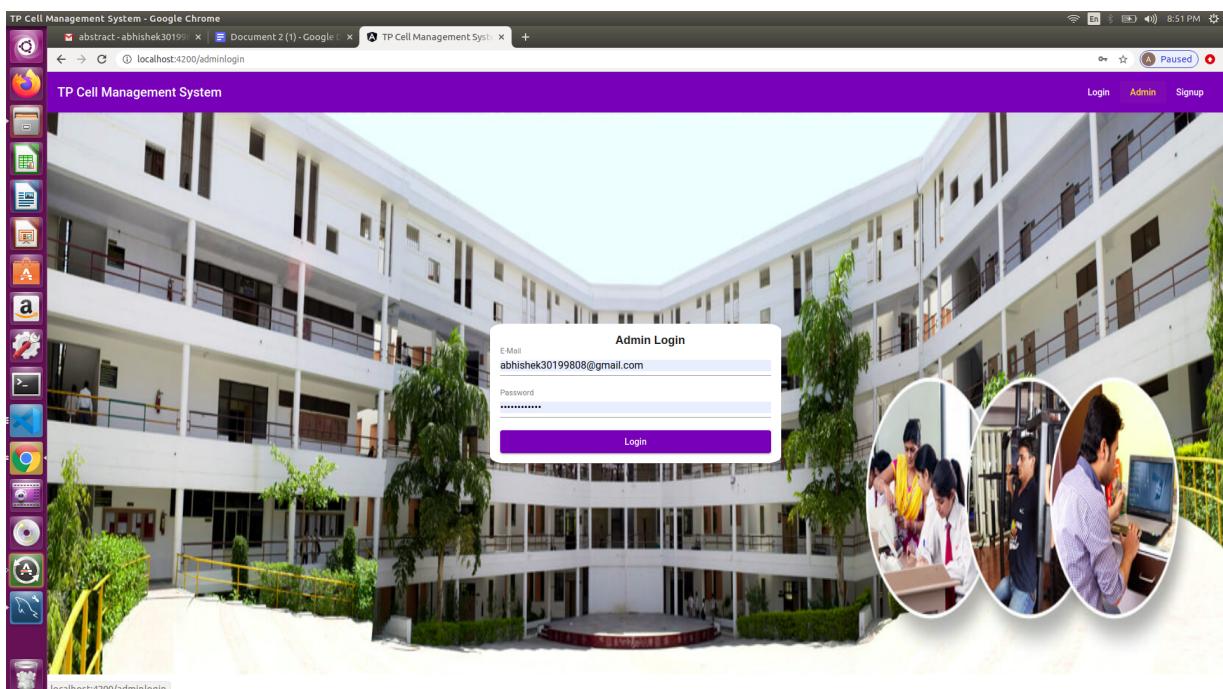


Figure 6.1: Admin:
admin login page uses for authentication by email and password with proper validation

