

Get-Job Portal System

Prepared by:

Riya Karmakar

Index

- **Introduction**
- **Project prerequisites**
- **Project software/hardware**
- **Create problem solving approach**
- **Version Control**
- **Overall problems**
- **Features of the activity UML diagram of Get-job portal**
- **Class diagram**
- **Use-case diagram**
- **ER diagram**
- **Web development:**
 1. **Development**
 2. **Deployment**
- **Future scope**
- **References**
- **Conclusion**

GET-JOB PORTAL SYSTEM

INTRODUCTION: - Whether entering the employment for the first time or re-entering after a break or switching career, job search is a challenging task. But how about tools/applications, making this tedious process look friendly, systematic, and easy to reach out, to employees or to students or to the jobseekers. Searching and landing up with a dream job is a tedious process for a job seeker. This project is aimed at making such challenges much easier despite the geographic location of the either the job seeker or the company. “Get-job” is an online web application which is a job search portal. This portal enables candidate looking for jobs to register themselves into this portal and look up for different jobs according to their qualifications and apply for those jobs conveniently. Jobseeker can also update their details entered during registration as well as their skills. On the other hand, Employers can also register to this portal and post their jobs which would enable them to find suitable candidates for their vacant positions. With this type of job search portals, the trend of paper resumes get replaced by online resumes. Candidates and employers are just a few clicks away to get connected. Another advantage is once the candidate is registered and applies for a job, his/her information stays with the company database for both the present and future use for available positions.

PROJECT PREREQUISITES: -

Need some prior knowledge about -

- What types of Python code distribution exist?
- Why building a web application can be a good choice?
- What is Cloud Computing (AWS/AZURE/GCP)?
- What is Django Framework?
- Version Control Technology (Git/CICD pipeline(code commit))

PROJECT HARDWARE/SOFTWARE: -

- Operating system – Windows/Open source (Linux/Ubuntu).
- Programming language – Python, HTML, CSS, JS.
- Database – MySQL, RDS.
- Cloud - AWS

- Framework – Django, Bootstrap.
- GitHub/Gitlab
- UI design

CREATE PROBLEM SOLVING APPROACH: -

- Launch VMs/Instance with the help of cloud using compute service (EC2/Lambda) and deploy the client service web application on it.
- Storing
 - a. S3 bucket
 - b. Databases like MySQL, RDS
- Export MySQL database into S3 bucket and analyze the data using AWS Athena and Glue.
- Create blue green deployment with the help of Elastic Beanstalk.

VERSION CONTROL: -

- Set up a Git repository (use public GIT) for version-control.
- Enable S3/CloudFront bucket versioning for version-control.

OVERALL PROBLEMS: -

With the help of this web-application we can do the following operations:

Company:

- Company can register and create an account.
- Company can login to the job portal.
- Company can edit their profile.
- Company can post jobs.
- Company can see posted jobs of their company.
- Company can edit their job.
- Company can delete their job.
- Company can see candidates who applied for jobs in their company.
- Company can change their password.

Jobseeker:

- Jobseeker can register and create an account.

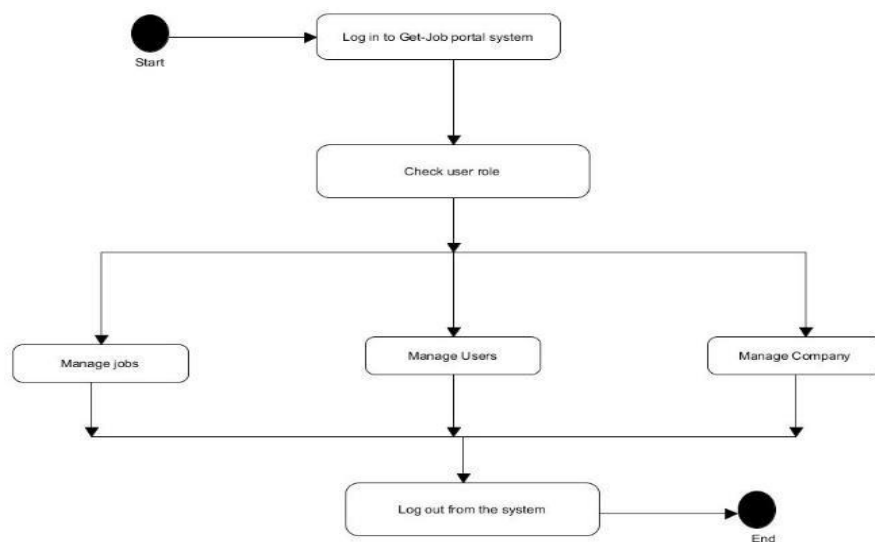
- Jobseeker can login to the job portal.
- Jobseeker can edit his/her profile.
- Jobseeker can see latest jobs.
- Jobseeker can apply for any job.
- Jobseeker can change password.

Admin:

- Admin can login to the job portal
- Admin can see the total number of registered companies and jobseekers.
- Admin can see the status(pending/accepted/rejected) of the companies.
- Admin can change the status(accept/reject) of the companies.
- Admin can delete any company profile.
- Admin can delete any jobseeker's profile.
- Admin can change password.

FEATURES OF THE ACTIVITY UML DIAGRAM OF GET-JOB PORTAL: -

Activity Diagram is also one important UML diagram that gives the flow of execution of the system. It's a pictorial representation of the different activities of a Get-job portal system. A function performed by the system can be called an activity of the system. It shows the flow of Users, Jobs, Companies, and user profile:



Get-job portal activity diagram

CLASS-DIAGRAM: -

GET-JOB PORTAL SYSTEM CLASS DIAGRAM: -

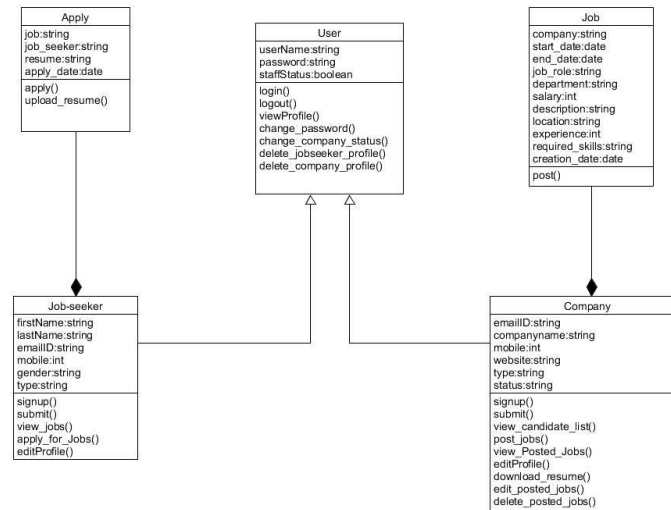
- User class- manages all the operations of Users.
- Job-seeker class-manages all the operations of jobseekers.
- Company class-manages all the operations of Companies.
- Job class-manages all the operations of Jobs.
- Apply class-manages all the operations of Apply.

CLASSES AND THEIR ATTRIBUTES GET-JOB PORTAL SYSTEM CLASS DIAGRAM: -

- User class attributes- userName, password, staffStatus.
- Job-seekerclass attributes- firstName,lastName, password, mobile, emailID, gender, type.
- Company class attributes-companyName, password, emailID, website, mobile, type, status.
- Job class attributes-company, start_date, end_date, job_role, department, salary, description, location, experience, required_skills, creation_date.
- Apply class attributes-job, job_seeker,resume,apply_date.

