**Finance Minister**

Purpose:

The Finance Minister application is designed to help individuals calculate their monthly expenses, savings, and major payments such as car and house payments based on their hourly salary. It aims to provide a clear breakdown of how one's salary could be allocated towards different financial aspects.

Reasons for Creating the Application:

- Empower individuals to manage their finances effectively.

- Provide a tool for calculating and planning monthly expenses.

- Offer insights into financial distributions for savings, car payments, and house payments.

Goals for the Application:

- Enable users to input their hourly salary and calculate monthly budget allocations.

- Display a detailed breakdown of recommended expenses, savings, and payments.

- Provide insights into estimated costs for buying a car, purchasing a house, and long-term savings.

Target Audience:

The target audience for the Finance Minister application includes:

- Working individuals aged 18-60.

- Both genders.

- Individuals seeking financial planning and management tools.

- People from various socioeconomic backgrounds interested in budgeting and financial organization.

Outline of the Final Python Tkinter GUI Application:

1. \*\*Introduction:\*\*

- Display the application name and purpose.

- Show the logo or a welcome message.

2. \*\*Input Section:\*\*

- Input field for users to enter their hourly salary.

- Calculate button to trigger the financial calculations.

3. \*\*Result Display:\*\*

- Display section for the calculated breakdown of expenses, savings, and payments.

4. \*\*Details Section:\*\*

- Buttons to view detailed information about car payments, house payments, and savings.

- Show details about each category in a separate frame.

5. \*\*Functionality:\*\*

- Allow users to reset the inputs and start afresh.

- Display images and details based on user selections for car, house, and savings.

6. \*\*User Experience:\*\*

- Ensure a user-friendly interface with clear labels and instructions.

- Enhance readability and usability by appropriately sizing text and images.

- Offer an intuitive flow for users to input data and view results.

7. \*\*Final Touches:\*\*

- Ensure error handling for incorrect inputs.

- Make sure the application runs smoothly without unexpected crashes.