LaTeX code for the Objective of the project and Output:

```
\documentclass[16.5pt]{extarticle}
\usepackage{fancyhdr}
\usepackage{datetime2}
\usepackage[left=2.5cm, right=2.5cm, top=2.5cm, bottom=2.5cm]{geometry}
\usepackage{titlesec}
\pagestyle{fancy}
\fancyhead{} % clear all header fields
\fancyhead[RO,LE]{\textbf{Palak Choudhary} \\ \textbf{0801CS221104}}
\fancyfoot{} % clear all footer fields
\fancyfoot[LE,RO]{\thepage}

\begin{document}

\titleformat{\section}[block]{\bfseries\filcenter\LARGE}{\thesection}{1em}{}

\section*{LaTeX}

\titleformat{\section}[block]{\bfseries\filcenter}{OBJECTIVE OF THE
PROJECT}{1em}{}

\section*{OBJECTIVE OF THE PROJECT}

\begin{enumerate}
  \item \textbf{User-Interactive Mini-Games:} The program provides a
menu-driven interface for users to choose from a variety of mini-games, including
Random Number Guess, Tic-Tac-Toe, Hangman, Rock Paper Scissors, Dice
Simulator, and a Number Pattern game. The objective is to engage users in playing
these games.

  \item \textbf{Game Implementation:} Each mini-game is implemented with its
own game rules and logic. For example:
    \begin{itemize}
```

```
      \item \textbf{Random Number Guess:} The user has to guess a randomly generated number.
      \item \textbf{Tic-Tac-Toe:} A two-player game with a 3x3 board.
      \item \textbf{Hangman:} The user must guess a word by suggesting letters.
      \item \textbf{Rock Paper Scissors:} The user plays against the computer for a defined number of rounds.
      \item \textbf{Dice Simulator:} Simulates rolling a dice with user-specified face numbers.
      \item \textbf{Number Pattern:} The user needs to identify patterns and predict the next number in a sequence.
    \end{itemize}

  \item \textbf{Debugging with pdb:} The code utilizes Python's built-in \texttt{pdb} (Python Debugger) module for debugging. It sets breakpoints using \texttt{pdb.set\_trace()} at various points in the code to facilitate debugging and inspection of variables and program flow. This can help identify and fix issues or improve code functionality.

  \item \textbf{Profiling with cProfile:} The code uses Python's \texttt{cProfile} module to profile the performance of the code. Profiling measures the time taken by different functions in the code, helping to identify bottlenecks and optimize the program.

  \item \textbf{Structured Execution:} The program runs in a structured manner, allowing users to choose and play games repeatedly. Users can play multiple rounds and then decide whether to play again or exit.

  \item \textbf{Loading and Displaying Profiling Statistics:} The code loads profiling data from a file named 'profile\_results.pstats' and displays the profiling statistics in a tabular format, sorted by cumulative time. This allows developers to analyze and optimize code performance.
\end{enumerate}

\titleformat{\section}[block]{\bfseries\filcenter}{OUTPUT}{1em}{}
```

\newpage

\section*{OUTPUT}

\section*{PYTHON MINI GAMES}

Which game would you like to play?

1. Random Number Guess

2. Tic-Tac-Toe

3. Hangman

4. Rock Paper Scissors

5. Dice Simulator

6. Number Pattern

Enter your choice: 1

\section*{RANDOM NUMBER GUESS!}

The number to guess is a 2-digit number

Enter your guess: 50

Too low. Guess again!

Enter your guess: 60

Too low. Guess again!