CLASSES AND THEIR METHODS GET-JOB PORTAL SYSTEM CLASS DIAGRAM: -

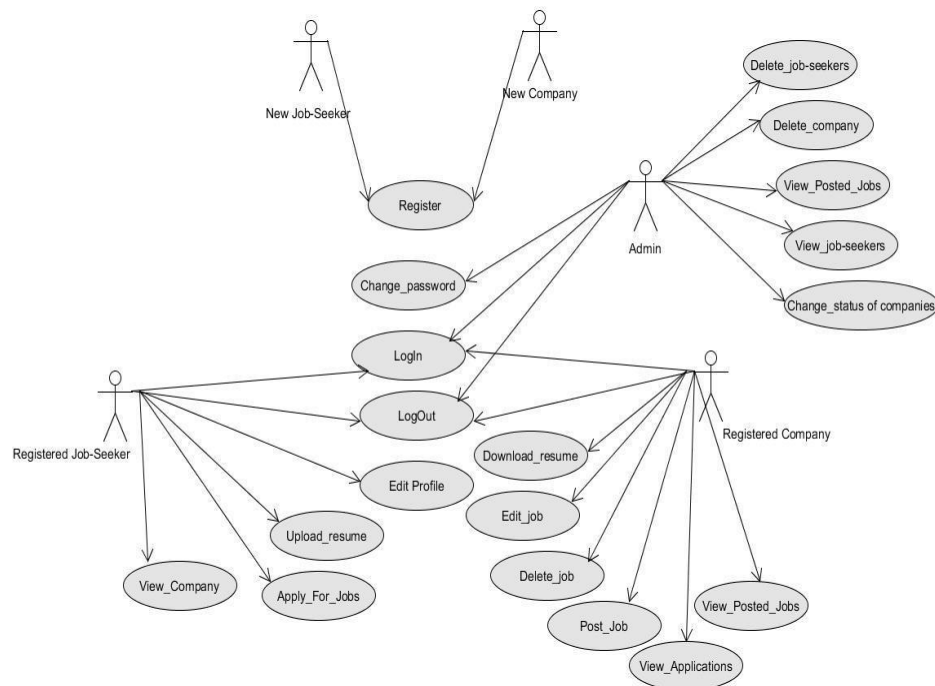
- User class methods- login (), logout(), viewprofile(), change_password, change_company_status(), delete_jobseeker_profile(), delete_company_profile()
- Job-seeker class methods-signUp(), submit(),edit_profile(), view_jobs(), apply_for_jobs().
- Company class methods- signUp(), submit(), view_candidate_list(), post_jobs(), view_posted_jobs(),edit_profile(), download_resume(), edit_posted_jobs(), delete_posted_jobs.
- Job class methods – post ()
- Apply class methods – apply (), upload_resume().



Class diagram

USE-CASE DIAGRAM: -

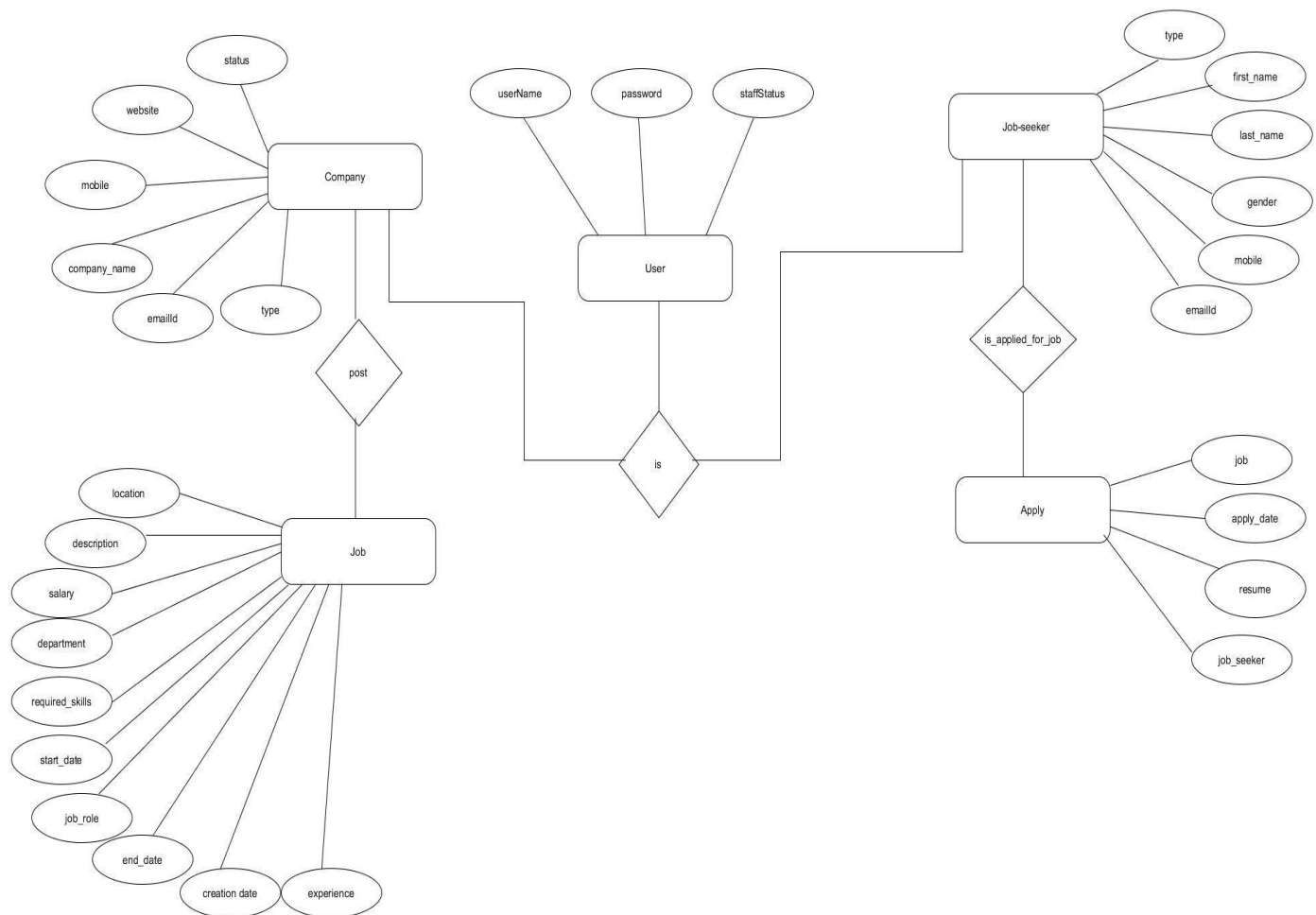
While understanding the static nature of a system is insufficient, Use-case diagrams helps to give the dynamic view of the system. Use -case diagrammodels the system and subsystems of an application. Use-case diagram is a clear visualization of the usage of that application.



Use-case diagram

GET-JOB PORTAL SYSTEM DATABASE ER DIAGRAM: -

This entity relationship diagram gives a pictorial representation of all the database tables and the relation between the entities. It also shows the cardinality i.e the many to one or one to one or one to many relationships amongst the tables. This is the first step in designing the database. All the idea of requirements and specification details about the different entities in the database are conceived in the beginning and then transformed into a diagram. This step takes time but once finalized, a good, strong, and robust database is a cake walk.



Entity Relationship Diagram

WEB APPLICATION DEVELOPMENT:

Phase-1: Development

After discussion about the problem statement, we have started to develop this web application. We have made this web application using Python Django framework.

Django is basically a high-level Python web application framework that enables the rapid development of web applications. It achieves so with pragmatic, much cleaner design and is also easy to use (in comparison of other frameworks) thus is very popular among web developers. It is a backend framework used to resolve problems of connectivity with databases, other server problems, SEO solutions, etc. So that a web developer need not write the same code for the similar modules (like database connection, admin interface) for each website. All the functionality comes in the Django framework in the form of web applications. You just have to import those applications according to your need and thus you can concentrate more on the unique application of your website rather than dealing with all these backend problems.

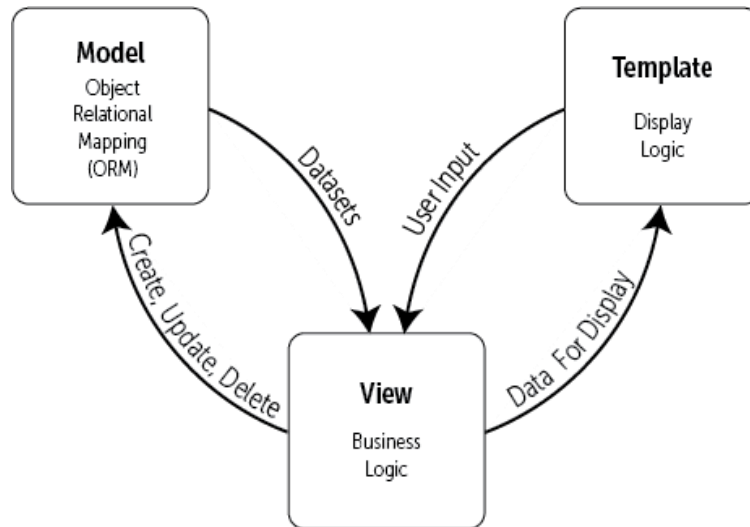
For every website/web application on the Internet, there are 3 main components or code partitions: Input Logic, Business Logic, and UI Logic. These partitions of code have specific tasks to achieve, the input logic is the dataset and how the data gets to organize in the database. It just takes that input and sends it to the database in the desired format. The Business logic is the main controller which handles the output from the server in the HTML or desired format. The UI Logic as the name suggests are the HTML, CSS, and JavaScript pages.