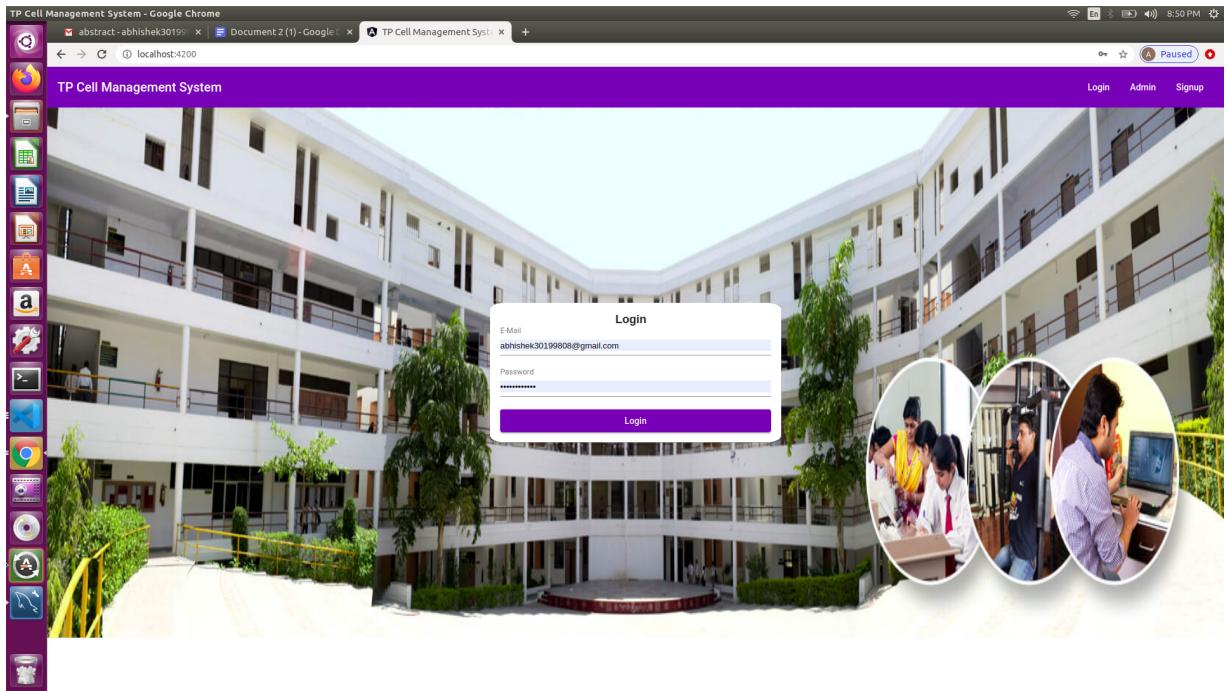


Figure 6.2: Student Login Page
student login page uses for authentication by email and password with proper validation

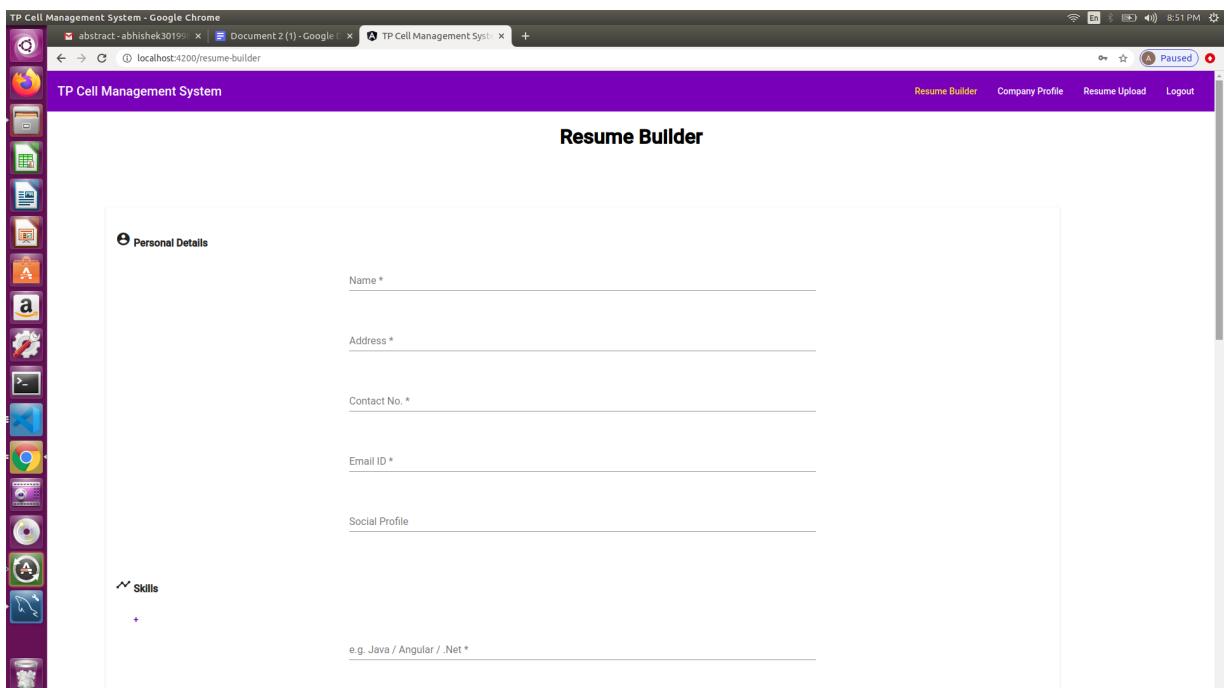


Figure 6.3: Resume Builder
Resume builder build a specific format resume for everone and take printout and download in local system for uploading on server

