School of Engineering and Applied Science

Final Report On

"University Placement Automation"

Course Code: CSE540

Course Name: Cloud Computing

BY

Priyanshi Deliwala AU1741047
Nimil Shah AU1741048
Aayushi Ganatra AU1741050

UNDER THE GUIDANCE OF

Prof. Sanjay Chaudhary
and Technical Assistant, Aayush Shah

Acknowledgements

We are profoundly grateful to **Prof. Sanjay Chaudhary** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards to the Technical Assistant and Project Coordinator, Aayush Shah whose invaluable guidance supported us in completing this project.

Priyanshi Deliwala
Nimil Shah
Aayushi Ganatra

Contents

1 Project Goal **2**

2 Objectives **3**

3 Design of the Project **4**

3.1 Architectural Design 4

3.2 Use Case Of The System 5

4 List of Cloud Services **6**

5 Development of Cloud Services **8**

5.1 Hardware and Software platforms used 8

5.2 Brief about each service 8

5.3 List of PaaS / SaaS 9

5.4 Technical details about each service 10

5.5 List of programs written or to be written 11

6 Implementation of each cloud service **12**

6.1 Details about each PaaS / SaaS from implementation point of view . 12

7 Testing of services and snap shots of results **14**

8 Conclusion **21**

References **22**

List of Figures

3.1 **Architectural Design** 4
3.2 **Use Case of the System** 5

5.1 **EC2 Instance** 9
5.2 **AWS Services** 10

6.1 **Database Deployment on AWS** 12
6.2 **Database Deployment on AWS** 13

7.1 **API Testing using POSTMAN** 14
7.2 **API Testing using POSTMAN** 15
7.3 **JSON Data using APIs** 15
7.4 **Back-end API** 16
7.5 **Front-end API** 16
7.6 **DynamoDB Connectivity** 17
7.7 **Test Connection with Local Database** 17
7.8 **Database Deployment on AWS** 18
7.9 **Successful EC2 Instance Creation** 18
7.10 **Creating Virtual Server using WinScp** 19
7.11 **NodeJS application Deployment using EC2** 19
7.12 **Final Student Side Module** 20
7.13 **Final Company Side Module** 20

Chapter 1

Project Goal

- In existent systems of Placement Cells, the work is done by human intervention which increases the chances for errors and it also causes much more stress to the people which are engrossed in the work.
- Increased use of automated systems widely has led to its use in many fields and it can also be seen now in Educational infrastructures.
- Our goal with this project is to develop a web based application of university automation so that all the work of college can be done without much manual work
- The system handles student as well as company data and efficiently displays all this data to respective sides.
- The inter-connectivity among these modules (Student, Placement Officer, and the Companies) reduces the time to perform different operational task.
- All the tasks that are important and need to be performed efficiently are included in web application. Our system basically focuses on providing student with the information of the companies and their eligibility criteria. So that the students can look for which companies their profile suits best and prepare for the same.
- Each user is provided with role-based access rights and each user has its own username and tab where he / she can handle his / her own personal details.

Chapter 2

Objectives

1. This project is intended to create an online application for the college's placement department.
2. All the tasks that are important and need to be performed efficiently are included in web application.
3. Our system basically focuses on providing student with the information of the companies and their eligibility criteria. So that the students can look for which companies their profile suits best and prepare for the same.
4. Our services involve: Collecting student records, Authentication of the student profiles, Proper login with time and role based secured access is provided to Placement Officer, Company, and students.
5. Students logging should be able to upload their information in the form of a Resume.
6. Unique id of student here will naturally act as primary key in database differentiating between students so that students can view only their own information and do not hinder with other student's data.
7. This makes the system highly reliable and secure.
8. Students can view the news feed, and check all the ongoing and upcoming events and what preparation they can do based on requirements.
9. Our framework also ensures that the user interface is user-friendly so that neither the students nor the company have any issues understanding the web application definition and have no navigation concerns.