Django is based on **MVT (Model-View-Template)** architecture.

Model: The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres).

View: The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/JavaScript files.

Template: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.



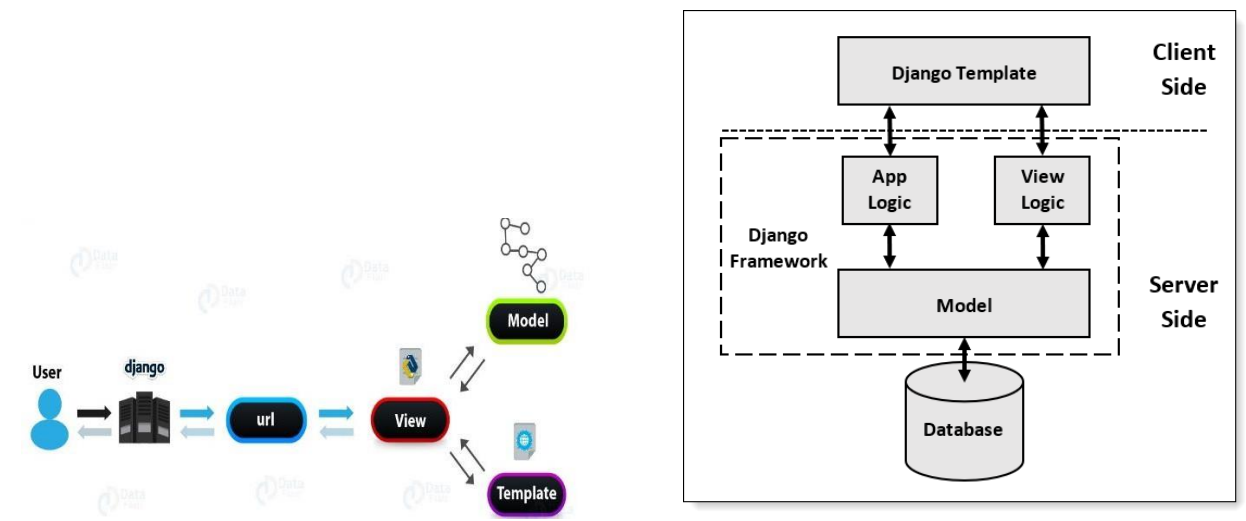
MTV architecture

A Django Project when initialized contains basic files by default such as manage.py, settings.py, etc.

- **manage.py**- This file is used to interact with your project via the command line (start the server, sync the database... etc.).
- **_init_.py** – It is a python package. It is invoked when the package or a module in the package is imported. We usually use this to execute package initialization code, for example for the initialization of package-level data.
- **settings.py** – As the name indicates it contains all the website settings. In this file, we register any applications we create, the location of our static files, database configuration details, etc.
- **urls.py** – In this file, we store all links of the project and functions to call.
- **wsgi.py** – This file is used in deploying the project in WSGI. It is used to help your Django application communicate with the webserver.

The **migrations** folder is where Django stores migrations, or changes to your database. There's nothing in here you need to worry about right now.

- **admin.py** - is where you register your app's models with the Django admin application.
- **apps.py** - is a configuration file common to all Django apps
- **models.py** - is the module containing the models for your app.
- **tests.py** - contains test procedures that run when testing your app.
- **views.py** - is the module containing the views for your app.



Flow of Django framework

In models.py file we have mentioned the classes (like job_seeker, company, job etc.) for our database to maintain our job portal. And in there we have also mentioned the attributes of that particular class and the datatype of that attribute. We have to register those classes on admin.py to create the database properly. In templates we have created our HTML files (like index.html, admin_login.html, user_login.html etc.). In views.py we have defined our business logic for this web application. And in the urls.py we have mentioned the path of every web page.

In our web application there is three type of user we have mentioned. One is Jobseeker who is looking for jobs, another is Company who will post their job in this portal and also select the candidates, and another is Admin user who will manage jobseeker and company.

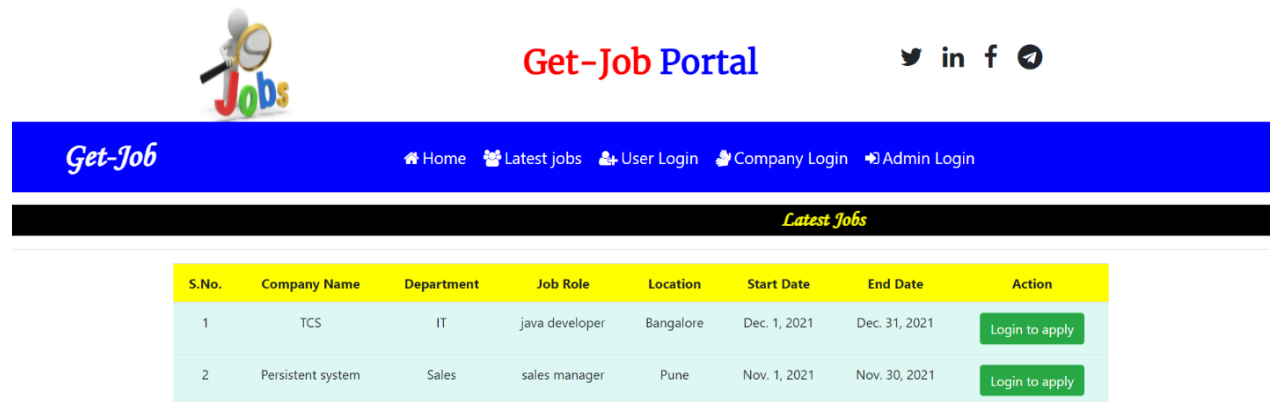
Home page:

This is our Get-job portal home page. There are five options for home, latest jobs to see the latest jobs in this portal, user login option for jobseeker login, company login and admin login. And in footer section there is contact details for this portal.



Latest Jobs:

In latest jobs section all the latest jobs of this portal are available. When a jobseeker wants to apply for that job, he/she will click the **login to apply button**. After clicking that button the jobseeker will go to the jobseeker login page directly.

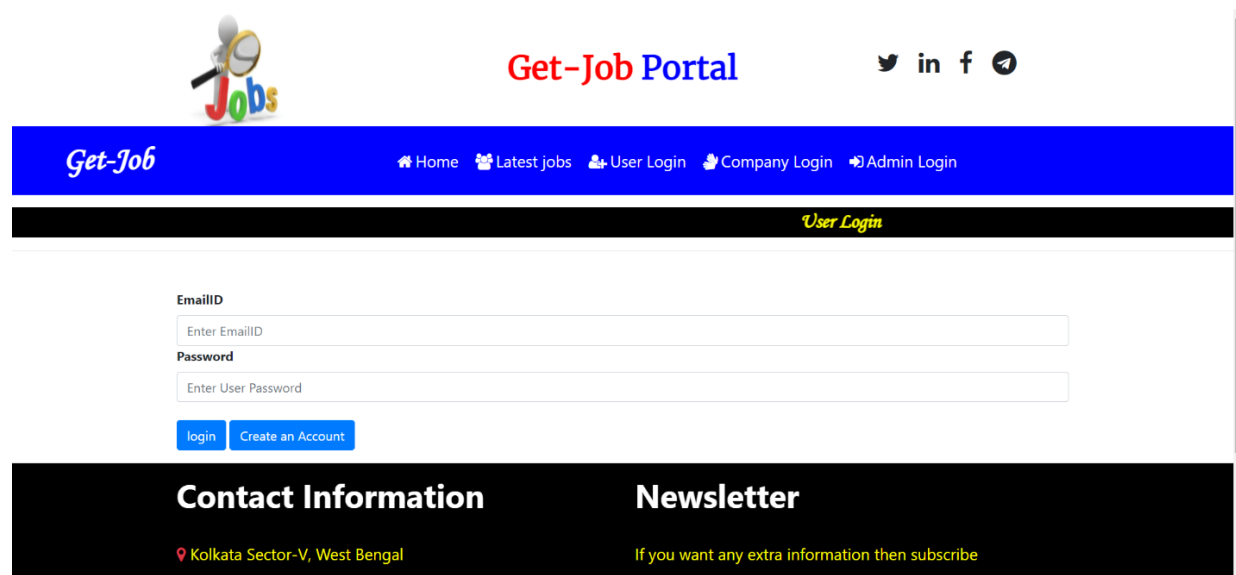


The screenshot shows the 'Get-Job Portal' website. The header includes the logo, the text 'Get-Job Portal', and social media icons for Twitter, LinkedIn, Facebook, and Telegram. Below the header is a blue navigation bar with the 'Get-Job' logo and links for Home, Latest jobs, User Login, Company Login, and Admin Login. The main content area is titled 'Latest Jobs' and contains a table with job listings.