Job Title *

Experience in months *

Education

School/College *

Passing Year *

Percentage *

Other Details

[Open PDF] [Download PDF] [Print PDF] [Reset]

Show your picture in Resume [Choose file] No file chosen

Figure 6.4: Resume Builder 2

Resume builder build a specific format resume for everyone and take printout and download in local system for uploading on server

TP Cell Management System - Google Chrome

abstract-abhishek3019@gmail.com | Document 2 (1) - Google Docs | TP Cell Management System | +

localhost:4200/file-upload

TP Cell Management System

Resume Builder Company Profile Resume Upload Logout

Select Resume

Upload

Figure 6.5: Upload Resume

upload resume page uses for uploading resume on server(firebase) and list in admin(TP Cell) dashboard)

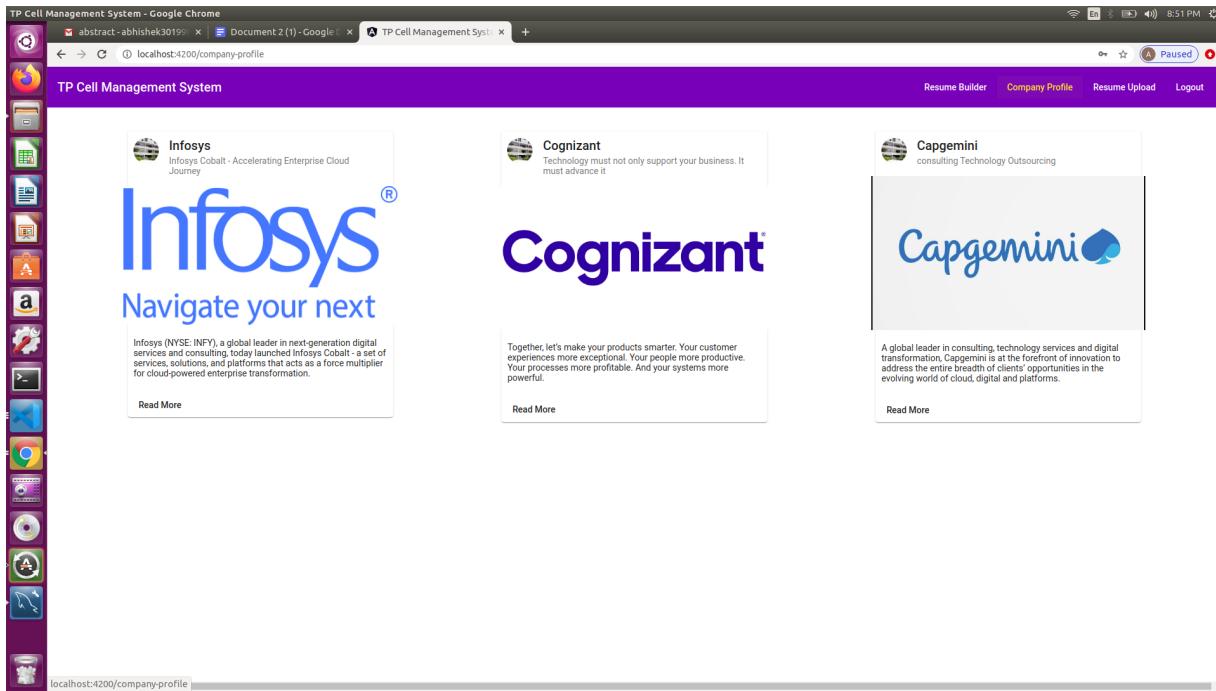


Figure 6.6: Company Profile
Company profile page uses for description of companies

