Salary Management Web Application

Kumar Sadhu

Abstract

The salary management web application is which shows the activity between the salary, Timesheet, payments, employees. Using our payroll module allows you to keep track of payments and other details. In addition to the history of when a salary change occurred, you can now store the reason behind that change as well as keep track of new job assignments and job descriptions. It simplifies the process of record maintenance of employees' information and their payslips in an organization. On the other end, The application can be software for the small business to track the data of the employees, where admins can do CRUD operation on the employee information. Through the platform, it automatically cre- ates Auto-Scaling, generates logs that you can view to troubleshoot issues. The application is deployed in AWS Elastic Beanstalk so, It reports information about environment and instance health, Monitoring like Response time, CPU utilization, Sum requests, and other Amazon Cloud Watch metrics.

Keywords

Django, AWS Elastic Beanstalk, Timesheet, Load balancer, Monitoring, Views, Templates, Models, URL'S, HTML, Payslips, Salary, Employee.

Introduction

Directly salary computation is done physically, it set aside such an extensive amount of effort to make payout all things considered. It likewise requires some investment to make payslip prepared. Because of manual procedure some time it requires some investment, It defers the pay dissemination. This is a major issue to oversee when pay isn't produced in time. The other primary issue errors, even with deceiving check anywhere a few mistakes will occur, this again makes a huge issue.

The client uses MS Excel and maintains their records however it is not possible for them to share the data from multiple systems in a multi-user environment, there is alot of duplicate work and the chance of mistake. At the point when the records are transformed, they have to refresh every single exceed expectations document. There is no alternative to discover and print recently spared records. There is no security anyone can get to any report and delicate information additionally reports of the rundown. This Payroll Management System is utilized to beat the whole issue which they are confronting as of now and making total automation of the manual framework to a modernized framework.

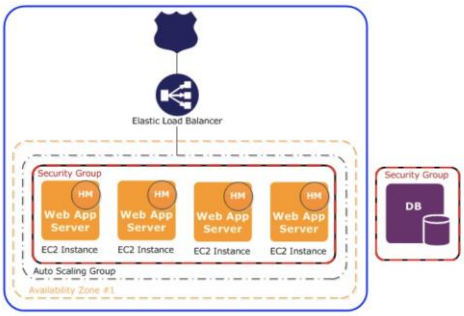


Fig. 1. The following diagram shows an example Elastic Beanstalk architecture for a web server environment tier, and shows how the components in that type of environment tier work together.

The proposed programming will take care of the considerable number of issues they are confronting now. This product is structured in such a way that it will create the pay consequently consistently in time. So there very little concerns.

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

In this project, The application is deployed into Elastic Beanstalk where it reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring. Amazon CloudWatch is a monitoring and management service that provides data and actionable insights for AWS, hybrid, and on-premises applications and infrastructure resources.

Salary Management is a distributed application, created to assess the exhibition of representatives working in any association. It keeps up the data about an organization, individual details, Salary details of their employees. The application is really a set-up of uses created utilizing the DJANGO Framework.

It is easy to comprehend and can be utilized by any individual who is not even familiar with the basic employee system. It is easy to understand and just requests that the client follow bit by bit tasks by giving him scarcely any alternatives. It is quick and can perform numerous tasks of an organization.

This software package has been developed using the powerful coding tools of HTML at Front End and python as Back End and uses the DJANGO framework. Due to the Visual highlights, the product is very easy to use. The project contains various modules like the Contact Us page and other useful links. This adaptation of the product has a multi-client approach. For further enhancement or ad- vancement of the module, the user's input will be considered.

This task essentially manages six modules and their further sub-modules. The first module is the worker module into which we can enter representative details, for example, his name, address, telephone number, his essential compensation, and some more. After that, we can see the details further by utilizing the representative username, and we can alter the details moreover. Next comes the paid module in this we can see the compensation given to the representative. The third module is where representatives can fill their time-sheet to get track of working hours and leave management. Next comes, the track of Log Information where a representative can view all the working days, paid working days, etc. The representative can download CSV. In the fifth module, the Representative can view payslip and download in the form of pdf. Next comes the admin module where admin has all rights to change representatives authorization, to do CRUD operations on representatives.

