Employee Celebrations Tracker

The Employee Celebrations Tracker is a Django-based web application that helps manage employee data and track upcoming birthdays and work anniversaries.

Here are the main functionalities of the project:

1. **User Authentication:** Users can create an account, log in, and log out of the application. Authentication ensures that only authorized users can access the employee data and upcoming events.
2. **Employee Data Management:** The application allows users to upload employee data from an Excel sheet. The data includes employee names, birthdates, hire dates, email addresses and likes-dislikes. The uploaded data is stored in a database and can be used for further analysis and tracking.
3. **Upcoming Birthdays:** The application calculates and displays the upcoming birthdays of employees. Users can view the upcoming birthdays within a specified range, such as the next 7 days, 14 days, or 1 month. This helps managers and team members stay aware of upcoming birthdays and plan celebrations.
4. **Work Anniversaries:** The application also tracks and displays the upcoming work anniversaries of employees. Users can view the upcoming work anniversaries within a month. This feature allows organizations to recognize and celebrate employees' years of service.
5. **Frontend Interface:** The application has a frontend interface that allows users to interact with the employee data and view the upcoming events. The interface is designed using HTML, CSS, and

JavaScript, providing a user-friendly and visually appealing experience.

6. **Error Handling and Validation:** The application includes error handling and validation mechanisms to ensure data integrity and user input validation. It validates the uploaded Excel file format and handles errors gracefully to provide a smooth user experience.

Overall, the Employee Celebrations Tracker simplifies the management of employee data and helps organizations celebrate birthdays and work anniversaries in a timely and organized manner.

There are two personas in our application:

Admin Persona:

- The admin persona typically has higher privileges and administrative access within the system.
- Responsibilities:
    - Manage employee data: The admin can view, add, edit, and delete employee data.
    - Upload data: The admin can upload data from an Excel sheet to update the employee database.
    - View upcoming events: The admin can see upcoming birthdays and work anniversaries.
    - Manage user accounts: The admin can create, edit, and delete user accounts, including other admin and employee accounts.
    - Manage permissions: The admin can assign and revoke permissions for different users and roles.

- Permissions:
    - can_upload_data: Allows the admin to upload data from an Excel sheet to update the employee database.
    - can_view_upcoming_events: Allows the admin to view upcoming birthdays and work anniversaries.

Employee Persona:

- The employee persona typically has limited access and is primarily focused on viewing and interacting with their own data.
- Responsibilities:
    - View upcoming events: The employee can see upcoming birthdays and work anniversaries.
- Permissions:
    - can_view_upcoming_events: Allows the employee to view upcoming birthdays and work anniversaries.

Technology Used:

Django framework: Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It’s free and open source. It follows the model–template–views (MTV) architectural pattern.

In Django's MVT architecture, the model still represents the data and business logic, the view handles the logic for processing requests and returning responses, and the template is responsible for the presentation of data.

Pandas: Pandas is a powerful data manipulation and analysis library for Python. It provides data structures and functions to efficiently manipulate and analyze structured data, including support for handling large datasets, cleaning and transforming data, and performing statistical calculations.

In our application, we have utilized pandas for reading and processing data from Excel files.

Amazon RDS (MySQL): Amazon RDS is a managed database service provided by Amazon Web Services (AWS). Amazon RDS takes care of database administration tasks such as backups, software patching, and scaling, allowing developers to focus on application development.

In this project, it is used to host the MySQL database for storing employee data.

Amazon EC2 (Elastic Compute Cloud): Amazon EC2 provides scalable and resizable compute capacity in the cloud. It allows you to create and manage virtual machines (instances) on the AWS cloud infrastructure.

In this project, EC2 instances are used to host Amazon RDS's database instance, jenkins and docker.