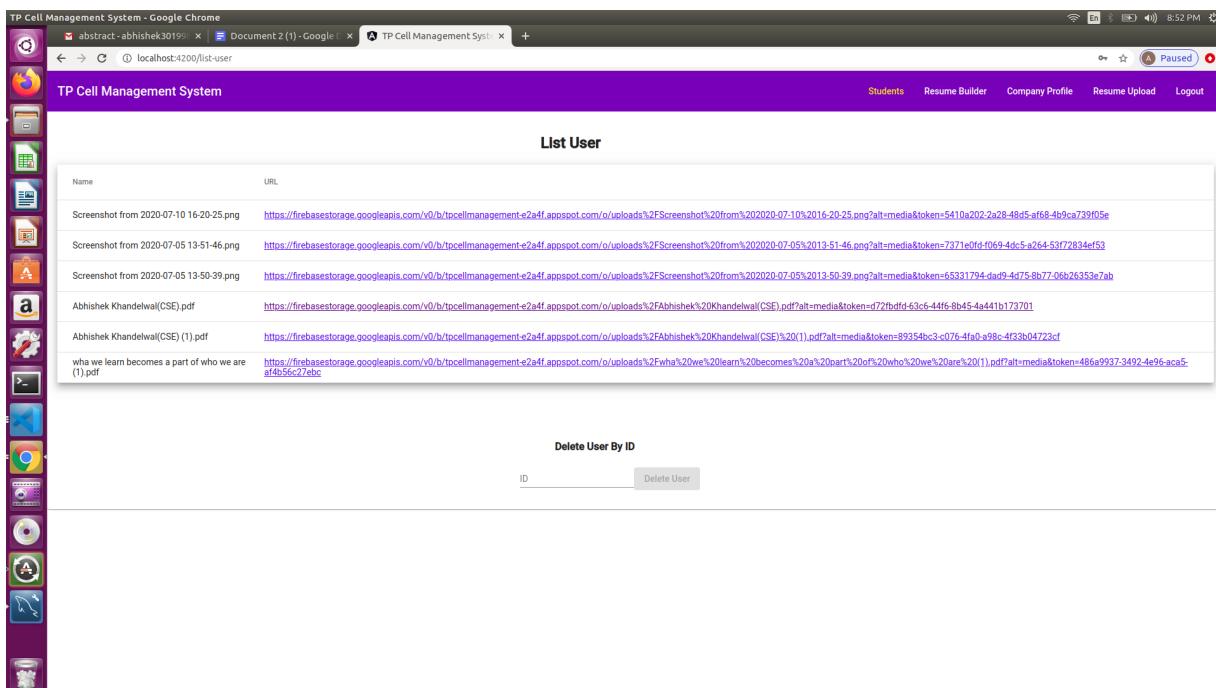


Figure 6.7: Student Resume
List of resumes of student in admin dashboard.admin can send this resume to companies based on profile as per student skill set

The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Schemas):** Shows the 'User' schema with tables: 'admin' and 'users'.
- Central Panel (Query Editor):** A query window titled 'Query 1' with the SQL command: `1 • SELECT * FROM User.admin;`. The result grid displays two rows of data from the 'admin' table.
- Bottom Panel (Object Info):** Shows the structure of the 'admin' table, which has columns: id (int(11) PK), name (varchar(100)), email (varchar(100)), password (varchar(100)), and mobilenr (varchar(100)).
- Bottom Panel (Action Output):** A log of actions taken on the database.

#	id	name	email	password	mobilenr
1	1	Abhishek Khandelwal	abhishek30199808@gmail...	A8890261606a	890261606
2	2	Harshil	abhishekcadbury3008@gm...	A8890261606a	9314932...

Figure 6.8: Admin Table
Admin Database Table

The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Schemas):** Shows the 'User' schema with tables: 'admin' and 'users'.
- Central Panel (Query Editor):** A query window titled 'Query 1' with the SQL command: `1 • SELECT * FROM User.users;`. The result grid displays one row of data from the 'users' table.
- Bottom Panel (Object Info):** Shows the structure of the 'users' table, which has columns: id (int(11) AI PK), name (varchar(100)), email (varchar(100)), password (varchar(100)), and mobilenr (varchar(100)).
- Bottom Panel (Action Output):** A log of actions taken on the database.