Chapter 3

Design of the Project

3.1 Architectural Design

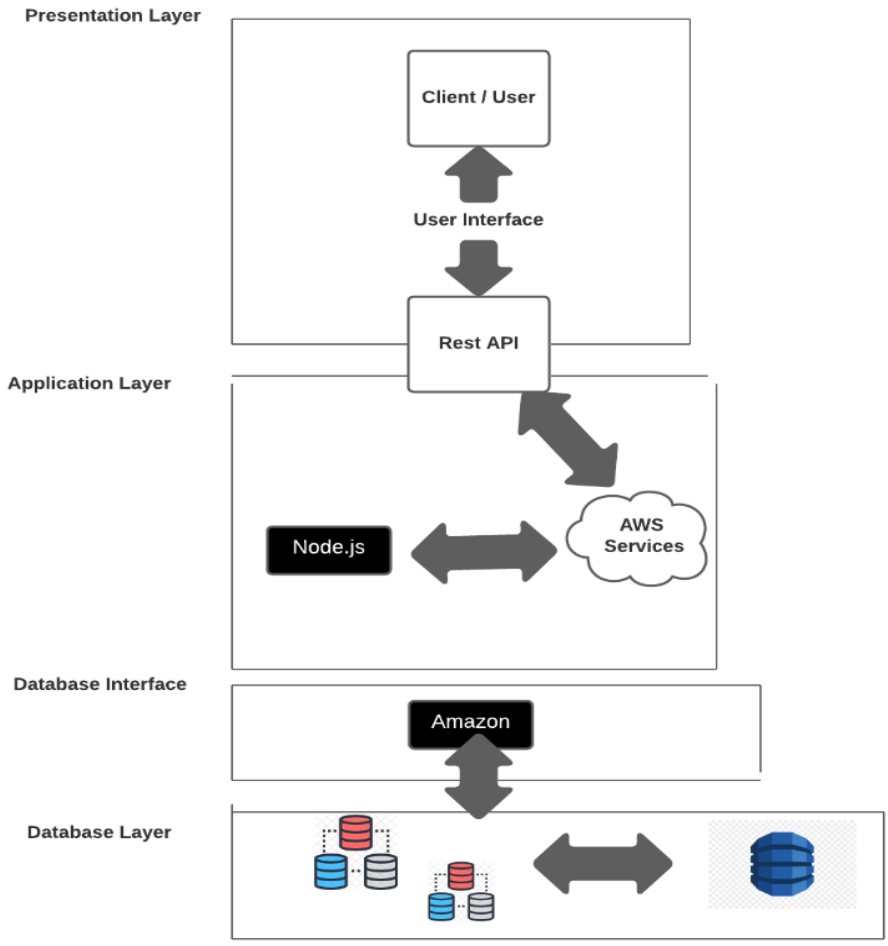

Figure 3.1: Architectural Design

3.2 Use Case Of The System

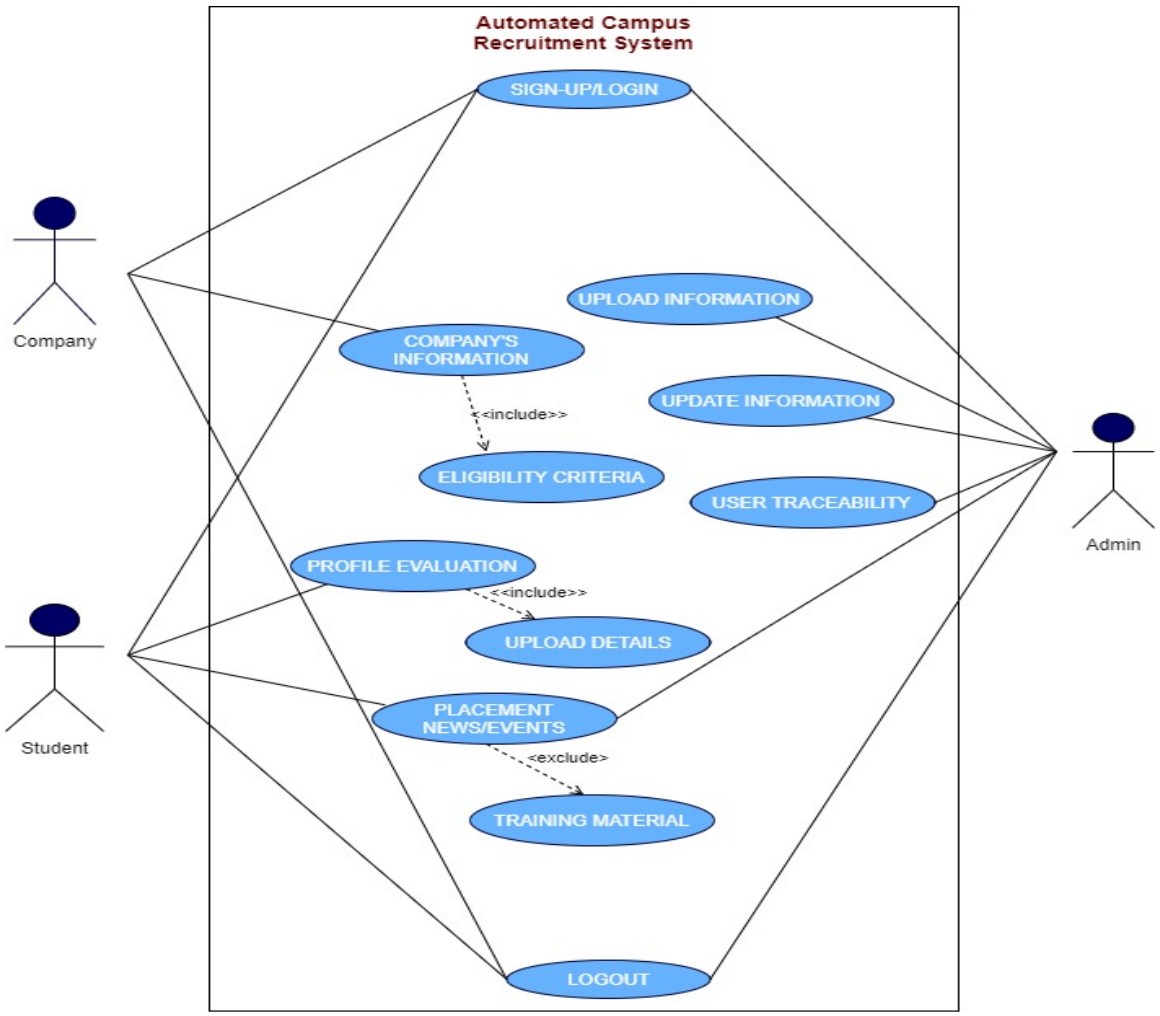

Figure 3.2: Use Case of the System

Chapter 4

List of Cloud Services

The aim of the proposed Automated Placement system is to avoid all the disadvantages of the current method. More functionality than the current framework is included.

1. **Admin (Placement Officer) Module**
    - Login
    - Service: Placement Officer can prepare schedule events about companies and flash it to the every student login at front page.
    - Update/Delete Student and it's Information
    - The administrator can view and approve the various application forms.
    - Service: Based on the company requirements, the Placement Officer can generate the list of the students with responses.
2. **Student Module**
    - Student Registration/Login
    - They can manage the profile by putting all the information.
    - A student can check the company's coming on in the university.
    - Refer to the company's requirements.
    - Students can access relevant resources placed online for them.
3. **Company Module**
    - Login
    - Service: Company's information, URL, Eligibility Criteria and no of vacancies will be provided.
    - Company can edit their own information.

- Company can see how many students applied based on the criteria provided.
- Service: Schedule Interview or further rounds.

Chapter 5

Development of Cloud Services

5.1 Hardware and Software platforms used

1) Hardware Platforms

- Amazon Web Services

2) Software Platforms

- Node JS (Webframework in JavaScript)
- phpMyAdmin Admin (Local Machine Database)
- Dynamo DB (Cloud Database)
- React JS (Frontend Framework)

5.2 Brief about each service

1) Admin (Placement Officer) Module

- Login
- Service: Placement Officer can prepare schedule events about companies flash it to the every student login at front page.
- Update/Delete Student and it's Information
- The Administrator can view and approve the various application forms.
- Service: Based on the company requirements, the Placement Officer can generate the list of the students with responses.

2) Student Module

- Student Registration/Login
- They can manage the profile by putting all the information.
- A student can check the company's coming on in the university.
- Refer to the company's requirements.
- Students can access relevant resources placed online for them.