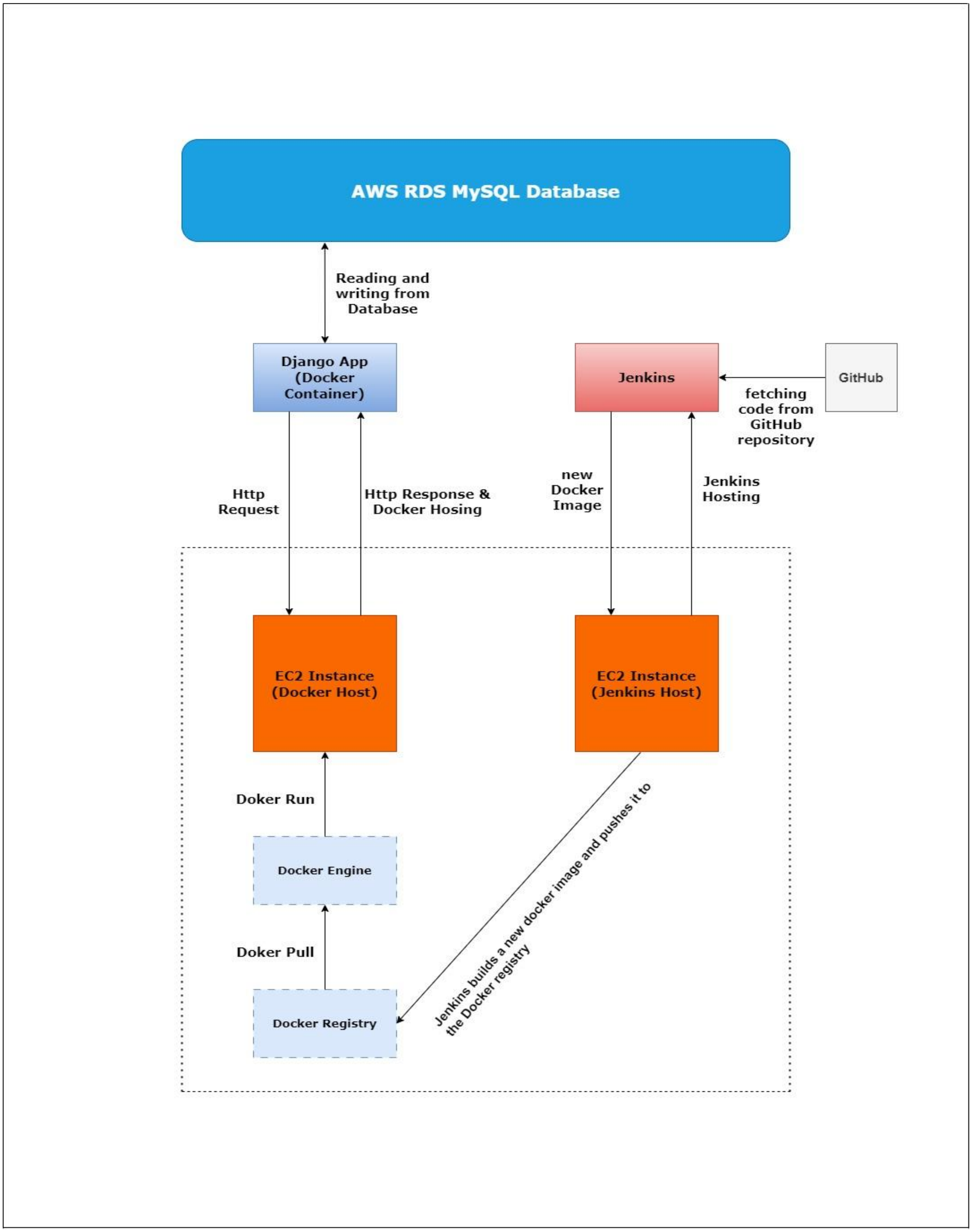
Docker: Docker is an open-source platform that allows you to automate the deployment, scaling, and management of applications using containerization. Containers provide a lightweight and consistent environment for running applications, making it easier to package and deploy them across different environments.

In this project, Docker is used to containerize the Django application, along with any required dependencies, ensuring consistency and portability.

Jenkins: Jenkins is an open-source automation server that enables continuous integration and continuous delivery (CI/CD) pipelines. It helps automate the building, testing, and deployment of applications.

In this project, Jenkins is used to set up a CI/CD pipeline for automating the deployment of the Django application. It is configured to automatically build the Docker image and deploy the application to the EC2 instances.

Architecture Diagram:


AWS RDS MySQL Database
Reading and writing from Database
Django App (Docker Container)
Jenkins
GitHub
fetching code from GitHub repository
Http Request
Http Response & Docker Hosing
new Docker Image
Jenkins Hosting
EC2 Instance (Docker Host)
EC2 Instance (Jenkins Host)
Doker Run
Docker Engine
Doker Pull
Docker Registry
Jenkins builds a new docker image and pushes it to the Docker registry