#	id	name	email	password	mobilenr
1	3	ra...	ab...	\$26\$10...	93149...

Figure 6.9: Student Table
Student Database table

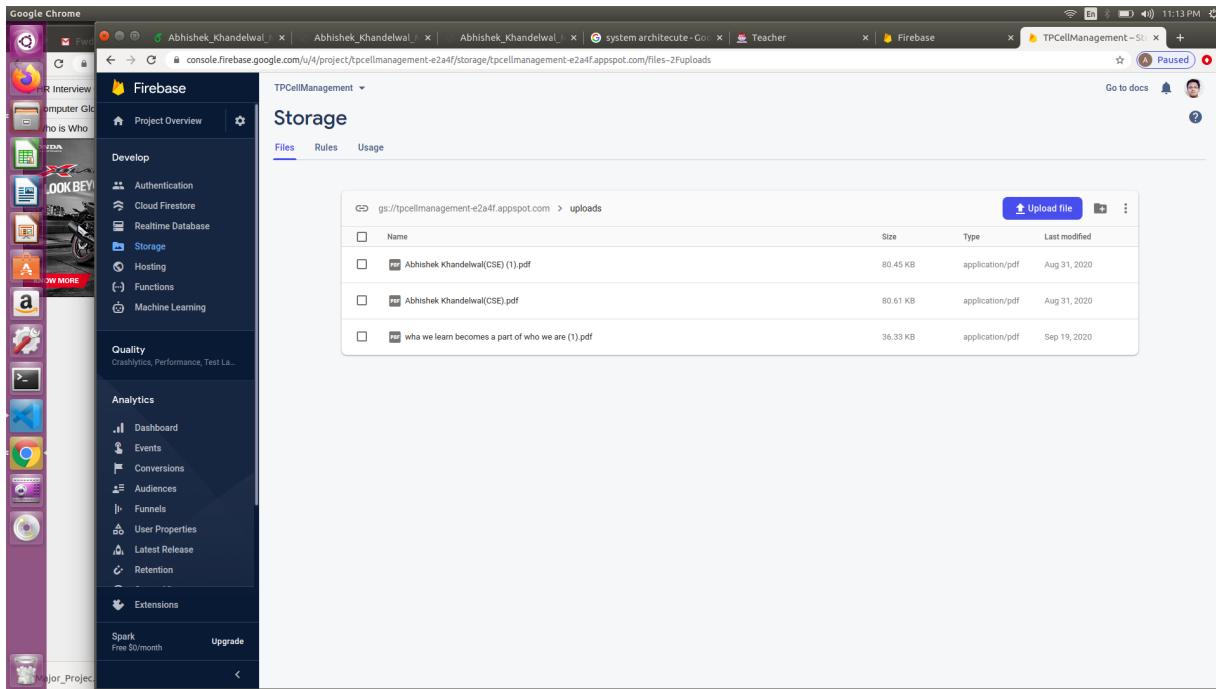


Figure 6.10: Back-End 3
Firebase storage dashboard where all students are store

The screenshot shows the Visual Studio Code interface with the project 'resumebuilder.component.ts - StudentManagementSystem - Visual Studio Code'. The Explorer sidebar shows the project structure under 'STUDENTMANAGEMENTSYSTEM' with 'src' and 'Auth' expanded. The code editor displays the 'resumebuilder.component.ts' file, which contains TypeScript code for a resume builder component. The terminal at the bottom shows the command 'node server.js' being run, with the output 'app listening on port: 3000'. The status bar indicates the file is 3.9.7.

```

import { Component, OnInit } from '@angular/core';
import { Resume, Experience, Education, Skill } from './script.service';
import pdfMake from 'pdfmake/build/pdfmake';
import pdfFonts from 'pdfmake/build/vfs_fonts';
@Component({
  selector: 'app-resumebuilder',
  templateUrl: './resumebuilder.component.html',
  styleUrls: ['./resumebuilder.component.css']
})
export class ResumebuilderComponent implements OnInit {
  resume = new Resume();
  degrees = ['B.E.', 'M.E.', 'BCA', 'MCA'];
  constructor(private scriptService: ScriptService) {
    this.resume = JSON.parse(sessionStorage.getItem('resume') || '{}');
    if (!this.resume.experiences || this.resume.experiences.length === 0) {
      this.resume.experiences.push(new Experience());
    }
    if (!this.resume.educations || this.resume.educations.length === 0) {
      this.resume.educations.push(new Education());
    }
  }
}