3) Company Module

- Login
- Service: Company's information, URL, Eligibility Criteria and no of vacancies will be provided.
- Company can edit their own information.
- Company can see how many students applied based on the criteria provided.
- Service: Schedule Interview or further rounds.

5.3 List of PaaS / SaaS

1) Paas: EC2 Instance

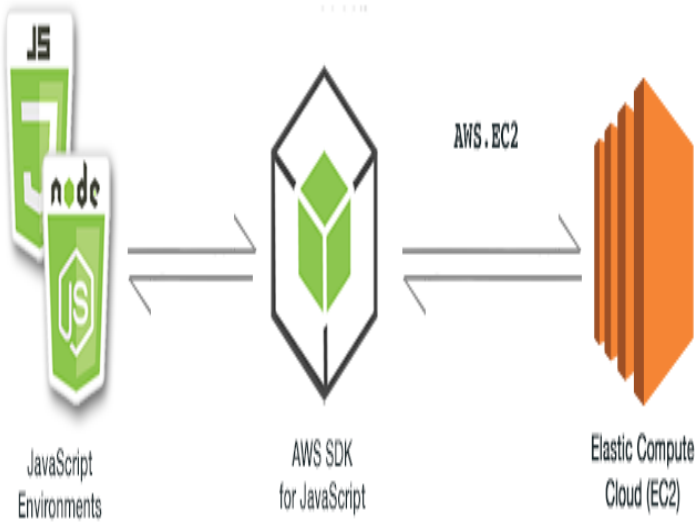

Figure 5.1: EC2 Instance

2) Saas: AWS Services

Figure 5.2: AWS Services

5.4 Technical details about each service

1) Amazon EC2:

- CPUs = 1 vCPU
- Ram = 1GB
- Storage = 20GB
- OS = ubuntu 18.04 LTS (virtualized)
- 100 Gbps of dedicated aggregate network bandwidth

2) Amazon S3

- Storage 5GB
- 10,000 Get Requests
- 1,000 Put Requests
- 10GB of data transfer in
- 10GB of data transfer

5.5 List of programs written or to be written

- Program to upload data on Server using GET Method
- Program to fetch data from Server using POST Method and fetch API using AJAX
- Programs to filter and fetch data for student's profile based on criteria selection factors such as GPA, Project and Work-Ex
- Programs to filter and fetch data for company's profile based on criteria selection factors such as Location, Designation and Salary

Chapter 6

Implementation of each cloud service

6.1 Details about each PaaS / SaaS from implementation point of view

1) Fault Tolerance

- Fault tolerance refers to the ability of a system to continue operating without interruption when one or more of its components fail.

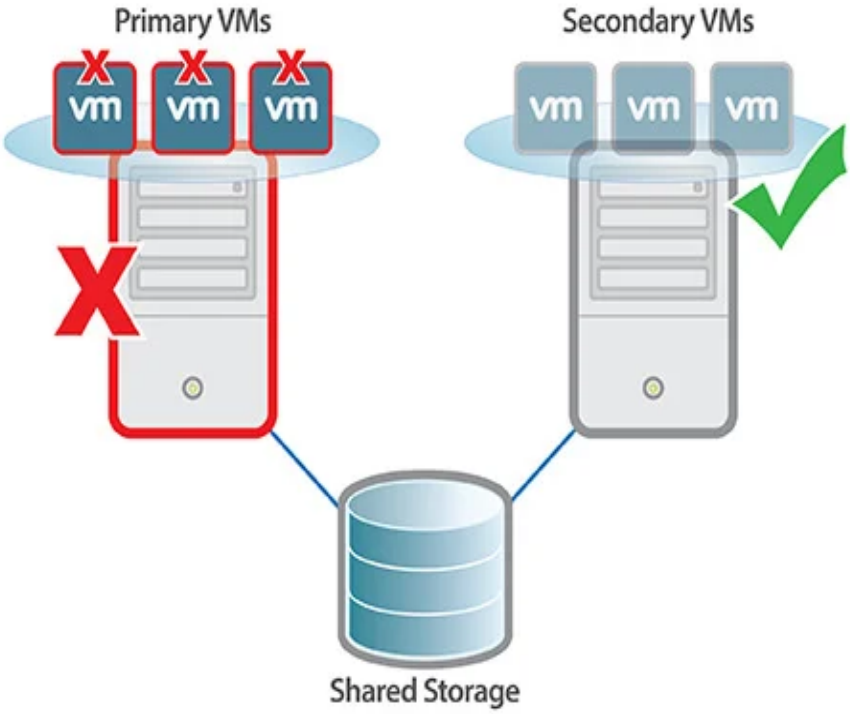

Figure 6.1: Database Deployment on AWS

2) EC2 Instance

- Creating Virtual Server using EC2 instance
- If one of the server fails, application does not get interrupted
- Instances are created from Amazon Machine Images (AMI).
- WINSCP and PUTTY were used to create Virtual Servers.
- Easily create Pre-configured servers

