A Project Report on

# Tax Bracket Calculator

BSCY



TAX BRACKET CALCULATOR

by

Yusra Farooq BCY243041

Abdul Qadeer BCY243064

A Project Report submitted to the DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING

in partial fulfillment of the requirements for the degree of BACHELORS OF SCIENCE IN CYBER SECURITY

2025

# Tax Bracket Calculator



by

Yusra Farooq BCY243041

Abdul Qadeer BCY243064

A Project Report submitted to the DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING

in partial fulfillment of the requirements for the degree of BACHELORS OF SCIENCE IN CYBER SECURITY

## DECLARATION

It is declared that this is an original piece of our own work, except where otherwise acknowledged in text and references. This work has not been submitted in any form for another degree or diploma at any university or other institution for tertiary education and shall not be submitted by us in future for obtaining any degree from this or any other University or Institution.

Yusra Farooq

Reg No. BCY243041

Abdul Qadeer

Reg No. BCY243064

January, 2025

## CERTIFICATE OF APPROVAL

It is certified that the project titled “Tax Bracket Calculator” carried out by Yusra Farooq Reg. No. BCY243041, Abdul Qadeer, Reg. No. BCY243064 under the supervision of Mr. SM Waqas Ayub Shah Capital University of Science & Technology, Islamabad, is fully adequate, in scope and in quality, as a first semister project for the degree of BS Cyber Security.

--------------------------------------

Supervisor:

Mr. SM Waqas Ayub Shah

Lab Engineer

Department of Electrical and Computer Engineering

Faculty of Engineering

Capital University of Science & Technology, Islamabad

--------------------------------------

HoD:

Dr. Noor Mohammad Khan

Professor

Department of Electrical and Computer Engineering

Faculty of Engineering

Capital University of Science & Technology, Islamabad

## ACKNOWLEDGEMENT

We would like to thank “Mr. SM Waqas Ayub Shah” whose guidance helped to complete our report in the given time. Last but not the least, this project cannot be completed without the effort and co-operation of group members.

**Table of Contents**

[DECLARATION iii](#_heading=h.gjdgxs)

[CERTIFICATE OF APPROVAL iv](#_heading=h.30j0zll)

[ACKNOWLEDGMENT v](#_heading=h.1fob9te)

ABSTRACTION………………………………………………………1

[Chapter 1 1](#_heading=h.3znysh7)

[INTRODUCTION 1](#_heading=h.2et92p0)

* 1. [Overview 1](#_heading=h.tyjcwt)
  2. [Project Idea 1](#_heading=h.3dy6vkm)
  3. [Purpose of the Project 1](#_heading=h.1t3h5sf)
  4. [Project Specifications 1](#_heading=h.4d34og8)
  5. [Applications of the Project 2](#_heading=h.2s8eyo1)

[Chapter 2 3](#_heading=h.17dp8vu)

[PROBLEM STATEMENT 3](#_heading=h.3rdcrjn)

[Chapter 3 4](#_heading=h.26in1rg)

[PROJECT DESIGN METHODOLOGY 4](#_heading=h.lnxbz9)

[Chapter 4 7](#_heading=h.35nkun2)

[CODE 7](#_heading=h.1ksv4uv)

[Chapter 5 9](#_heading=h.44sinio)

[PROJECT RESULTS 9](#_heading=h.2jxsxqh)

[Chapter 6 11](#_heading=h.z337ya)

[CONCLUSION AND FUTURE WORK 11](#_heading=h.3j2qqm3)

[Key Takeaways: 11](#_heading=h.1y810tw)

* + 1. [Learning and Growth: 11](#_heading=h.4i7ojhp)
    2. [Impact: 11](#_heading=h.2xcytpi)

[References 1](#_heading=h.1ci93xb)

### Abstraction

The **Tax Bracket Calculator** is a user-friendly C++ program designed to simplify tax computations for individuals. It enables users to calculate their tax liabilities accurately by categorizing income into predefined tax brackets and applying applicable rates. The program takes inputs such as annual income and deductions, validates them, and computes taxable income before determining the total tax owed. This project demonstrates the practical application of core programming concepts like conditional statements, input validation, and formatted output. Beyond its functionality, the Tax Bracket Calculator serves as an educational tool, bridging the gap between theoretical knowledge and real-world applications. The calculator is a step toward making financial planning more accessible and comprehensible, especially for students and individuals without access to costly tax software

.