S.No.	Company Name	Department	Job Role	Location	Start Date	End Date	Action
1	TCS	IT	java developer	Bangalore	Dec. 1, 2021	Dec. 31, 2021	Login to apply
2	Persistent system	Sales	sales manager	Pune	Nov. 1, 2021	Nov. 30, 2021	Login to apply

Job-seeker login/signup:

Here the jobseeker can login for applying jobs if they already have an account in this portal. If the jobseeker doesn't have any account in this portal, he/she has to create an account by submitting his/her name, email address, contact number, gender and password.



The screenshot shows the 'Get-Job Portal' website with the 'User Login' section highlighted. The header and navigation bar are the same as in the previous screenshot. The main content area is titled 'User Login' and contains a login form with fields for EmailID and Password, and buttons for login and Create an Account. Below the login form is a black footer section with 'Contact Information' and 'Newsletter' links.

Contact Information
Kolkata Sector-V, West Bengal

Newsletter
If you want any extra information then subscribe

Get-Job

[Home](#) [Latest jobs](#) [User Login](#) [Company Login](#) [Admin Login](#)

User Sign Up

First Name

Enter your first name

Last Name

Enter your last name

Contact Number

Enter your Contact Number

Email ID

Enter your Email ID

Password

Enter your Password

Confirm Password

Confirm your Password

Gender

☐ Male ☐ Female

Sign Up

Job-seeker home page:

After logging in this portal jobseekers can see their profile and edit their profile and there will be another navigation bar. There will be some options like job list, change password and logout.

Get-Job

[Home](#) [Job List](#) [Change Password](#) [Logout](#) Rohit Raj

Job-Seeker Profile

First Name

Rohit

Last Name

Raj

Contact Number

254862384

Email ID

rohit@gmail.com


Gender

Male

Edit

Job list for Jobseekers:

In here jobseeker can see jobs which is available in this portal, can see the job details and can apply for that job by submitting their resume.

Get-Job Portal🐦 in f

Get-Job🏠 Home 👤 Job List 🔑 Change Password 🚪 Logout Rohit Raj

Latest Job list

S.No.	Company Name	Department	Job Role	Location	Start Date	End Date	Action
1	TCS	IT	java developer	Bangalore	Dec. 1, 2021	Dec. 31, 2021	Apply
2	Persistent system	Sales	sales manager	Pune	Nov. 1, 2021	Nov. 30, 2021	Applied

Job details:

Get-Job🏠 Home 👤 Job List 🔑 Change Password 🚪 Logout Rohit Raj

Job Details

java developer

[TCS\(View all jobs\)](#)

📍 Bangalore

📅 Nov. 23, 2021

₹ 800000.0 ₹

Job Description : job vacancy 25

Experience : 2 years


Skills Required : Java,DS,SpringBoot





Start Date : Nov. 21, 2021


End Date : Dec. 31, 2021





[Apply for this job](#)

Resume upload:

Get-Job Portal





 Home  Job List  Change Password  Logout Rohit Raj

Apply For Job

Upload Resume


Choose File





No file chosen


Submit





Change password:

If the jobseeker wants to change password, then they can do it here. It is also similar for company.

Get-Job Portal





 Home  Job List  Change Password  Logout Rohit Raj

Change Password

Current Password

New Password

Confirm New Password

Change

Company signup:

By submitting company name, contact number, email, website and password any new company can sign up in this portal, but they have to wait for admin approval to login in this portal.

The screenshot shows the 'Company Signup' page of the 'Get-Job' portal. The header is blue with the 'Get-Job' logo and navigation links: Home, Latest jobs, User Login, Company Login, and Admin Login. Below the header is a black bar with the text 'Company Sign Up' in yellow. The main content area is white and contains a form with the following fields: 'Company Name' (placeholder: Enter your Company name), 'Contact Number' (placeholder: Enter your Contact Number), 'Email ID' (placeholder: Enter your Email ID), 'Website' (placeholder: Enter your Company Website), 'Password' (placeholder: Enter your Password), and 'Confirm Password' (placeholder: Confirm your Password). A blue 'Sign-Up' button is located at the bottom of the form.

Company home page:

Here Company can see the company details and also, they can edit the details. And there is another navigation bar which contains Add job, job list, candidates applied, change password and logout.

The screenshot shows the 'Company Profile' page of the 'Get-Job' portal. The header is blue with the 'Get-Job' logo and navigation links: Home, Add Job, Job List, Candidates Applied, Change Password, Logout, and persistent@gmail.com. Below the header is a black bar with the text 'Company Profile' in yellow. The main content area is white and contains a form with the following fields: 'Company Name' (value: Persistent system), 'Contact Number' (value: 367632169890), 'Email ID' (value: persistent@gmail.com), and 'Website' (value: www.persistent.com). A blue 'Edit' button is located at the bottom of the form.

Post jobs:

Company can post any job by submitting the details of that particular job.

Job role

Experience(In Years)

Required Skills

Salary

Location


Start Date

End Date





Description

Company job list:

In this section company can see the list of jobs which company posted till date.



Get-Job Portal



Get-Job


[Home](#) [Add Job](#) [Job List](#) [Candidates Applied](#) [Change Password](#) [Logout](#) [persistent@gmail.com](#)

[View Jobs](#)

S.No.	Department	Job Role	Action
1	Sales	sales manager	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Applied candidates list:

In this section company can see the candidates who applied for job, download candidates resume and also can delete their application.



Get-Job Portal

🐦 in f 📍

Get-Job


🏠 Home ➕ Add Job 📋 Job List 📋 Candidates Applied 🔑 Change Password 🚪 Logout tcs@email.com

Applied Candidate List

S.No.	Full Name	EmailID	Contact No	Job Role	Company Name	Apply Date	Resume	Action
1	maya karmakar	maya@gmail.com	8348600306	java developer	TCS	Nov. 29, 2021	Download Resume	Delete
2	Rohit Raj	rohit@gmail.com	254862384	java developer	TCS	Nov. 29, 2021	Download Resume	Delete

Admin login:

By clicking admin credentials admin can login in this portal.



Get-Job Portal

🐦 in f 📍

Get-Job

🏠 Home 📋 Latest jobs 👤 User Login 🏢 Company Login 🚪 Admin Login

Admin Login

UserName

Enter User Name

Password

Enter User Password

Login

Admin home page:

This admin home page where can see how many jobseekers and companies have registered in this portal. In navigation bar there is many options like company (accepted, pending ,rejected and all companies) ,view jobseekers ,change password and logout.

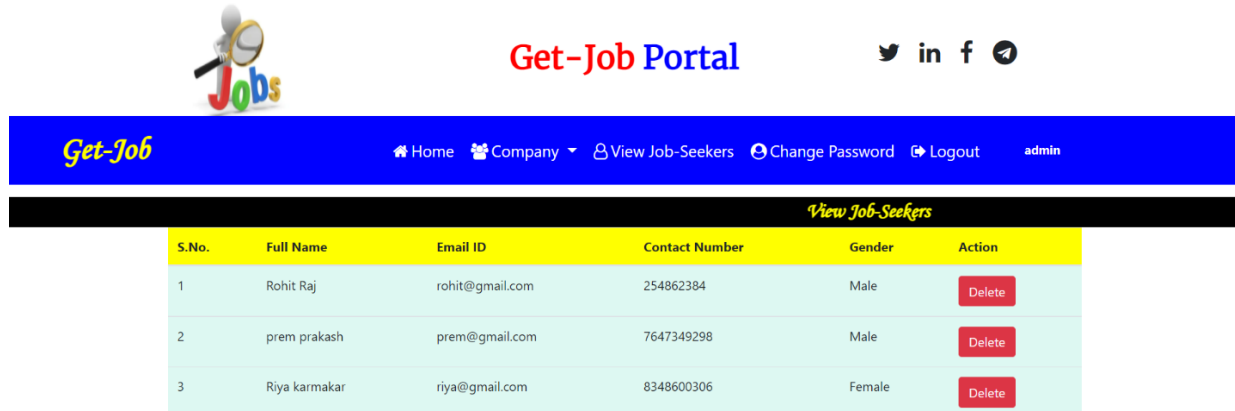


The screenshot shows the Admin Home page of the Get-Job Portal. The header includes the logo, the text 'Get-Job Portal', and social media icons. The navigation bar has links for Home, Company, View Job-Seekers, Change Password, Logout, and the user 'admin'. The main content area displays two statistics: 'Total number of Companies' and 'Total number of Job-Seekers', both with a value of 3.