Objectives

To store up-to-date information of the employees. • To minimize the manual checking of each daily time record of the employee.

• Run accurate payroll on time.

• Automatically handles the details of capacity provisioning, application health monitoring, testing, load balancing.

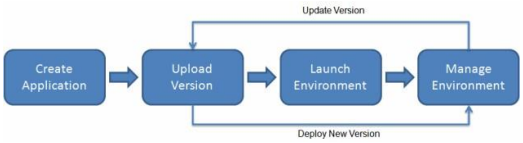
• Employees will be able to create online requests for any kind of absences, which will automatically be added to an integrated approval workflow system.

Significance and Implications

• The three most common issues faced by payroll specialists, manage timesheets, keep track of personal files, correct taxation of benefits.

• It is well known that for companies with many employees, the correct management of daily presence can become a difficult task for the HR department, especially when dealing with employees working in different locations, all around the city, district or country, working based on a non-standard quota.

• AWS Elastic Beanstalk provides so many benefits like fast and simple to deploy the application, integrate with the environment, developer productivity, complete resource



control.

• It also scales the application up and down on applications specific need using easily adjustable Auto Scaling settings.

Environment Setup

The python package manager is pip, used to install and manage Python software packages. Pip also can upgrade, show, uninstall project dependencies, etc. Installing a package through pip, just run pip install <package>. With the help of pip freeze, it will list all the dependencies listed in case-insensitive sorted order.

For project setup, need to perform some set of actions to make applications available, creation of a virtual environ- ment, installing packages using pip install <package> com- mands. Then save all the packages in the file with Pip freeze > requirements.txt. Installing awsebcli, virtualenv, pip, python 3.6. The AWS Elastic Beanstalk command-line in- terface that can use to create, configure, and manage elas- tic beanstalk environments. you can set up the environ- ment through https://github.com/aws/aws-elastic-beanstalk- cli-setup.

Final stage of Django deployment, AWS Elastic Beanstalk configuration files (.ebextensions) to your web application's source code to configure your environment and customize the AWS resources that it contains. The option settings sec- tion of a configuration file defines values for configuration options. Configuration options let you configure your Elastic Beanstalk environment, the AWS resources in it, and the software that runs your application. Configuration files are only one of several ways to set configuration options.

The environment is the heart of the application. In the Fig1, the environment is shown within the top-level solid line. When you create an environment, Elastic Beanstalk provisions the resources required to run your application. AWS resources created for an environment include one elastic load balancer (ELB in the diagram), an Auto Scaling group, and one or more Amazon Elastic Compute Cloud (Amazon EC2) instances. Every environment has a CNAME (URL) that points to a load balancer. The environment has a URL, such as myapp.us-west-2.elasticbeanstalk.com. This URL is aliased in Amazon Route 53 to an Elastic Load Balancing URL—something like abcdef-123456.us- west-2.elb.amazonaws.com—by using a CNAME record. Amazon Route 53 is a highly available and scalable Domain Name System (DNS)web service. It provides secure and reliable routing to your infrastructure. The domain name that you registered with your DNS provider will forward requests

to the CNAME.

The load balancer sits in front of the Amazon EC2 instances, which are part of an Auto Scaling group. Amazon EC2 Auto Scaling automatically starts additional Amazon EC2 instances to accommodate increasing load on your application. If the load on your application decreases, Amazon EC2 Auto Scaling stops instances, but always leaves at least one instance running.

Resources Created by Elastic Beanstalk

The below resources are created and also managed by AWS Elastic Beanstalk.

• EC2 instance – An Amazon Elastic Compute Cloud (Amazon EC2) virtual machine configured to run web apps on the platform that you choose. Each platform runs a specific set of software, configuration files, and scripts to support a specific language version, framework, web container, or combination of these. Most platforms use either Apache or Nginx as a reverse proxy that sits in front of your web app, forwards requests to it, serves static assets, and generates access and error logs.