### Introduction

Tax calculation is an essential yet often misunderstood process for many individuals. Misinterpretation of tax brackets, deductions, and taxable income can lead to errors in tax payments, potentially causing financial strain or penalties. The **Tax Bracket Calculator** project addresses these challenges by offering a simple, reliable, and efficient solution.

This C++ program enables users to determine their tax liabilities based on their annual income and deductions. By incorporating fundamental programming concepts, such as conditional statements and input validation, the calculator provides accurate results in a user-friendly format. The project not only emphasizes financial literacy but also highlights how programming can solve practical problems effectively. With a focus on clarity and precision, the Tax Bracket Calculator bridges the gap between theoretical programming knowledge and real-world applications, making it a valuable tool for students, individuals, and small businesses alike.

## 

## Chapter 1

### INTRODUCTION

#### 1.1 Project Overview

The **Tax Bracket Calculator** is a C++ program designed to assist individuals in calculating their tax liabilities based on their annual income and deductions. By automating the process of tax computation, this tool simplifies complex tax calculations and ensures accurate results. It categorizes the user's income into predefined tax brackets, applying specific tax rates to calculate the total tax owed.

This project highlights the application of fundamental programming concepts, such as conditional statements, loops, and input validation, to address a practical problem. The program not only demonstrates the importance of financial literacy but also provides users with a clear understanding of how deductions and tax brackets affect their taxable income and overall tax burden.

#### 1.2 Project Idea

The idea for this project originated from the need for a straightforward, affordable tool to calculate taxes. Most tax software solutions are either expensive or overly complex, making them inaccessible to a broad audience. The Tax Bracket Calculator, written in C++, fills this gap by offering a simple, efficient, and user-friendly alternative.

By taking user inputs, such as annual income and deductions, and applying predefined tax rules, the program ensures that the calculations are both accurate and easy to understand. The project also emphasizes the educational value of programming, allowing students to apply theoretical knowledge in a real-world context.

#### 1.3 Purpose of the Project

The primary purpose of the Tax Bracket Calculator is to enhance understanding of tax computations while providing users with an accurate and easy-to-use solution for determining their tax liabilities. It aims to:

* Offer a reliable tool for calculating taxes without relying on expensive software or accountants.
* Educate users about how income, deductions, and tax brackets interact to determine taxable income and tax payable.
* Help students develop problem-solving and programming skills by working on a practical application.

#### 1.4 Project Specifications

##### Programming Language:

The program is developed in **C++**, utilizing key programming concepts such as:

* **Conditional Statements:** To determine applicable tax brackets.
* **Input Validation:** To ensure logical and acceptable user inputs.
* **Formatted Output:** Using libraries like <iomanip> to display results with precision and clarity.

##### User Inputs:

1. **Annual Income:**
   * The total income earned by the user in a year.
2. **Deductions:**
   * Allowable deductions such as savings, medical expenses, or other tax-exempt items.

##### Calculations:

1. **Taxable Income:**
   * Calculated using the formula:  
     **Taxable Income = Annual Income - Deductions**
2. **Tax Brackets and Rates:**
   * The program applies the following tax rates based on taxable income:
     + **Bracket 1:** Up to PKR 1,000,000 → **No Tax**
     + **Bracket 2:** PKR 1,000,001 - PKR 500,0000 → **20%**
     + **Bracket 3:** Above PKR 500,0000 → **50%**

##### Output:

The program provides the following outputs to the user:

1. **Taxable Income:** The income remaining after deductions are subtracted.
2. **Tax Amount:** The total tax calculated based on the identified tax bracket.
3. **Deductions:** The total allowable deductions entered by the user.

#### 1.5 Applications of the Project

##### 1. Financial Planning:

The Tax Bracket Calculator is an effective tool for users to plan their finances better. It enables them to:

* Understand how taxable income is calculated by accounting for deductions.
* Get an accurate estimate of their tax liabilities, helping them make informed financial decisions.

##### 2. Educational Purpose:

This project serves as a learning platform for students by:

* Simplifying the concepts of tax brackets, taxable income, and deductions.
* Demonstrating how real-world problems can be solved using basic programming constructs.
* Helping students develop problem-solving skills and improve their understanding of C++ concepts like loops, conditionals, and formatted output.