Category	Count
Total number of Companies	3
Total number of Job-Seekers	3

Job-seekers list:

Here admin can delete any jobseeker.

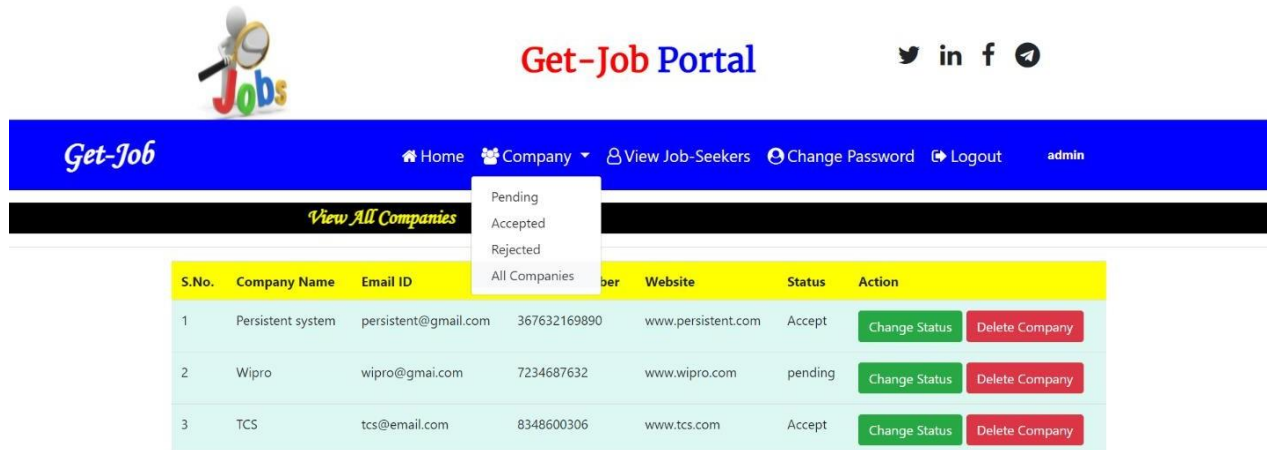


The screenshot shows the View Job-Seekers page of the Get-Job Portal. The header and navigation bar are identical to the Admin Home page. The main content area displays a table of job seekers with columns for S.No., Full Name, Email ID, Contact Number, Gender, and Action. Each row has a 'Delete' button in the Action column.

S.No.	Full Name	Email ID	Contact Number	Gender	Action
1	Rohit Raj	rohit@gmail.com	254862384	Male	Delete
2	prem prakash	prem@gmail.com	7647349298	Male	Delete
3	Riya karmakar	riya@gmail.com	8348600306	Female	Delete

Company list:

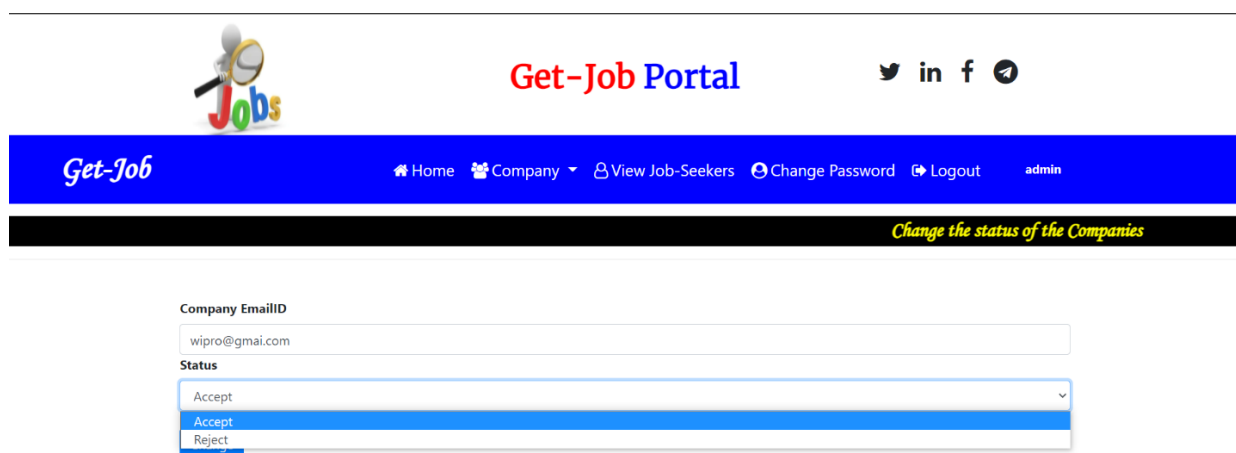
Here admin can see all the companies who registered in this portal. Admin can change the status of company and also can delete any company.



The screenshot shows the 'Get-Job Portal' interface. At the top, there is a logo with a magnifying glass over the word 'Jobs', the text 'Get-Job Portal', and social media icons for Twitter, LinkedIn, Facebook, and Telegram. Below this is a blue navigation bar with the 'Get-Job' logo and links for Home, Company (with a dropdown), View Job-Seekers, Change Password, Logout, and a user profile for 'admin'. A black bar below the navigation contains the text 'View All Companies'. A dropdown menu is open under the 'Company' link, showing options: Pending, Accepted, Rejected, and All Companies. The main content area displays a table of registered companies.

S.No.	Company Name	Email ID	Member ID	Website	Status	Action
1	Persistent system	persistent@gmail.com	367632169890	www.persistent.com	Accept	<button>Change Status</button> <button>Delete Company</button>
2	Wipro	wipro@gmail.com	7234687632	www.wipro.com	pending	<button>Change Status</button> <button>Delete Company</button>
3	TCS	tcs@email.com	8348600306	www.tcs.com	Accept	<button>Change Status</button> <button>Delete Company</button>

Change status:



The screenshot shows the 'Get-Job Portal' interface for changing company status. The top navigation bar and logo are identical to the previous screenshot. Below the navigation bar, a black bar contains the text 'Change the status of the Companies'. The main content area features a form with two fields: 'Company EmailID' with a text input containing 'wipro@gmail.com', and 'Status' with a dropdown menu. The dropdown menu is open, showing three options: 'Accept', 'Accept', and 'Reject'.