• Instance security group – An Amazon EC2 security group configured to allow inbound traffic on port 80. This resource lets HTTP traffic from the load balancer reach the EC2 instance running your web app. By default, traffic isn't allowed on other ports.

• Amazon S3 bucket – A storage location for your source code, logs, and other artifacts that are created when you use Elastic Beanstalk.

• Load balancer – An Elastic Load Balancing load balancer configured to distribute requests to the instances running your application. A load balancer also eliminates the need to expose your instances directly to the internet.

• Load balancer security group – An Amazon EC2 secu- rity group configured to allow inbound traffic on port 80. This resource lets HTTP traffic from the internet reach the load balancer. By default, traffic isn't allowed on other ports.

• Auto Scaling group – An Auto Scaling group configured to replace an instance if it is terminated or becomes unavailable.

• AWS CloudFormation stack – Elastic Beanstalk uses AWS CloudFormation to launch the resources in your environment and propagate configuration changes. The resources are defined in a template that you can view in the AWS CloudFormation console.

• Amazon CloudWatch alarms – Two CloudWatch alarms that monitor the load on the instances in your environment and that are triggered if the load is too high or too low. When an alarm is triggered, your Auto Scaling group scales up or down in response.

• Domain name – A domain name that routes to your web app in the form subdomain.region.elasticbeanstalk.com.

Methodology

1.Django Application

The Django application is Ridiculously fast, Fully loaded, Reassuringly secure, Exceedingly scalable, Incredibly versatile. It is free and also open-source software where web development concepts in a matter of hours.

The terminology used in Django is model, templates, URL config, views. wherein Django model layer provides an abstraction layer for manipulating and creating the data of a web application. The models include indexes, fields, meta options, model classes. The model layer has Query sets where it is making queries, lookup expressions. It contains an advanced version where we can add custom fields, custom lookups, conditional expressions, SQL library.

In my project, There are five model classes where Employee Model, Salary Model, Timesheet, Department, Designation. where Django automatically provides a User Model for user registration, login. The Employee model consists of basic employee details which include Contact, CTC, and also references columns as foreign key from Department, Designation table.

The Salary Model references Employee model, Consist month, year, working days, other allowances. The salary month contains all the payslip details and it is indirectly linked to the Time-sheet log. The Timesheet references employee model, date, status where employees need to fill the timesheet every day. It will save all the monthly statements of the employee. The Department and Designation models have all the records of departments, designation in the organization. For Example, Department - Accounts, Testing, etc. Designation - Manager, engineer, developer. where through these tables we can to authorization in the application.

The user authentication in user objects are the core of the authentication system. They typically represent the people interacting with your site and are used to enable things like restricting access, registering user profiles, associating content with creators etc. Django framework provides authenticating users, permissions, authorization, default permissions.
The calculation of the monthly salary is the main task. So, The Salary calculation method lists all the objectives, it inherits the business knowledge. where it has all calculations fields like PF calculation, HRA, Tax calculations, working days, leaves, Basic pay, Deductions.

The 2nd module of the Django web application is the View layer, the main concept encapsulates the logic responsible

for users' requests and responses. There are references like Template view, Create view, request, and response objects. In class-based views where the user can build views and also edit views, API reference. By the views request and responses, We can Generate PDF, CSV. In this project, I handled operations like generating PDF'S and also CSV'S.

There is a login-view where it is used authenticate the user and redirect to the home page template, it gets a request from the user login template. The log-out view will redirect the user login page template, which disables the auth(Django provides authentication default in the framework).

The Home view, Contact Us which gets reference from the request object, renders to the home page, and contact Us template.

The main module of the project is to generate and view payslip for a month, for the following requirements, the Salary-Select view, salary-details view, and gen-pdf view are executed. The salary select-view gets username and request as the references where it filters employee, salary month, salary year to view on the salary-select template. so that, user can select the appropriate month to view the payslip or download the payslip. when the employee submits the following form. when the employee was performed using the HTTP "POST", the salary month and salary year fields are referenced in the view to calculate the salary details(salary calculation method). Then its renders to salary details template and displays all the values on the page.