##### 3. Practical Use:

The program has a wide range of practical applications, such as:

* **For Individuals:**
  + Assisting individuals in personal tax computation.
  + Eliminating reliance on expensive software or accountants for basic tax calculations.
* **For Small Businesses:**
  + Helping small businesses estimate taxes for employees or operational income.
  + Providing a cost-effective alternative to complex accounting software.
* **For Broader Accessibility:**
  + Offering a simple, affordable, and portable tax calculation solution for a wide range of users.

## 

## Chapter 2

### 

### PROBLEM STATEMENT

Calculating taxes can be a complex and tedious task for many individuals. Misunderstanding tax brackets and deductions often leads to errors or overpayment. The need for an affordable, straightforward tool for accurate tax calculation is evident. This project aims to address this gap by developing a user-friendly program that simplifies the tax calculation process.

## Chapter 3

### PROJECT DESIGN METHODOLOGY

The Tax Bracket Calculator was designed to follow a structured methodology, ensuring accuracy and user-friendliness. The following steps describe the key components of the program's workflow:

### 1. User Input and Validation

* The program begins by prompting the user to input two key details:
  + **Annual Income**: The total income earned by the user over a year.
  + **Deductions**: Allowable deductions such as medical expenses, savings, or other tax-exempt items.
* To ensure the program handles only valid data, both inputs are validated:
  + **Annual Income Validation**: The program ensures that the income entered is non-negative. If the user enters a negative value, they are prompted to re-enter the income until a valid value is provided.
  + **Deductions Validation**: Similar to income, deductions are also validated to ensure they are non-negative. This prevents incorrect or illogical calculations.
* This input validation improves the program's robustness by preventing runtime errors and ensuring logical results.

### 2. Tax Bracket Identification

* Once the inputs are validated, the program calculates the **taxable income** using the formula:  
  **Taxable Income = Annual Income - Deductions**
* The program then identifies the applicable tax bracket for the taxable income using a series of conditional statements:
  + If taxable income is less than or equal to **PKR 1,000,000**, no tax is applied.
  + If taxable income is greater than **PKR 1,000,000** but less than or equal to **PKR 5000,000**, a tax rate of **20%** is applied.
  + For taxable income above **PKR 5000,000**, a tax rate of **50%** is applied.
* These tax brackets are implemented through nested conditional statements (if-else blocks), ensuring that the correct rate is applied based on the user’s taxable income.