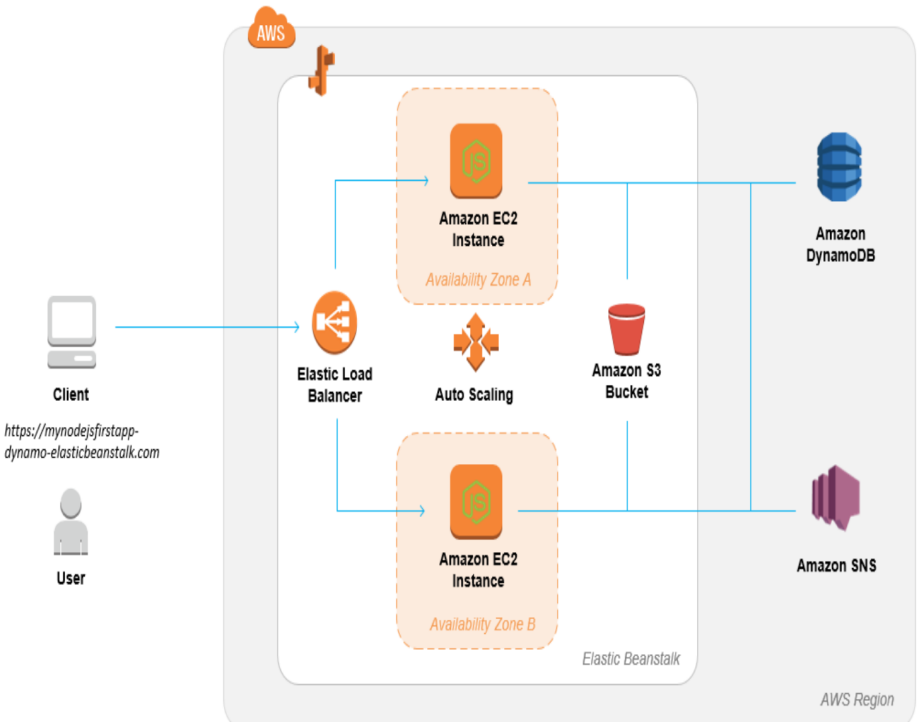

Figure 6.2: Database Deployment on AWS

Chapter 7

Testing of services and snap shots of results

1. API Testing using POSTMAN

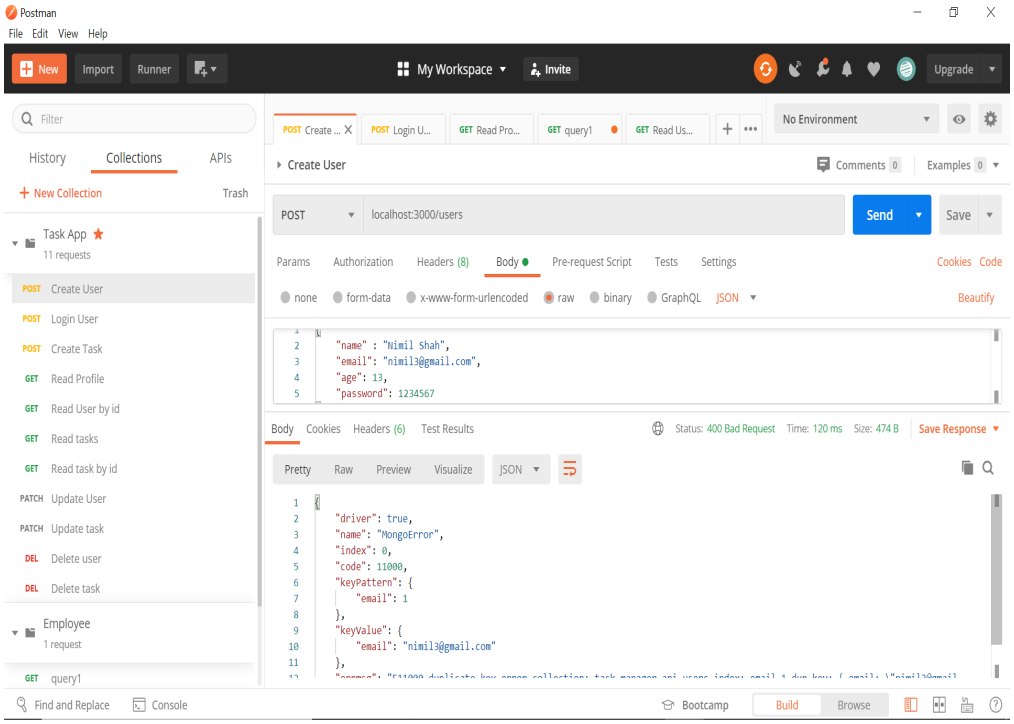

Figure 7.1: API Testing using POSTMAN

2. API Testing using POSTMAN

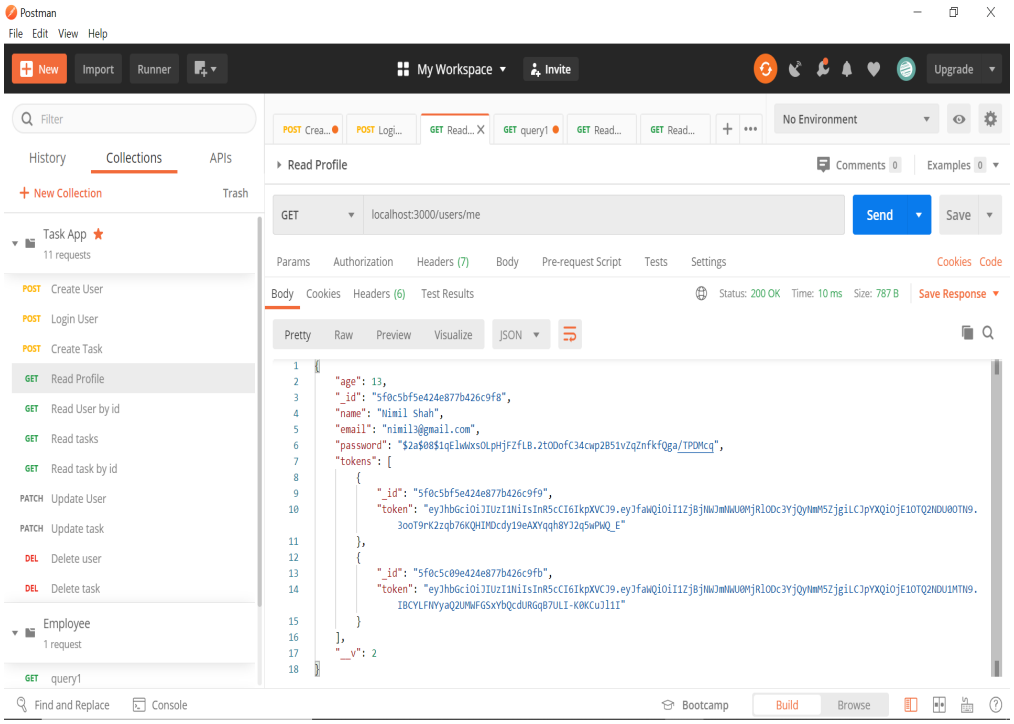

Figure 7.2: API Testing using POSTMAN

3. JSON Data using APIs

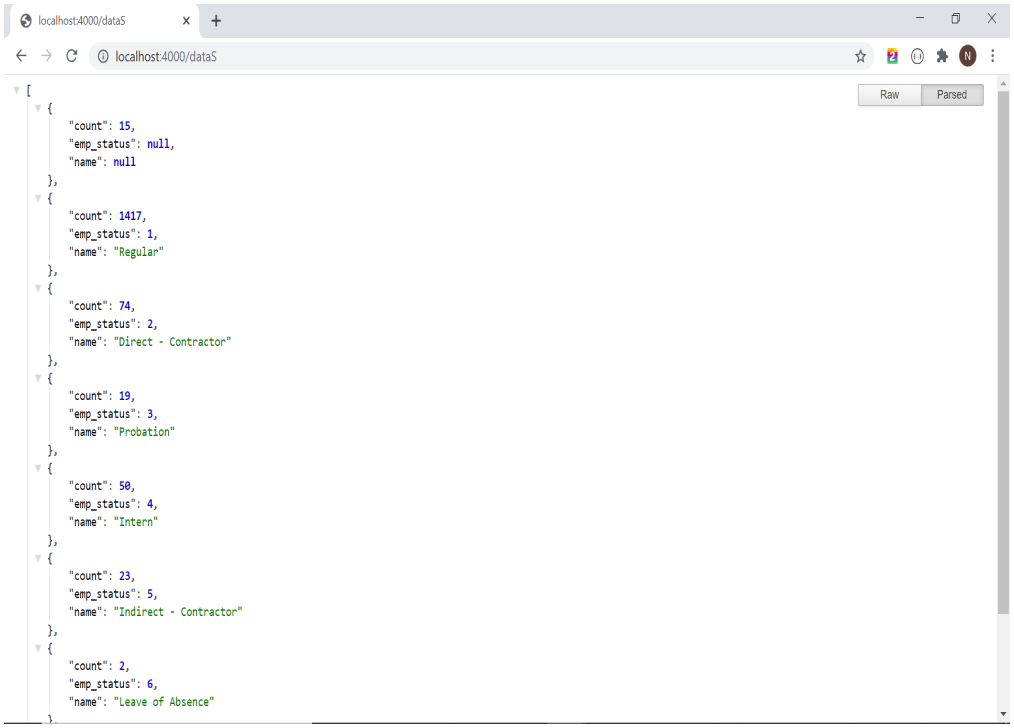

Figure 7.3: JSON Data using APIs

4. Back-end API

```
app.post('/d1',(req,res)=>{
  req.setTimeout(0)

  let user = req.body;
    console.log(user);
     id=user.id;
     console.log(id);
  connection.query("SELECT * FROM mytable WHERE CGPA_N >='"+ id +" ' " , function (error, rows, fields)
      if (error) throw error;
       res.send(rows);
       console.log(rows)
       });

   });
```

Figure 7.4: Back-end API

5. Front-end API

```
handleChange(event) {
  this.setState({selectValue: event.target.value});
}
handleSubmit(event) {
 alert('The selected option is : ' + this.state.selectValue);
console.log(this.state.selectValue)
  event.preventDefault();


  const requestOptions = {
    method: 'POST',
    headers: {'Accept': 'application/json', 'Content-type':'application/json',},
    body: JSON.stringify({ id: this.state.selectValue })
};

fetch('http://localhost:4000/d1', requestOptions)
    .then((response) => response.json())

    .then((data) => {
    this.setState({ todos: data })
    console.log(this.state.todos)

}
```