To generate pdf, Django can import report lab toolkit xhtml2pdf import Pisa. so this HTML template gets converted into pdf. The HTML will encode into the Pisa document and HTTPResponse for the view is pdf. If the current request from a user was performed using the HTTP "POST" then it will filter all the data from models render the following data to report lab toolkit Pisa document.

The other important module of the web application is the Timesheet module, for this module, there are Timesheet-view, Timesheetselect-view, Timesheet log view, export CSV. When the employee fills timesheet form and submits the form, the date from the date picker, status is saved into the timesheet model, and also the working days, leave in the salary model gets updated. So, the pay slip and log info can be retrieved by the employee. Then it renders to timesheet template.

The Timesheet-view gets username and request as the references where it filters employee, salary month, salary year to view on the timesheet-select template. when the form is submitted by the employee, it renders to Monthly statements log information template or downloads the CSV. where it lists all the information of working status on the particular dates. To generate CSV, need to import csv, writing all the fields names, all the content from the queryset, then each row is written into CSV.

Django lets the user to design URLs with no framework limitations. Django determines the root URLconf module to use. Ordinarily, this is the value of the ROOT-URLCONF setting, but if the incoming HttpRequest object has a urlconf attribute (set by middleware), its value will be used in place of the ROOT-URLCONF setting. Django runs through each URL pattern, in order, and stops at the first one that matches the requested URL, matching against path-info. The path converters in the Django are str, int, uuid, path, slug.

The 3rd Module of the Django web application is a template layer that provides a designer-friendly syntax for rendering the information to be presented to the user. k, Django needs a convenient way to generate HTML dynamically. The most common approach relies on templates. A template contains the static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted. where it has built-in tags and filters, template APIs, custom tags, custom filters, HTML, CSS, Scripting, javascript, Ajax controls, accessing static files like images, bootstrap, and other controls.

There two different types of templates the user can access, admin templates, and application templates. The built in tags and filters used in the application are block, csrf-token, cycle(if, for, tr, td), debug, extends, filter, for..empty. date- time. The admin templates are accessed by the administra- tor by extends tags it will import all the change-list.html, base.html(admin HTML pages). Django provides auto- matic templates for the admin page and we can create supe- ruser. There are Employee-changelist, CSV-form, employee-success templates. The CSV form template is to import the employee CSV to save the records to the employee model.

The application templates are accessed through views response, by extends, block content tag, it can inherit all base templates to other templates. In the application, there are home, login, nav, pdf, salary details, salary-select,employee-details, timesheet, timesheet-log, timesheet-select. The nav template describes all the links to other templates(HTTP get method). These templates are the user interface of the application.

The 4th module of the application is Forms where Django provides a framework to facilitate the creation of forms and the manipulation of the form data. By default, the fields in a ModelForm will not localize their data. To enable localization for fields, you can use the localized-fields attribute on the Meta class. Form API, Built-in widgets, built-in fields, forms for the model, integrating media, Formsets, customizing validations. In the application, there are Salary month year form, time-sheet form, CSV import form which is used in the admin template.

Another main module of the application is the development process where it provides various components and tools to help in the development and testing of the application.

settings.py provides all the overview, full list of settings how to handle all the installed applications, allowing the hosts, time zone, database controls, templates accessing, sessions, sites, static files, Auth and other settings. If ALLOWED-HOSTS is empty and DEBUG=True, subdo- mains of localhost were allowed. apps.py provide to register all the application(included Management application) cre- ated in the project, Django-admin, manage.py. For testing, we can write and run test cases in the framework, Django provides included testing tools. For deployment, it maintains all the WSGI servers, deploying static files, tracking code errors, deployment checklist. Migrations are Django's way of propagating changes you make to your models (adding a field, deleting a model, etc.) into your database schema.