```

Figure 6.11: Front-End 1
content hierarchy of project

```

src > app > resumebuilder > resumebuilder.component.ts ...
1 import { Component, OnInit } from '@angular/core';
2 import { Resume, Experience, Education, Skill } from './resume';
3 import { ScriptService } from './script.service';
4 import pdfMake from 'pdfmake/build/pdfmake';
5 import pdfFonts from 'pdfmake/build/vfs_fonts';
6
7 @Component({
8   selector: 'app-resumebuilder',
9   templateUrl: './resumebuilder.component.html',
10  styleUrls: ['./resumebuilder.component.css']
11})
12
13 export class ResumebuilderComponent implements OnInit {
14
15  resume = new Resume();
16
17  degrees = ['B.E.', 'M.E.', 'BCA', 'MCA'];
18
19  constructor(private scriptService: ScriptService) {
20    this.resume = JSON.parse(sessionStorage.getItem('resume')) || new Resume();
21    if (!this.resume.experiences || this.resume.experiences.length === 0) {
22      this.resume.experiences = [];
23      this.resume.experiences.push(new Experience());
24    }
25    if (!this.resume.educations || this.resume.educations.length === 0) {
26      this.resume.educations = [];
27      this.resume.educations.push(new Education());
28    }
29    if (!this.resume.skills || this.resume.skills.length === 0) {
30      this.resume.skills = [];
31      this.resume.skills.push(new Skill());
32    }
33
34    console.log('Loading External Scripts');
35    this.scriptService.load('pdfmake', 'vfsFonts');
36  }
37
38  addExperience() {
39    this.resume.experiences.push(new Experience());
40  }
41
42  addEducation() {

```

Figure 6.12: Front-End 2
code of resume builder resumebuilder.ts file

```

src > app > Auth > login > login.component.ts ...
1 import { Component, OnInit, ViewChild, eCodegenComponentFactoryResolver } from '@angular/core';
2 import { FormBuilder, FormGroup, Validators } from '@angular/forms';
3 import { AuthService } from '../auth.service';
4 import { MatSnackBar, MatSnackBarConfig, MatSnackBarHorizontalPosition, MatSnackBarVerticalPosition } from '@angular/material/snack-bar';
5 import { Router } from '@angular/router';
6
7 @Component({
8   selector: 'app-login',
9   templateUrl: './login.component.html',
10  styleUrls: ['./login.component.css']
11})
12 export class LoginComponent implements OnInit {
13
14  public loginForm: FormGroup;
15  public submitted = false;
16  public isLoading = false;
17  public code;
18
19  public validMessage = 'Login Successfully..';
20  public invalidMessage = 'Enter Valid Credentials';
21  public actionButtonLabel = '';
22  public horizontalPosition: MatSnackBarHorizontalPosition = 'center';
23  public verticalPosition: MatSnackBarVerticalPosition = 'bottom';
24  public setAutoHide = true;
25  public autoHide = 2000;
26  public action = true;
27  public addExtraClass = false;
28
29
30  constructor(private authService: AuthService, private formBuilder: FormBuilder, private snackBar: MatSnackBar) {
31
32
33  ngOnInit() {
34    this.loginForm = this.formBuilder.group({
35      email: ['', [Validators.required, Validators.pattern(`^([A-Za-z0-9_\-.])+@[A-Za-z0-9_\\.-]+\\.[A-Za-z]{2,}$`)]],
36      password: ['', [Validators.required, Validators.minLength(8)]],
37    });
38  }
39
40  // snack bar
41  validLoginOpenSnackbar() {
42    const config = new MatSnackBarConfig();

```

Figure 6.13: Front-End 3
code of student login page

```

server.js - StudentManagementSystem - Visual Studio Code
OPEN EDITORS
STUDENTMANAGEMENTSYSTEM
TS upload-file.service.spec.ts
TS user.service.ts
TS user.service.spec.ts
> assets
> environments
★ favicon.ico
index.html
TS main.ts
TS polyfills.ts
# styles.css
TS test.ts
# browserslistrc
.editorconfig
.gitignore
angular.json
config.js
karma.conf.js
mysqlConfig.js
package.json
package-lock.json
README.md
routes.js
server.js
tsconfig.json
tsconfig.app.json
tsconfig.base.json
tsconfig.spec.json
tslint.json
userDAO.js
userservice.js
util.js
> OUTLINE
> TIMELINE
> NPM SCRIPTS
JS server.js > app.use(callback)
1 let app = require('express')();
2
3 server = require('http').Server(app),
4 bodyParser = require('body-parser'),
5 cors = require('cors'),
6 http = require('http'),
7 path = require('path');
8 const jwt = require('jsonwebtoken');
9 app.use(cors());
10
11 let userRoute = require('./routes');
12
13
14 app.use(bodyParser.json());
15 app.use(bodyParser.urlencoded({extended: false }));
16
17 app.use('/user', userRoute);
18
19 app.use((req, res, next) => {
20   res.setHeader("Access-Control-Allow-Origin", "*");
21   res.setHeader(
22     "Access-Control-Allow-Headers",
23     "Origin, X-Requested-With, Content-Type, Accept, Authorization"
24   );
25   res.setHeader(
26     "Access-Control-Allow-Methods",
27     "GET, POST, PATCH, PUT, DELETE, OPTIONS"
28   );
29   next();
30 });
31
32
33 server.listen(3000,function(){
34   console.log('app listening on port: 3000');
35 });
36