Figure 7.5: Front-end API

6. DynamoDB Connectivity

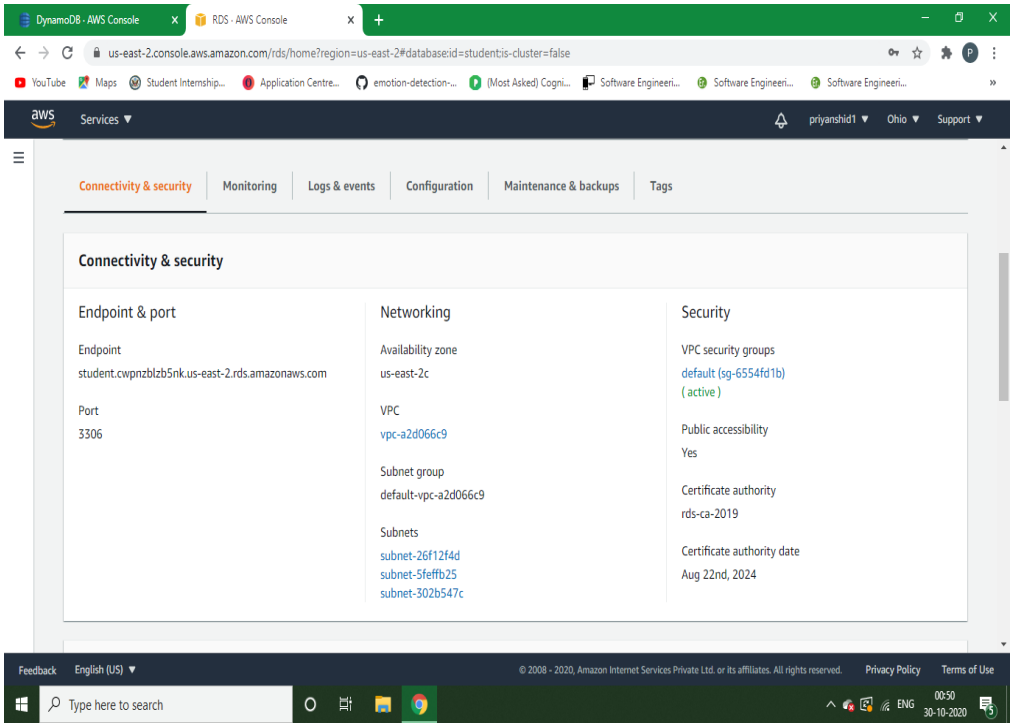

Figure 7.6: DynamoDB Connectivity

7. Test Connection with Local Database

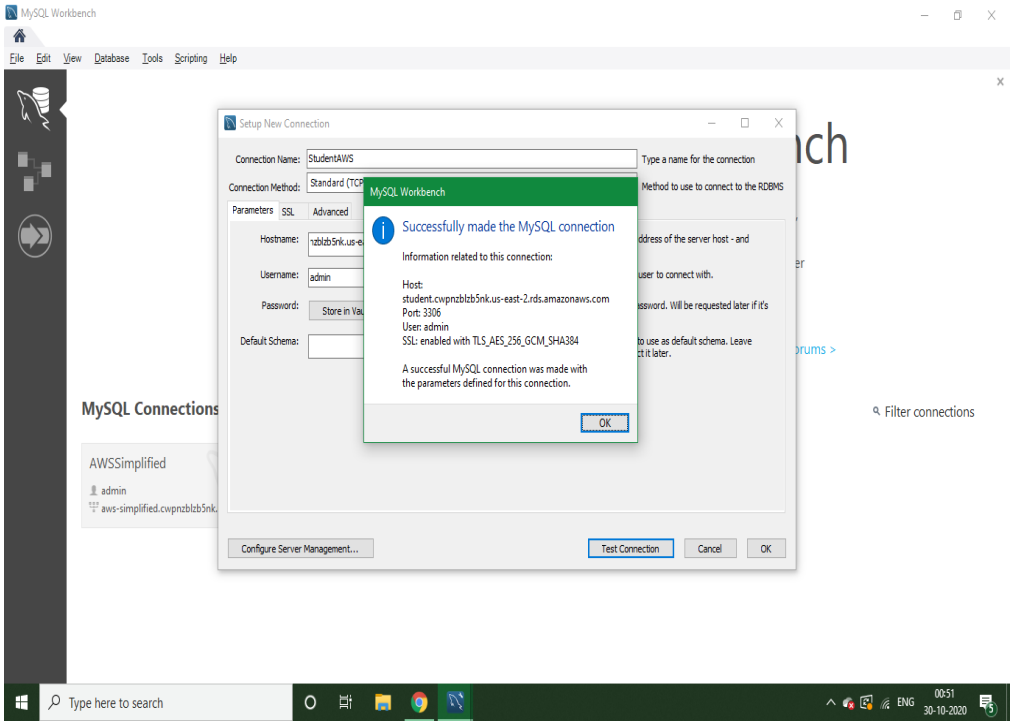

Figure 7.7: Test Connection with Local Database

8. Database Deployment on AWS

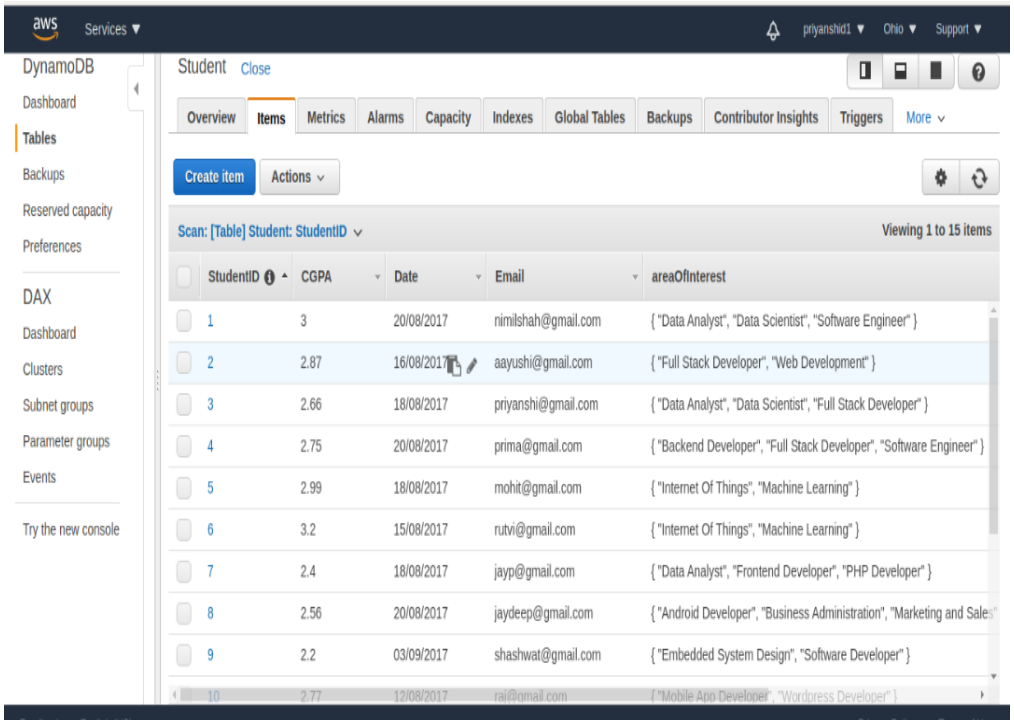

Figure 7.8: Database Deployment on AWS

9. Successful EC2 Instance Creation

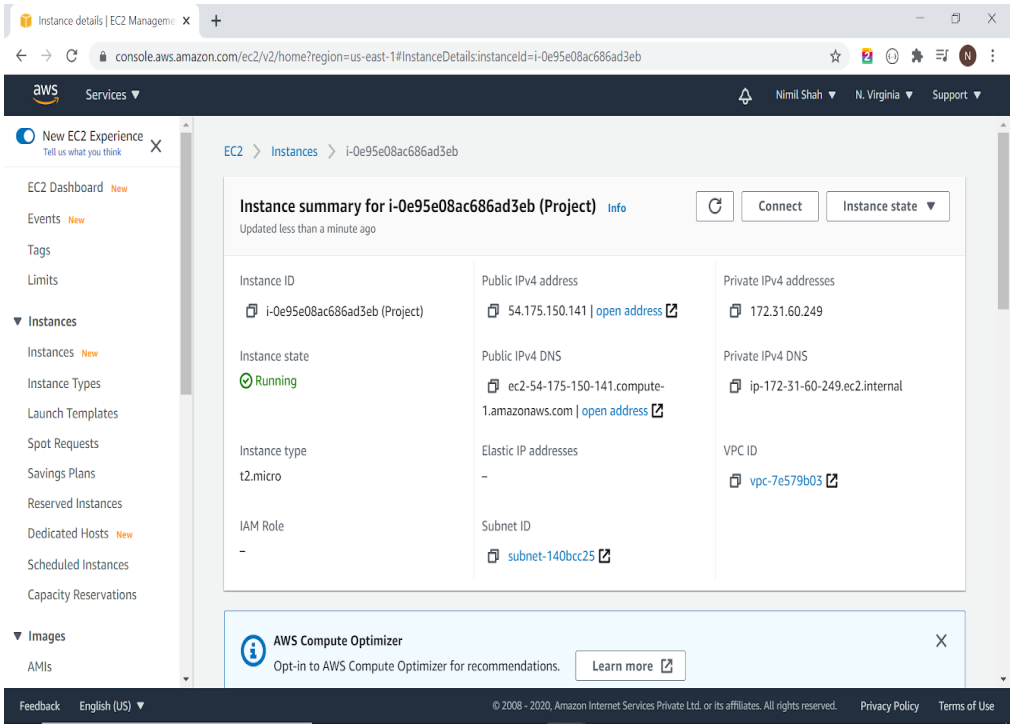

Figure 7.9: Successful EC2 Instance Creation

10. Creating Virtual Server using WinScp

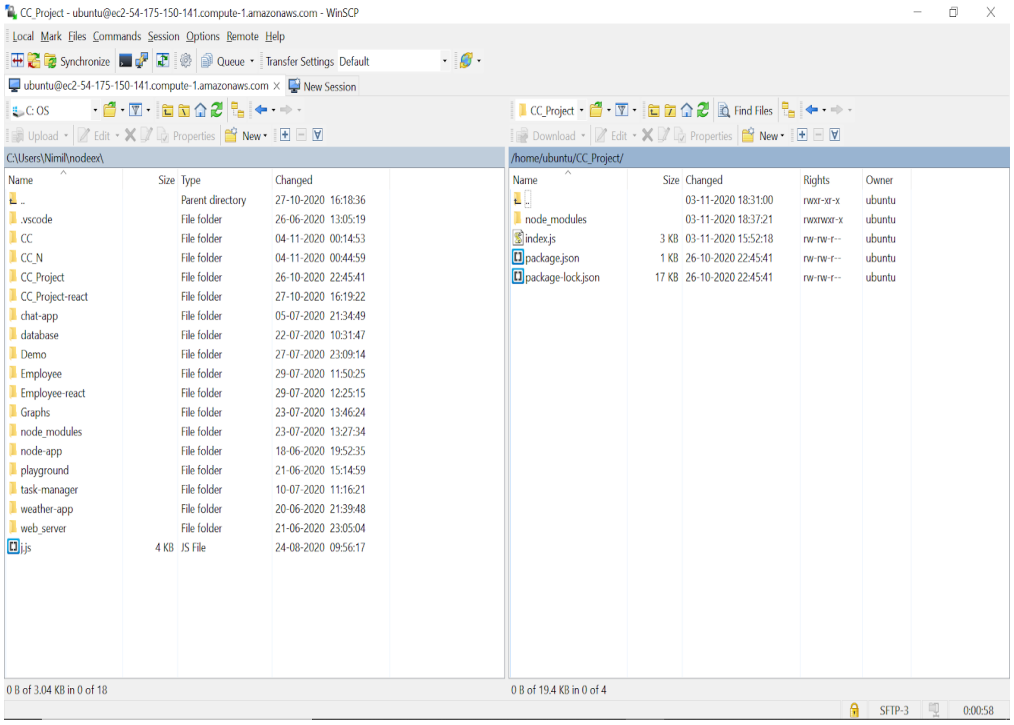

Figure 7.10: Creating Virtual Server using WinScp

11. NodeJS application Deployment using EC2

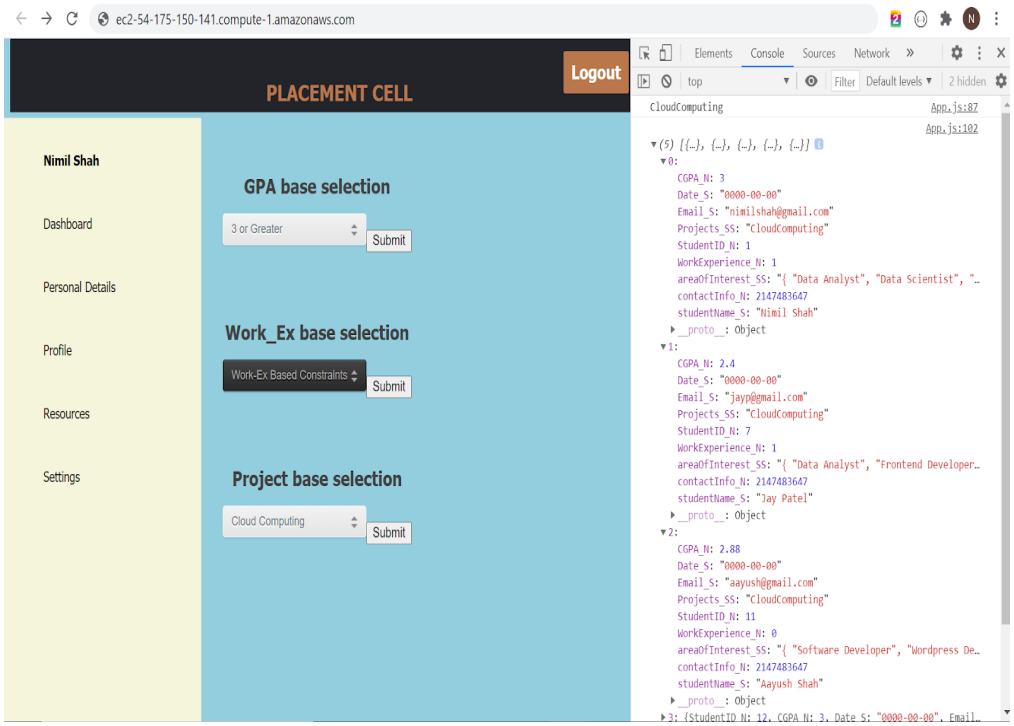

Figure 7.11: NodeJS application Deployment using EC2

12. Final Student Side Module

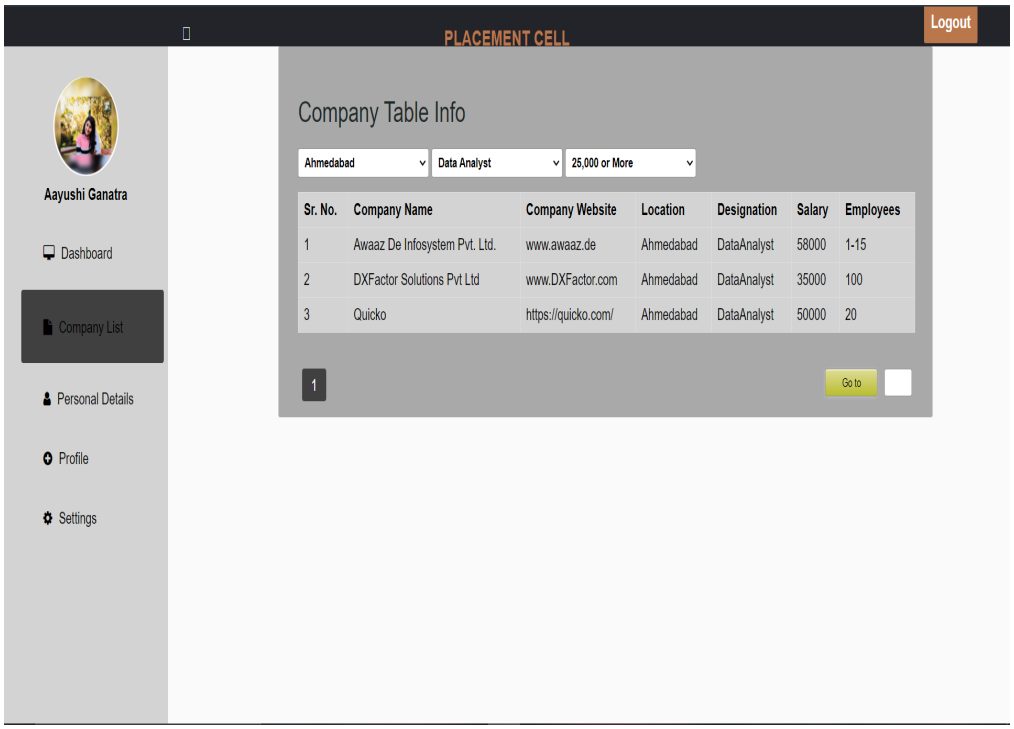

Figure 7.12: Final Student Side Module

13. Final Company Side Module

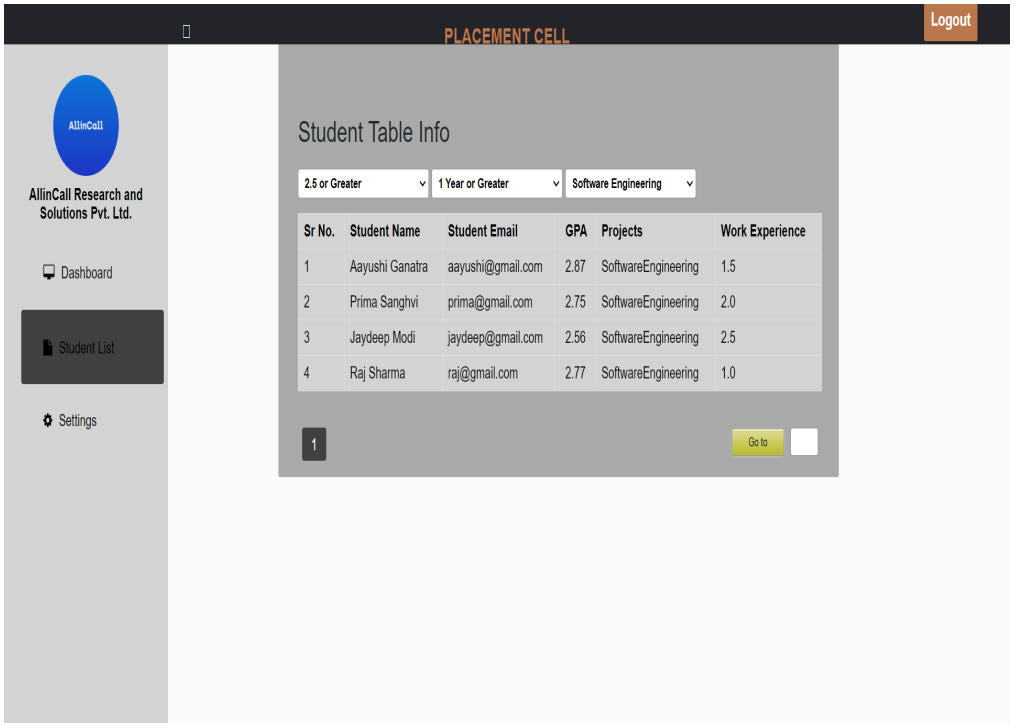