Enter your guess: 80

```
Too low. Guess again!

Enter your guess: 90

Too high. Guess again!

Enter your guess: 85

Too high. Guess again!

Enter your guess: 83

Right guess!!

Play again? (y/n): y
```

```
\begin{center}
\vspace{1\baselineskip} % Add 1 line of space
\end{center}

\section*{PYTHON MINI GAMES}
```

```
Which game would you like to play?

1. Random Number Guess

2. Tic-Tac-Toe

3. Hangman

4. Rock Paper Scissors

5. Dice Simulator
```

```
6. Number Pattern

Enter your choice: 2

\section*{TIC-TAC-TOE}

\begin{tabular}{c|c|c}
  &   &  \\
\hline
  &   &  \\
\hline
  &   &
\end{tabular}

Player X, enter a position (0-8): 2

\begin{tabular}{c|c|c}
  &   & X \\
\hline
  &   &  \\
\hline
  &   &
\end{tabular}

Player O, enter a position (0-8): 3

\begin{tabular}{c|c|c}
  &   & X \\
\hline
O &   &  \\
\hline
  &   &
\end{tabular}

Player X, enter a position (0-8): 4
```

```
\begin{tabular}{c|c|c}
  &   & X \\
\hline
O & X &  \\
\hline
  &   &
\end{tabular}

Player O, enter a position (0-8): 5

\begin{tabular}{c|c|c}
  &   & X \\
\hline
O & X & O \\
\hline
  &   &
\end{tabular}

Player X, enter a position (0-8): 6

\begin{tabular}{c|c|c}
  &   & X \\
\hline
O & X & O \\
\hline
X &   &
\end{tabular}

Player X wins!

Play again? (y/n): y

\begin{center}
\vspace{1\baselineskip} % Add 1 line of space
```

```
\end{center}

\section*{PYTHON MINI GAMES}

Which game would you like to play?

1. Random Number Guess

2. Tic-Tac-Toe

3. Hangman

4. Rock Paper Scissors

5. Dice Simulator

6. Number Pattern

Enter your choice: 3

\section*{HANGMAN}

\_ \_ \_ \_ \_

Guess a letter: c

Wrong guess! You have 5 attempts left.

\begin{verbatim}
    --------
    |      |
    O       |
           |
           |
           |
\end{verbatim}

\_ \_ \_ \_ \_

Guess a letter: f

Wrong guess! You have 4 attempts left.

\begin{verbatim}
       --------
       |      |
       O       |
       |      |
               |
               |
\end{verbatim}

\_ \_ \_ \_ \_

Guess a letter: a

Wrong guess! You have 3 attempts left.

\begin{verbatim}
       --------
       |      |
       O       |
      /|      |
               |
               |
\end{verbatim}

\_ \_ \_ \_ \_

Guess a letter: d
```

```
Wrong guess! You have 2 attempts left.

\begin{verbatim}
       --------
       |      |
       O      |
      /|\     |
              |
              |
\end{verbatim}

\_ \_ \_ \_ \_

Guess a letter: e

\_ e \_ \_ e

Guess a letter: l

Wrong guess! You have 1 attempt left.

\begin{verbatim}
       --------
       |      |
       O      |
      /|\     |
      /       |
              |
\end{verbatim}

\_ e \_ \_ e

Guess a letter: b
```

Wrong guess! You have 0 attempts left.

\begin{verbatim}

```
  --------
  |      |
  O      |
 /|\     |
 / \     |
         |
```

\end{verbatim}

Out of attempts. The word was: merge

Play again? (y/n): y

\begin{center}
\vspace{1\baselineskip} % Add 1 line of space
\end{center}

\section*{PYTHON MINI GAMES}

Which game would you like to play?

1. Random Number Guess

2. Tic-Tac-Toe

3. Hangman

4. Rock Paper Scissors

5. Dice Simulator

6. Number Pattern

```
Enter your choice: 4

\section*{ROCK PAPER SCISSORS}

How many points do you want to play for? 2

Enter your choice: rock/paper/scissors: rock

You won! The computer chose scissors

Points: You: 1 Computer: 0

Enter your choice: rock/paper/scissors: rock

You won! The computer chose scissors

Points: You: 2 Computer: 0

GAME OVER!

You won! Congratulations!

Your points: 2

Computer's points: 0

Play again? (y/n): y

\begin{center}
\vspace{1\baselineskip} % Add 1 line of space
\end{center}

\section*{PYTHON MINI GAMES}

Which game would you like to play?
```

1. Random Number Guess

2. Tic-Tac-Toe

3. Hangman

4. Rock Paper Scissors

5. Dice Simulator

6. Number Pattern

Enter your choice: 5

\section*{DICE SIMULATOR}

Enter the number of faces on your dice:4

Your random choice is: 1

Play again? (y/n): y

\begin{center}
\vspace{1\baselineskip} % Add 1 line of space
\end{center}

\section*{PYTHON MINI GAMES}

Which game would you like to play?

1. Random Number Guess

2. Tic-Tac-Toe

```
3. Hangman

4. Rock Paper Scissors

5. Dice Simulator

6. Number Pattern

Enter your choice: 6

\section*{Welcome to the Number Pattern Game!}

Find the next term in this Powers of 2 pattern:

[1, 2, 4, 8, 16]

Your answer: 32

Correct!

\begin{center}
\vspace{1\baselineskip} % Add 1 line of space
\end{center}

Game Over!

Your final score: 1

\begin{center}
\vspace{1\baselineskip} % Add 1 line of space
\end{center}

Play again? (y/n): n

\begin{center}
```

```
\vspace{1\baselineskip} % Add 1 line of space
\end{center}

Bye!

\end{document}
```