Approach for Resume Generator

Tools and Technologies

1. **Libraries**:
    - **FPDF**:
        - Facilitates custom layout creation and content addition.
    - **argparse**:
        - Handles command-line arguments for customizable input parameters (e.g., font size, font color, and background color).

Methodologies

1. **Object-Oriented Programming (OOP)**:
    - The project will be structured using classes and methods to promote reusability, readability, and scalability.
    - Example: The `ResumePDF` class encapsulates all functionalities related to PDF creation.
2. **Customizable Design**:
    - Customizable font size, font color, and background color via command-line arguments.
3. **Command-Line Interface**:
    - `parse_args` function to parse user-provided parameters.

Predefined Sections

- Name

- Contact Information

- Education

- Technical Skills

- Experience

- Projects

Code Structure

1. ‘ResumePDF’ Class:

- Handles PDF generation and layout.

- Includes methods for adding sections, headings, bullet points, and customization options.

2. `parse_args()` Function:

- Parses command-line arguments for customization.

3. `generate_resume()` Function:

- Creates the PDF and saves it as `custom_resume.pdf`.

Key Methods

- `header()`: Adds the resume's header, including the name and background color.
- `body()`: Adds the main content, including various sections.
- `add_bullet_points(points)`: Adds bullet points under a section.
- `underline_heading(title)`: Adds a heading with an underline.
- `hex_to_rgb(hex_color)`: Converts a hex color code to an RGB tuple.