Figure 7.13: Final Company Side Module

Chapter 8

Conclusion

- In the existing system maximum work goes manually and it is error prone system, takes time for any changes in the system.
- The major drawback is the manual searching and updating of the student data and with no any alert or notification method available for giving the required information to the student.
- The proposed online training and placement management system gives the automation in all the processes that include registration, updating, and searching. It provides the detail solution to the existing system problem.
- During the development of the project the prime objective was to keep the hardware and software requirement as minimum as possible so that it is able to support maximum user base.
- Additionally, the searching procedure should be very strong for the company officer in order to search any student as quickly as possible, based on the given criterion.
- It can be concluded that the project is a highly efficient GUI based component. It can be easily integrated with various systems.

References

[1] Rishikesh Bane, Saman Khan, "Automated System for Placements" in May 2015 –International Journal for Innovative Research in Science Technology (IJIRST) Bandung – Bali, May 2015.

[2] Gaurav G., Anchal J., International School of Informatics and Management "Training and Placement Cell", www.iis.jaipur.org

[3] Mr. Nilesh R., Dr. Seema S., Prof. Kavita S., Vidyalankar Institute of Technology "An Interactive Online Training and Placement System".

[4] Anjali, Jeyalakshmi, Anbubala, Sri Mathura devi, Ranjini, "Web Based Placement Management System" in (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (2) , 2016

[5] Gunderloy, Jorden BPB Publications (2000) - "Mastering SQL Server"

[6] Luke Welling and Laura Thomson (5th Edition) - "PHP and MySQL Web Development"