Company EmailID

wipro@gmail.com

Status

Accept

Accept

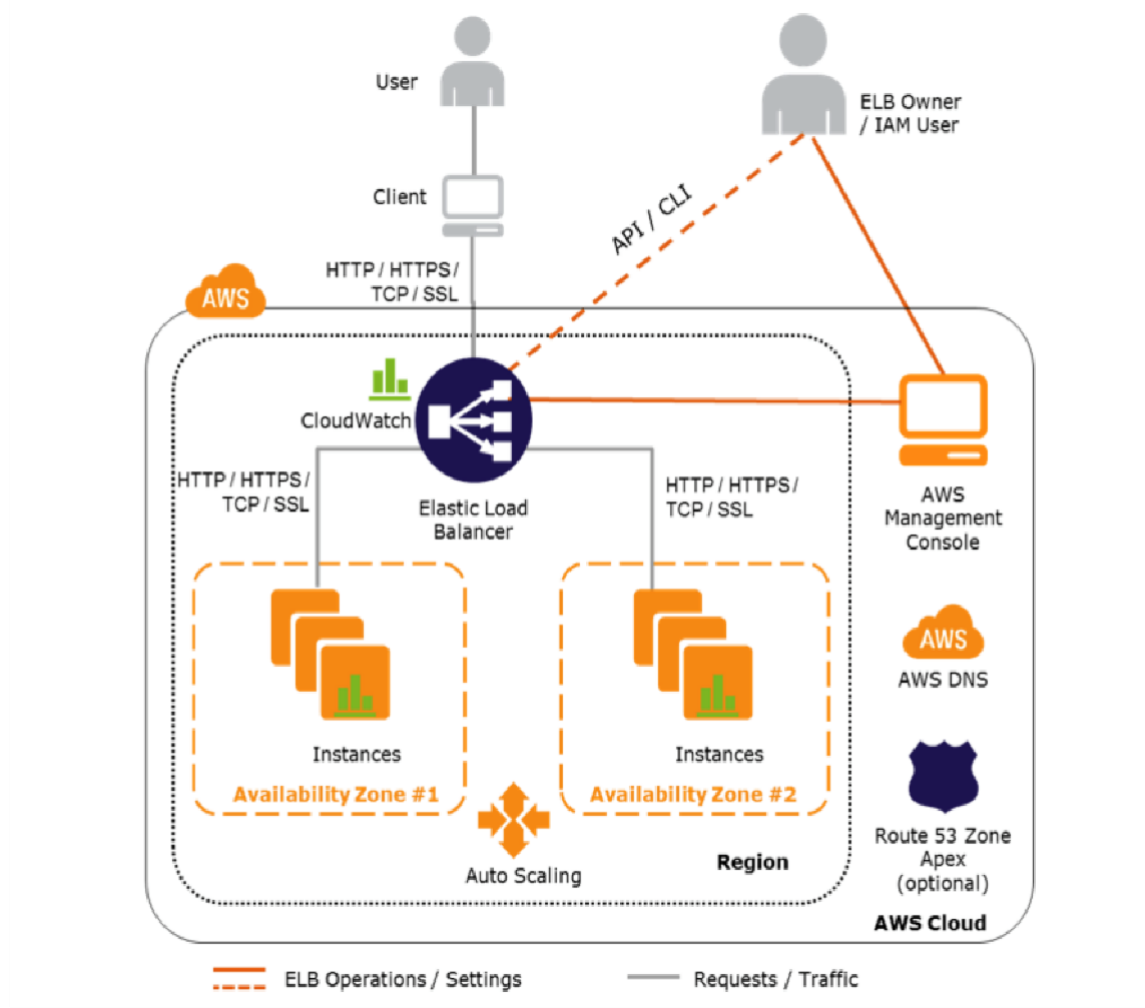
Reject

Phase-2: Deployment

For deployment of our web application, we have used **AWS (Amazon Web Service)** cloud computing service. At first, we create account on AWS. At first, we have created **IAM (Identity Access Management)** user by attaching Adminfullaccess policy. Then we have downloaded the IAM user credentials and with the help of those credentials we logged in AWS as IAM user. Then we have started our deployment procedure.

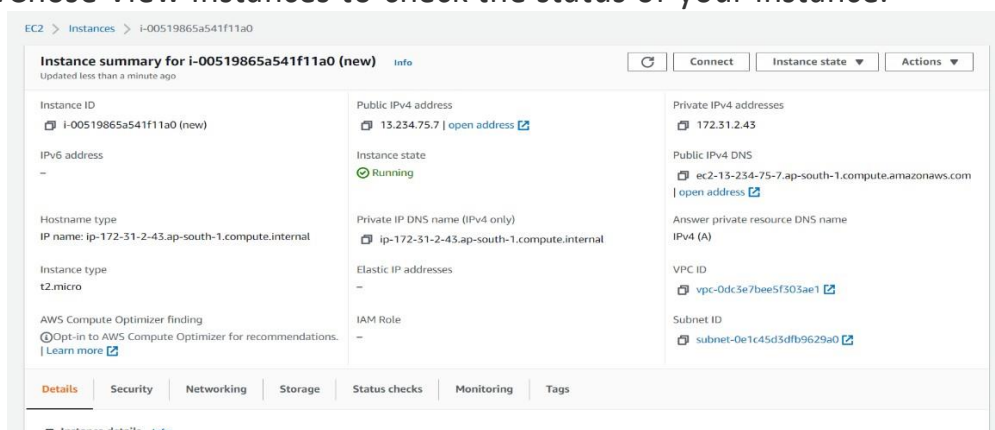
EC2: -

We have used EC2 services following this diagram:



At first, we have used **EC2(Elastic cloud computing)** services. With the following steps we have created out instance.

1. Opened the EC2 console.
Note: Be sure to select the AWS region that you want to launch the instance in.
2. From side bar we have chosen Instances then click launch instances.
3. Find the **AMI (Amazon Machine Image)** that we wanted to use to launch a new instance. To begin, open the menu next to the search bar, and then choose one of the following AMIs. We have chosen **Ubuntu** for launching our instance.
4. Selected the AMI, and then chose Launch.
5. Chose an instance type (t2. micro), and then chose Next: Configure Instance Details.
6. Selected Next: Add Storage. We have kept it as default storage.
7. Selected Next: Add Tags. We have added tag.
8. Selected Next: Configure **Security Group**. We have made a new security group with such inbound rules like Custom TCP we have chosen port 8000 to deploy our Django application locally, then HTTP port 80 for deploying the application overall the internet and lastly HTTPS port 22.
9. Selected Review and Launch. Review the instance details.
10. Selected create new key pair and made a new key and downloaded the key pair file in our local machine. Then launch the instance.
11. Chose View Instances to check the status of your instance.



After that we have connected the ubuntu instance using SSH key. Then we have cloned our project into that instance and installed all the packages which helped to deploy our web application. Then with the help of **Nginx** and **Gunicorn** we have hosted our web application on port 80 permanently. With the help of that instance, we have created an **AMI (Amazon Machine Image)**. Then we have created **template** with that **AMI id** which we made before.

Then we have used **Elastic load balancing** and **Auto-scaling** service.

- Auto scaling dynamically adds and removes EC2 instances, while Elastic loadbalancing manages incoming requests by optimally routing traffic so that no one instance is overwhelmed
- Auto Scaling helps to automatically increase the number of EC2 instances when the user demand goes up, and decrease the number of EC2 instances when demand goes down
- ELB service helps to distribute the incoming web traffic (called the load) automatically among all the running EC2 instances
- ELB uses loadbalancers to monitor traffic and handle requests that come through the Internet.
- Using ELB & Auto Scaling
 - makes it easy to route traffic across a dynamically changing fleet of EC2 instances
 - load balancer acts as a single point of contact for all incoming traffic to the instances in an Auto Scaling group.

After creating the template Created and attached a new load balancer as we created a new Auto Scaling group

1. Opened the Amazon EC2 console.
2. On the navigation pane, under **AUTO SCALING**, chose **Auto Scaling Groups**.
3. Chose **Create Auto Scaling group**.
4. Attached template which we have made before.
5. For **Load balancing**, chose **Attach to a new load balancer**.
 - a. Under **Attach to a new load balancer**, for **Load balancer type**, chose whether to create an Application Load Balancer or Network Load Balancer.

- b. For **Load balancer name**, entered a name for the load balancer, or keep the default name.
 - c. For **Load balancer scheme**, chose whether to create a public internet-facing load balancer, or keep the default for an internal load balancer. We have selected internet facing.
 - d. For **Availability Zones and subnets**, selected the public subnet for each Availability Zone in which we chose to launch your EC2 instances. (These prepopulate from step 2.).
 - e. For **Listeners and routing**, we have kept default routing port 80, chose **Create a target group**.
 - f. As we chose **Create a target group** in the last step, for **new target group name**, entered a name for the target group.
6. The we have added the minimum capacity, maximum capacity, and desired capacity of instance.
7. Proceed to create the Auto Scaling group. Your instances will be automatically registered to the load balancer after the Auto Scaling group has been created.

In target we can check the health of our instances.

The screenshot displays the AWS Management Console interface for an Auto Scaling group. The breadcrumb navigation at the top shows 'EC2 > Auto Scaling groups > get-job'. Below this, there are tabs for 'Details' (selected), 'Activity', 'Automatic scaling', 'Instance management', 'Monitoring', and 'Instance refresh'. The main content area is divided into two sections: 'Group details' and 'Launch template', each with an 'Edit' button in the top right corner.