```

Ln 28, Col 30 Spaces: 2 UTF-8 LF Javascript

Figure 6.14: Back-End 1
code for backend server cors and express module

```

routes.js - StudentManagementSystem - Visual Studio Code
OPEN EDITORS
STUDENTMANAGEMENTSYSTEM
TS upload-file.service.spec.ts
TS user.service.ts
TS user.service.spec.ts
> assets
> environments
★ favicon.ico
index.html
TS main.ts
TS polyfills.ts
# styles.css
TS test.ts
# browserslistrc
.editorconfig
.gitignore
angular.json
config.js
karma.conf.js
mysqlConfig.js
package.json
package-lock.json
README.md
routes.js
server.js
tsconfig.json
tsconfig.app.json
tsconfig.base.json
tsconfig.spec.json
tslint.json
userDAO.js
userservice.js
util.js
> OUTLINE
> TIMELINE
> NPM SCRIPTS
JS routes.js > router.post("/login")(callback) > query(SELECT * FROM users WHERE email = ?)(callback) > bcrypt.compare()(callback)
56   console.log(fetchName);
57   if(results.length > 0) {
58     bcrypt.compare(req.body.password, results[0].password , function(error,results) {
59       if(error){
60         res.status(401).json({
61           message: "Auth failed"
62         });
63       }
64       if(results){
65         const token = jwt.sign(
66           { email: fetchEmail },
67           "secret_this_should_be_longer",
68           { expiresIn: "1h" }
69         );
70         res.status(200).json([
71           token,
72           expiresIn: 3600,
73           code : 200,
74           name: fetchName,
75         ]);
76       });
77     });
78   });
79 }
80 });
81
82
83 // adminlogin api
84 router.post("/adminlogin", (req, res) => {
85   var fetchName;
86   var fetchEmail;
87   var email = req.body.email;
88   console.log('email'+ email);
89   dbConfig.getDB().query('SELECT * FROM admin WHERE email = ?',[email], function (error, results) {
90     if(error){
91       res.status(401).json({
92         message: "Auth failed"
93       });
94     }
95     fetchEmail = results[0].email;
96     fetchName = results[0].name;
97     console.log('fetchEmail' + fetchEmail);

```

Ln 74, Col 33 Spaces: 2 UTF-8 LF JavaScript

Figure 6.15: Back-End 2
code for generating token on server

Chapter 7

PROJECT SUMMARY AND CONCLUSIONS

7.1 CONCLUSION

It is used to provide proper guidance with mentors in college and help increase skills. It is a voting system which will give an interpretation of the skills particular technology and find creative students in their different types of areas. This online portal provides all requirements and skill sets for companies and updates regularly by TP Cell. Students logging should be able to upload their information in the form of a CV. Visitors/College representatives logging in may also access/search any information put up by students. This portal is useful for students as well as faculty members of college for enhancing skills and communication with industrial requirements up to date. The system provides the facility of viewing both the personal and academic information of the students and company and also insertion and deletion of records by the administrator. Eligible students will receive an email including the details of the company.

Chapter 8

FUTURE SCOPE

TP Cell Management System application will support students for enhancing their skills as per industry requirement. It provides interaction with companies in colleges.

References

https://www.tutorialspoint.com/uml/uml_component_diagram.htm

<https://www.guru99.com/deployment-diagram-uml-example.html>

<https://www.lucidchart.com/documents>

https://www.nrega.nic.in/netnrega/mgnrega_new/NregaHome.aspx

https://www.tutorialspoint.com/uml/uml_class_diagram.htm

<https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial>

Chapter 9

Papers