The admin module is an automated admin interface provided by Django. It reads metadata from your models to provide a quick, model-centric interface where trusted users can manage content on your site. It has an admin site, admin actions, admin documentation generator. whenever admin wants to make some changes to model fields, the superuser needs to register models in admin.py, and also superuser can customize the templates of the admin site by admin actions. In the admin site, the superuser can import employee CSV, download Timesheet monthly sheets, Do CRUD operations on all the models.

Security is a topic of paramount importance in the devel- opment of Web applications and Django provides multiple protection tools and mechanisms are security overview, clickjacking protection, cross-site request forgery protection, cryptographic signing, security middle-ware. It provides internationalization, localization by web UI formatting and form input.

2 Deployment in AWS Elastic Beanstalk

1. Set up a python virtual environment
2. Pip freeze > requirements.txt
3..eb extensions directory in django application. where configuration file django.config, with WSGI path.
4. Installing awsebcli, creating the environment, set up SSH instances, by the command eb deploy - EB CLI bundles up the contents of your project directory and deploys it to your environment.
5. By adding CNAME in the allowed hosts in settings.py, then deploy the application again in the environment.

3 AWS Elastic Beanstalk

The core idea of the project is running a production website, it is important to know that your application is available and responding to requests. To assist with monitoring your application's responsiveness, Elastic Beanstalk provides fea- tures that monitor statistics about your application and create alerts that trigger when thresholds are exceeded. Elastic

```
branch-defaults:
  default:
    environment: django-salarymanagement
    group_suffix: null
global:
  application_name: SalaryManagement
  branch: null
  default_ec2_keyname: aws-eb
  default_platform: Python 3.6
  default_region: us-west-2
  include_git_submodules: true
  instance_profile: null
  platform_name: null
  platform_version: null
  profile: eb-cli
  repository: null
  sc: null
  workspace_type: Application
```

Fig. 2. .elasticbeanstalk/config.yml

Beanstalk reports the health of a web server environment depending on how the application running in it responds to the health check. The colours are Grey, Green, Yellow and Red.

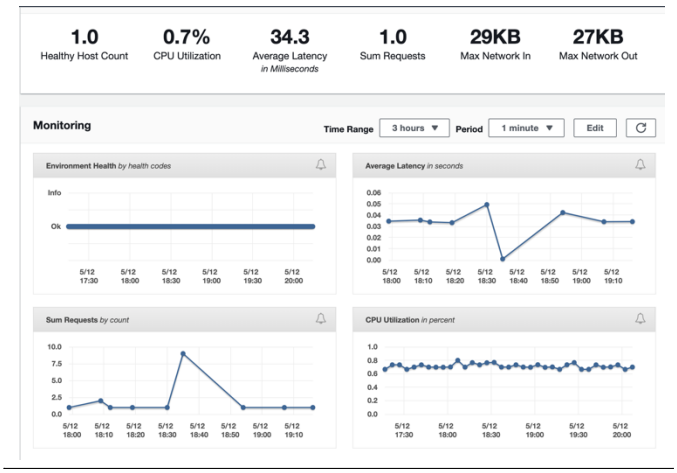


Fig. 3. Monitoring page.

The load balancer sends request to all the instances in the application environment to confirm instances are health. It determines instances health by EC2 instance status in single instance. The health settings, including HTTP health check URLs, cannot be used in these environment types.

The graph in below Figure of the monitoring page in the environment were provided by cloudwatch metrics.
CPU Utilization - percent of compute units in use.
NetworkIn, NetworkOut - No.of bytes sent and received.
RequestCount - Number of completed requests.
Number of seconds between when the load balancer relays a request to an instance and when the response is received.
Elastic Beanstalk installs a CloudWatch log agent with the default configuration settings on each instance it creates.

When you enable instance log streaming to CloudWatch Logs, Elastic Beanstalk sends log files from yourenviron- ment's instances to CloudWatch Logs. Different platforms stream different logs.