Group details

Desired capacity	2	Auto Scaling group name	get-job
Minimum capacity	2	Date created	Mon Dec 13 2021 08:41:55 GMT+0530 (India Standard Time)
Maximum capacity	3	Amazon Resource Name (ARN)	arn:aws:autoscaling:ap-south-1:153682897053:autoScalingGroup:2d0fc5af-adfd-4d33-bf85-0716b6a3d7c7:autoScalingGroupName/get-job

Launch template

Launch template	AMI ID	Instance type
new-job 🔗 lt-07b1694acd48b3a60	ami-0dc3773ab4e311958	t2.micro

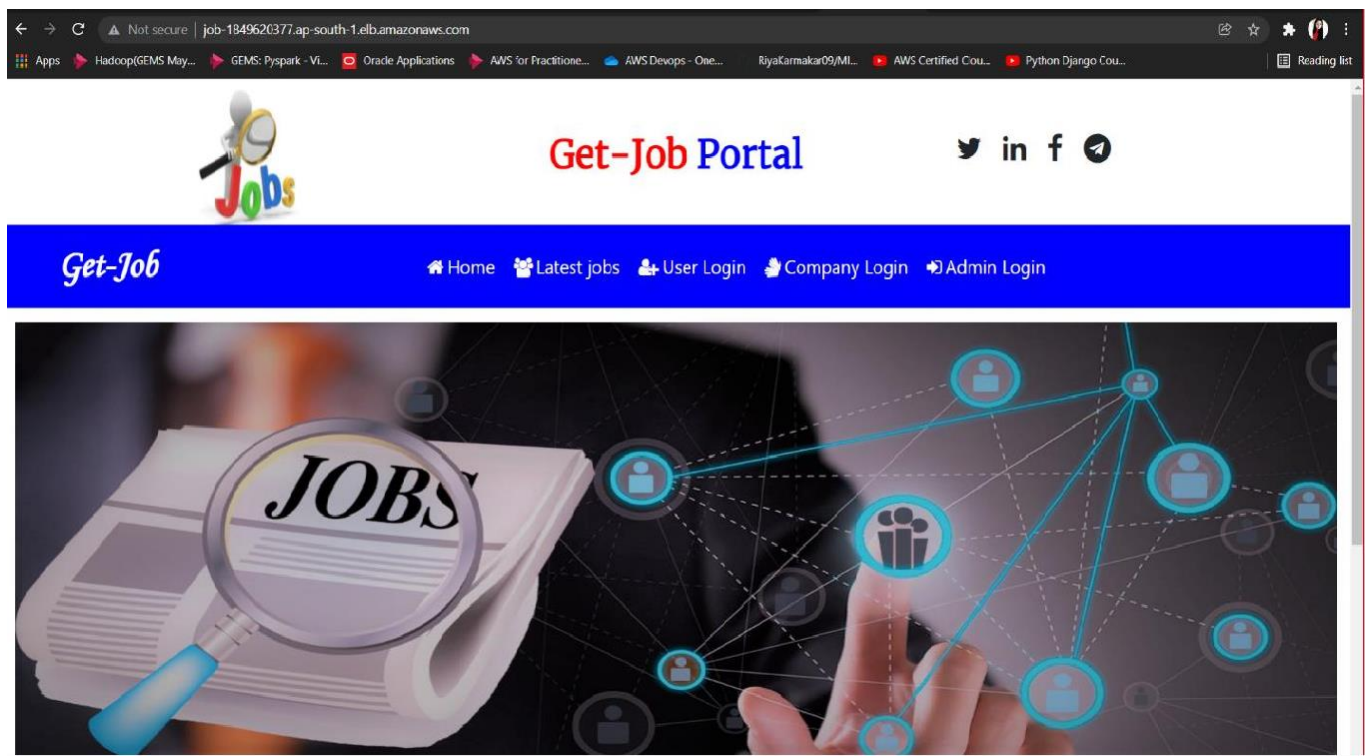
Create Load Balancer Actions

search: job Add filter

Name	DNS name	State	VPC ID	Availability Zones	Type	Created /
job	job-1849620377.ap-south-1....	Active	vpc-0dc3e7b5e5f303ae1	ap-south-1b, ap-south-1a	application	Decembe

Name	job
ARN	arn:aws:elasticloadbalancing:ap-south-1:153682897053:loadbalancer/app/job/d763f4012ebd2b0d
DNS name	job-1849620377.ap-south-1.elb.amazonaws.com (A Record)
State	Active
Type	application

Then we copied the DNS name of the load balancer and checked. It was working perfectly. This is the reference:



Amazon RDS snapshot to S3 bucket:

At first, we have created a **MySQL database on AWS RDS (Relational Database Service)**. We have made this database publicly accessible. We have given a proper database name and password and then created the database. Then we have mention the **RDS endpoint** and other credentials on Django application **settings.py** file.

```
WSGI_APPLICATION = 'jobportal.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE' : 'django.db.backends.mysql',
        'NAME' : 'jobportal',
        'HOST' : 'database.c1fjy3pn38fi.ap-south-1.rds.amazonaws.com',
        'USER' : 'admin',
        'PASSWORD' : 'Riya1998',
        'PORT' : '3306',
    },
}
```

Django administration

Home / Authentication and Authorization / Users

WELCOME, ADMIN / VIEW SITE / CHANGE PASSWORD / LOG OUT

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

JOB

Applys + Add

Companies + Add

Job_seekers + Add

Jobs + Add

Select user to change

Search

Action: [dropdown] Go 0 of 11 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	admin				o
<input type="checkbox"/>	era@gmail.com				o
<input type="checkbox"/>	emohask@gmail.com				o
<input type="checkbox"/>	persistent@gmail.com				o
<input type="checkbox"/>	prem@gmail.com		Prem	Prakash	o
<input type="checkbox"/>	prtya@gmail.com		prtya	paul	o
<input type="checkbox"/>	rahul@gmail.com		Rahul	rev	o
<input type="checkbox"/>	riya@gmail.com		Riya	karmakar	o
<input type="checkbox"/>	rohit@gmail.com		Rohit	Raj	o
<input type="checkbox"/>	tcsa@gmail.com				o
<input type="checkbox"/>	wipro@gmail.com				o

11 users

FILTER

By staff status

All

Yes

No

By superuser status

All

Yes

No

By active

All

Yes

No

Export

Format:

csv

xls

xlsx

tsv

ods

json

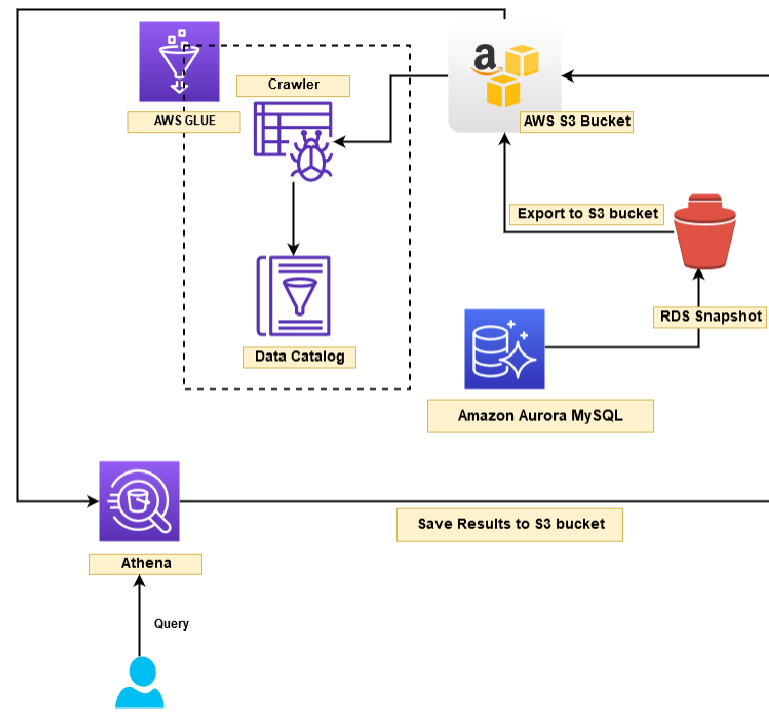
yaml

html

SUBMIT

This is database of our application. From here we can export and import data in any form (like CSV, xls, json, yaml)

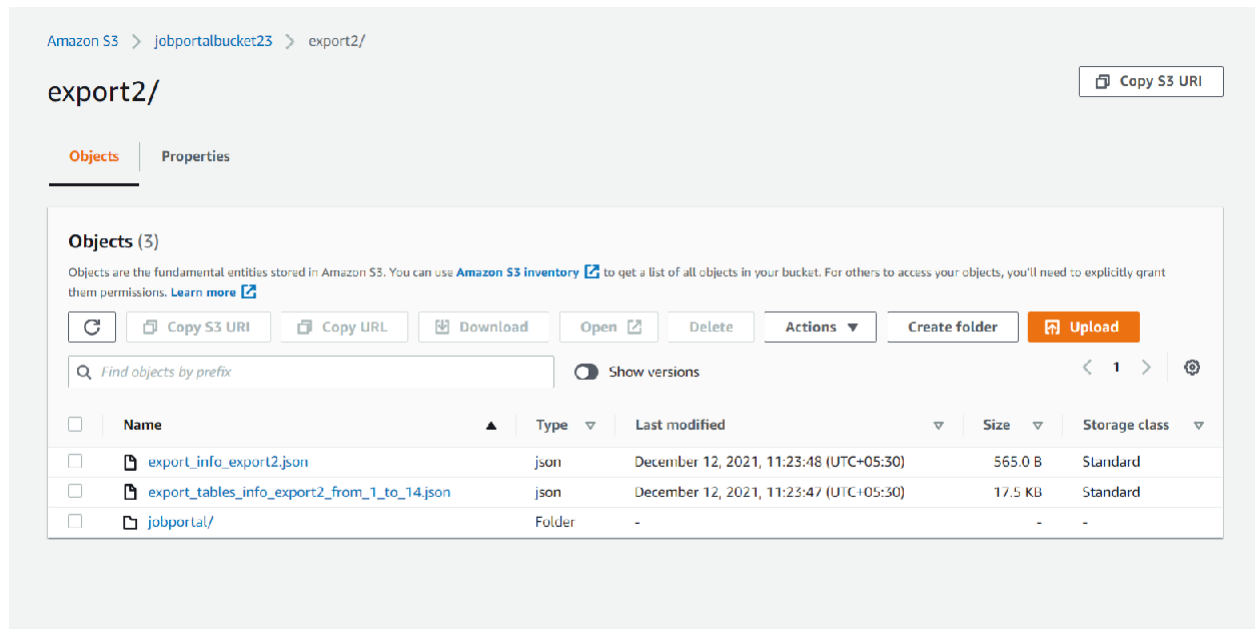
We have exported the database data into **S3(Simple Storage Service) bucket** analyze the data following this way:



These are the steps to export the data into **S3**:

1. At first, we selected the database and in actions we have chosen **take snapshot**.
2. In **snapshots**, select that particular snapshot and in actions chose the **Export to S3**.
3. Give an export identifier name.
4. Then we have created a new bucket on S3 console mentioned the S3 ARN in choose bucket section.
5. Then we chose **create new IAM role** in **select IAM role** section. AWS automatically created an IAM role for exporting the data.

6. Then chose a KMS key which we have made on **KMS (Key Management Service)** and give that key our IAM user access to use.
7. Then we have exported the data on S3 bucket.



After exporting the data into S3 bucket we have analyzed the data using **AWS Athena** and **AWS Glue** services.

These are the following steps for analyzing the data:

1. In **AWS Athena** console, went to the **data sources**.
2. Then we have created a new data source using **S3-AWS Glue data catalog**.
3. In AWS glue we have created a new **Crawler** in which we have mentioned the S3 bucket path name and give the database name.
4. Then we ran the crawler and automatically the tables were imported into **Athena query editor** section.
5. In Query editor section we have chose the database name and then we saw the tables.
6. Then We ran queries in Query section as you can see below.

Amazon Athena > Query editor

Editor Recent queries Saved queries Settings Workgroup primary

Data

Data Source: AwsDataCatalog

Database: jobportal

Tables and views: Create Filter tables and views

Tables (16)

- export_info_export2_json
- export_tables_info_export2_from_1_to_14_json
- jobportal_auth_group

SQL Ln 1, Col 32

```
1 SELECT username from jobportal_auth_user
```

Run again Cancel Save as Clear Create

Completed Time in queue: 0.103 sec Run time: 0.573 sec Data scanned: 0.16 KB

jobportal_auth_group_permissions jobportal_auth_permission jobportal_auth_user jobportal_auth_user_groups jobportal_auth_user_user_permissions jobportal_django_admin_log

Views (0)

Results (11) Copy Download results

Search rows

username

admin
riya@gmail.com
rohit@gmail.com
prem@gmail.com
rahul@gmail.com
persistent@gmail.com
tcs@gmail.com
wipro@gmail.com
cts@gmail.com
emphasis@gmail.com
priya@gmail.com

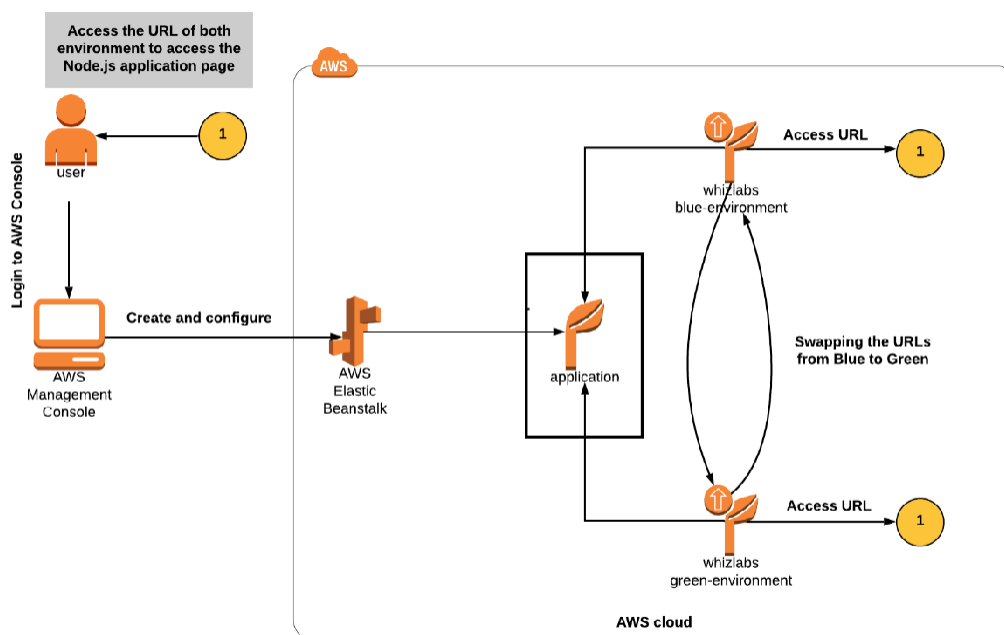
These query results are also saved in another S3 bucket.

Blue/Green Deployments with Elastic Beanstalk:

AWS Elastic Beanstalk is a **Platform as a Service (PaaS)** offered by Amazon for deploying and scaling web applications. It allows us to use a wide selection of application platforms. It also enables us to have a variety of application deployment options.

Blue/Green deployments are used to update an application from one version to another version without any downtime. The **blue environment** is your existing production environment carrying live traffic. The **Green environment** is an identical parallel new environment running a different or updated version of the application. Deploying of **Blue/Green environments** such that you can deploy a new version to a separate environment.

Following these steps, we have done the blue-green deployment:



Step 1: Created an Elastic Beanstalk Application.

Step 2: Created a Blue Environment for python Application.

Step 3: Created a Green Environment for Updated python Application.

Elastic Beanstalk > Environments

All environments Actions Create a new environment

Filter results matching the display values

	Environment name ▲	Health ▼	Application name ▼	Date created ▼	Last modified ▼	URL ▼	Running versions ▼	Platform ▼	Platform state
<input type="radio"/>	Blue-environment	Severe	blue-green-deployment	2021-12-15 09:52:55 UTC+0530	2021-12-15 09:57:29 UTC+0530	Blue-environment.eba-g4rugypa.us-east-1.elasticbeanstalk.com	blue-green-deployment-source	Python 3.8 running on 64bit Amazon Linux 2	Support
<input type="radio"/>	Green-environment	Severe	blue-green-deployment	2021-12-15 09:56:23 UTC+0530	2021-12-15 10:01:24 UTC+0530	Green-environment.eba-g4rugypa.us-east-1.elasticbeanstalk.com	blue-green-deployment-source-1	Python 3.8 running on 64bit Amazon Linux 2	Support

Future scope:

- We will host the web application on a domain name using Route 53.
- We will use cloud formation to deploy the web application and will make code pipeline.
- Import data from S3 bucket to MySQL database using lambda function.

References:

- Get started with Amazon EC2 Linux instances -
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html
- Attaching a load balancer to your Auto Scaling group -
<https://docs.aws.amazon.com/autoscaling/ec2/userguide/attach-load-balancer-asg.html>
- Creating a amazon RDS db instance -
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateDBInstance.html
- Exporting DB snapshot data to S3 bucket -
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ExportSnapshot.html?sc_channel=EL&sc_campaign=Demo_2020_vid&sc_medium=YouTube&sc_content=video6745&sc_detail=Documentation&sc_country=US
- <https://www.youtube.com/watch?v=lyNGeDg6EII&t=60s>
- <https://www.youtube.com/watch?v=9JhgmdXgszs&t=1s>
- <https://www.youtube.com/watch?v=T1N2KwGxGsY>

Conclusion:

Working on this project as team first time with our colleagues in professional journey where we implemented many technologies which we have learnt during training period and also it was a phase of learning new things and implementing it. We would like to extend our deepest regards to our organization for providing us such a wonderful opportunity of learning and polishing our skills enough to use them in development of this project, also our trainers played most crucial role in shaping our knowledge and they deserve loads of appreciation.

Thank you!