Technical Documentation for Employee Dashboard with Smart Chatbot & Email Alerts

1. Overview

This project is a comprehensive employee analytics dashboard built using **Python**, **Dash**, and **Plotly**, featuring:

- **Interactive Visualizations** for HR insights.
- **Smart Chatbot** with predefined Q&A and natural language processing.
- **Employee ID search tool**.
- **Automated Email Alerts** for high attrition risk employees.

The dashboard processes employee data, performs analytics, and visualizes KPIs and trends to support decision-making in workforce management.

2. Python Libraries Used

Library	Purpose
dash	Core framework for building the web app and reactive components.
dash_bootstrap_components	Enables use of Bootstrap components like alerts, buttons, rows, and columns.
plotly.express	For creating visualizations such as pie, bar, scatter, line, heatmap, etc.
pandas	Data handling, filtering, aggregation, and date/time processing.
plotly.io	Sets default themes like plotly_white.
datetime	Used for time-based filtering and formatting in callbacks.
smtplib	Sends email alerts using Gmail SMTP.
email.message.EmailMessage	Constructs and formats HTML/plaintext email messages.
os	Handles file path resolution (especially for reading datasets).
re	Regular expressions used for smart query parsing.

3. Data Handling and Preprocessing

- **Data Source**: CSV file `employee_dataset.csv`
- **Cleaning Steps**:
    - Dropping empty or unnamed columns.
    - Parsing Hire_Date with pd.to_datetime().
    - Removing rows with missing hire dates.

4. Dashboard Layout & Features

- **Header Section**: Project title, chatbot input, dropdown, and chat response.
- **Search Section**: Employee ID-based search tool.
- **KPIs (6 cards)**:
    - Total Employees
    - Avg. Productivity
    - Avg. Satisfaction
    - Avg. Age
    - Avg. Annual Salary
    - Avg. Tenure (Years at Company)
- **Filters**:
    - Department
    - Job Title
    - Hire Date (Start & End: Year, Month, Day)
- **Graphs**:
    - Donut Chart: Remote work frequency by department
    - Bar Chart: Projects handled per job title
    - Line Chart: Performance over years at company
    - Scatter Plot: Age vs performance
    - Box Plot: Satisfaction by department
    - Heatmap: Performance vs satisfaction

5. Smart Chatbot Engine

a. Predefined Q&A

Handled using a dictionary. Triggered via dropdown or natural language.

b. Dynamic Query Parsing (Smart NLP)

- **Synonyms Mapping**: Maps natural language terms to dataset fields (e.g., “tenure” -> `Years_At_Company`)

- **Aggregation Mapping**: Maps query types to pandas methods (e.g., “average” -> `mean`)

c. Chatbot Responses:

- Metric Aggregation (e.g., average salary)
- Group-wise Aggregation (e.g., avg. performance by department)
- Top N Queries (e.g., top 5 performers)
- Unique Value Extraction (e.g., list all departments)
- Employee Detail Lookup (via ID input)
- Fallback response when unrecognized.

d. Output Rendering:

- Uses html.Div, html.P, and dash_table.DataTable to show results.

6. Email Alert Functionality

Purpose:

To alert the HR team when employees with high attrition risk are detected (Retention Risk Index > 1.5).

Components:

- Email composed using EmailMessage.
- Risky employee table generated with pandas.
- Sent via smtplib.SMTP_SSL() using Gmail credentials.

Triggers:

- Manual button click (send-email-btn)
- Status displayed in email-alert

7. Backend Logic: Callbacks

- update_dashboard: Filters dataset based on selections and returns all metrics and graphs.
- update_status: Refreshes dashboard update time and sends email if triggered.
- handle_chat: Handles chat logic (Q&A, NLP processing, ID search).

8. Technical Configuration

- Port: 9116
- Auto-refresh: dcc.Interval every 60 seconds
- Responsive layout using dbc.Container and dbc.Row
- Theme: dbc.themes.LITERA
- Plotly Theme: plotly_white

9. Future Improvements (Can be done)

- Add user authentication for email access.
- Improve NLP with spaCy or transformers.
- Add export-to-Excel functionality for table results.
- Enable filtering by more parameters (like location, gender, etc.).

10. File Paths

- Dataset location: `C:\anaconda\pythonfiles\employee_dataset.csv`

11. Notes

- SMTP uses app-specific password (high security requirement).
- Chatbot supports both dropdown selection and free-form natural language.
- Date filtering uses separate dropdowns for year, month, and day (not a date picker).

12. Author & Contact Details

Developer: Gantana Abhishek
Contact Email: abhishekgantana@gmail.com
Tools Integrated: Dash, Plotly, Power BI, Pandas, SMTP