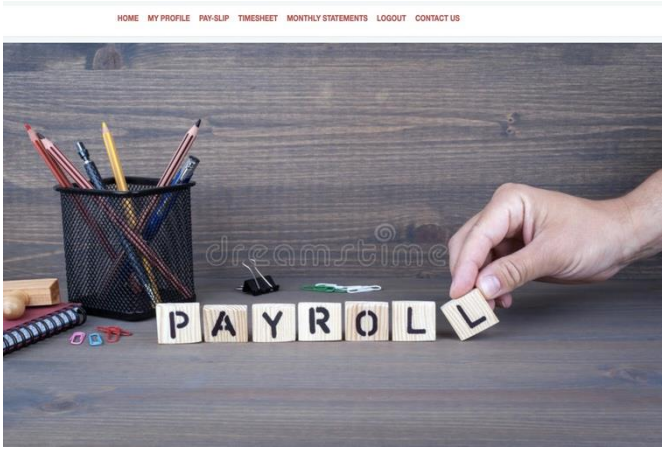


Fig. 5. Home page.

Results

The application provides Payroll Processing, Tax Filing, Year-end Reporting, Automatic Calculation Basic Pay, HRA. The manual system requires filing the necessary payroll data by hand, which consumes time and increases clutter. This application generates and stores payroll and employment data. Below are the few results of the application. The Beanstalk provides auto-scaling where if CPU utilization crosses 80 percent the environment will create another instance where load balancer comes into action, Elastic Load Balancing to scale and balance workloads. It provides tools to monitor health of the application.

Pay Slip of the Month : 4 and Year : 2020

First Name	sad
Last Name :	hu
Contact number :	9898989898
Email :	krishansd@gmail.com
Department :	safv
Designation :	Lecturer
SALARY STATEMENT :	
CTC per year :	$ 350000
CTC per month :	$ 29166
EARNINGS :	
Basic :	$ 29166
Dearness Allowance(DA) :	$ 5833.2
House Rent Allowance(HRA) :	$ 14583.0
Conveyance Allowance(CA) :	$ 80
Other Allowance :	$ 800.0
Total Earning :	$50462.2
DEDUCTIONS :	
Provident Fund(PF) :	$ 4199.9
Professional Tax(PT) :	$ 252.31
Tax Deducted at Source(TDS) :	$ 18333.33
Total Deductions :	$ 22785.54
Total Take-away Salary :	$ 27676.66
LEAVE DETAILS :	
Number of working days :	30
Number of days leave :	25
Amount to be deducted :	$ 23050.0
Net Take-away Salary :	$ 4626.66
Total : $ 4626.66	

Fig. 4. pay slip of employee.

Conclusion

The proposed system was can help the small organizations to manage employee pay related data as personal information, salary information, Timesheet log Info. It is integrated with AWS Elastic Beanstalk and provides a one-stop experience for you to manage the lifecycle of the application. Its has load balancer, Auto-scaling, Cloud watch, health montoring. It supports the infrastructure to access globally, apart from this creating new instances when CPU utilization, Sum request gets peak.

It also gave me the experience and exposure. The application used different products and software to get a task completed were I was able to get hands on experience with that software and technology. I was able to learn the key aspects which determines an employee in the sector and what is looked out while hiring a professional into the industry. Overall, I can conclude that doing project was an amazing experience.

References

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/awseb-dg.pdf
https://docs.djangoproject.com/en/3.0/
https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/eb3-cmd-commands.html
https://www.codingforentrepreneurs.com/blog/html-template-to-pdf-in-django/
https://books.agiliq.com/projects/django-admin-cookbook/en/latest/import.html
https://www.youtube.com/channel/UC33uwXXDrI5TxG4IXnjS28g

http://www.jonthornton.com/jquery-timepicker/
https://docs.djangoproject.com/en/3.0/topics/db/queries/
https://www.w3schools.com/howto/howto-js-slideshow.asp
https://hakibenita.com/scaling-django-admin-date-hierarchy
https://www.w3schools.com/colors/colors-picker.asp
https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/health-enhanced.html
https://docs.djangoproject.com/en/3.0/ref/forms/