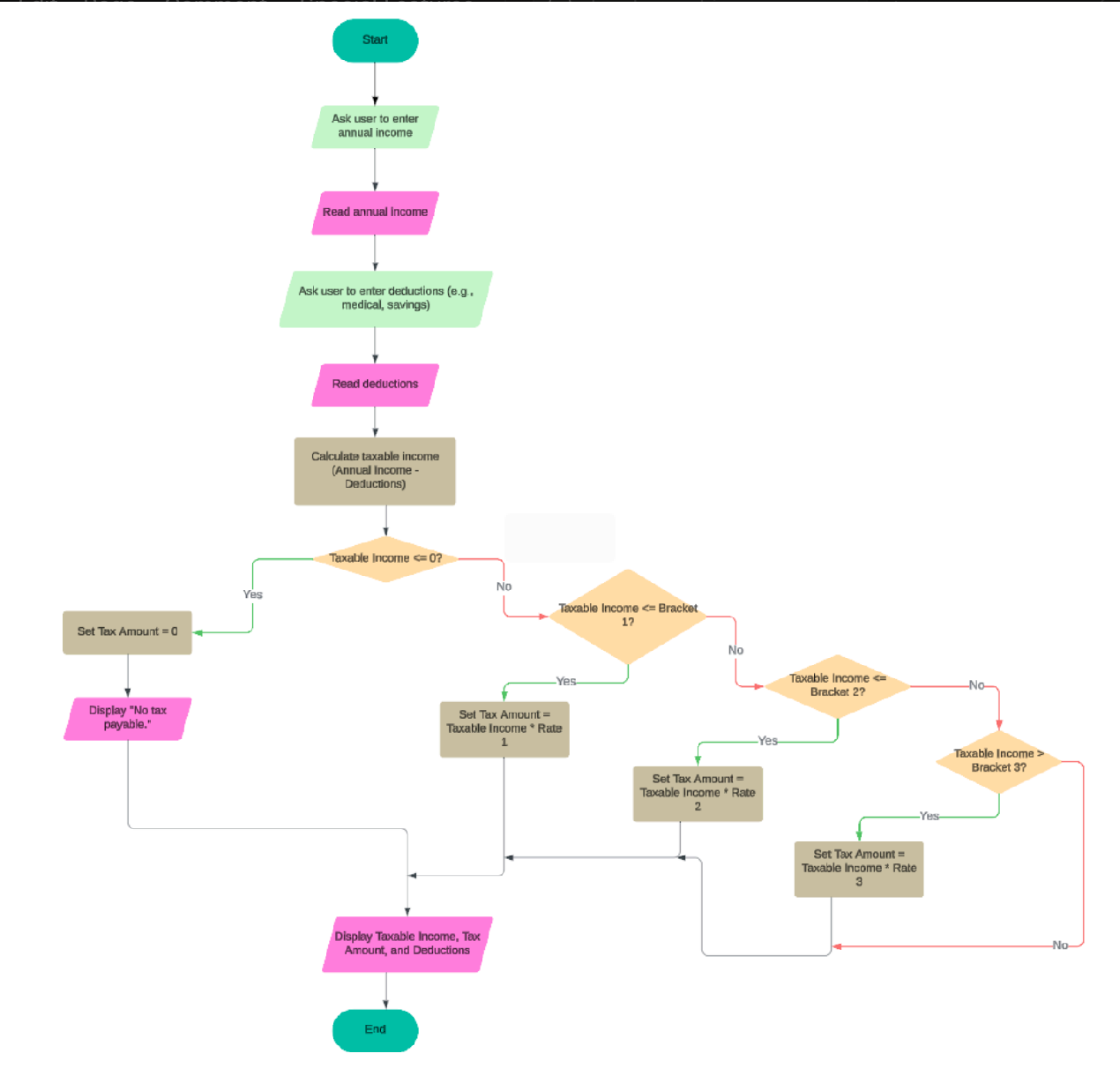
### 3. Tax Calculation

* Once the tax bracket is identified, the program calculates the tax amount using the formula:  
  **Tax Amount = Taxable Income × Tax Rate**
* The calculation considers different rates for different brackets, ensuring accurate results:
  + For instance, if the taxable income falls in **Bracket 2**, the program calculates tax at **20%** for the entire taxable income.
  + Similarly, if the taxable income exceeds **Bracket 3**, the tax rate of **50%** is applied to the amount above that bracket threshold.
* The program also accounts for cases where deductions reduce taxable income to zero or below, ensuring that no tax is applied and the user is informed accordingly.

### 4. Output Display

* After the tax calculations, the program displays the results to the user in a clear and organized format. The output includes:
  + **Taxable Income**: The income remaining after deductions are applied.
  + **Total Tax Amount**: The tax amount computed based on the identified tax bracket.
  + **Deductions**: The total deductions entered by the user.
* The output is formatted using the iomanip library, ensuring that monetary values are displayed with two decimal places for clarity and precision.
* In cases where no tax is applicable (e.g., taxable income ≤ 1,000,000), the program explicitly informs the user by displaying a message such as:  
  **"No tax payable."**

**Flowchart:**

****

## Chapter 4

### CODE

#include <iostream>

#include <iomanip>

using namespace std;

double cal\_taxable\_income(double annual\_income, double deductions)

{

double taxable\_income;

double tax\_amount = 0;

const double bracket1 = 100000;

const double bracket2 = 500000;

const double rate1 = 0.2;

const double rate2 = 0.3;

const double rate3 = 0.5;

// Calculating Taxable Income

taxable\_income = annual\_income - deductions;

if (taxable\_income < 0) {

taxable\_income = 0;

}

if (taxable\_income <= 100000)

{

tax\_amount = 0; // No tax payable

cout << "No tax Payable" << endl;

}

else if (taxable\_income <= bracket1)

{

tax\_amount = taxable\_income \* rate1;

}

else if (taxable\_income <= bracket2)

{

tax\_amount = taxable\_income \* rate2;

}

else

{

tax\_amount = taxable\_income \* rate3;

}

cout << fixed << setprecision(2); // Set fixed-point notation with 2 decimal places

// cout << "Taxable Income: " << annual\_income - deductions << endl;

cout << "Taxable Income: " << taxable\_income << endl;

cout << "Deductions: " << deductions << endl;

return tax\_amount;

}

int main()

{

double annual\_income, deductions;

cout << "Enter your Annual Income: ";

cin >> annual\_income;

// Validate annual income

while (annual\_income < 0) {

cout << "Annual income cannot be negative. Please enter a valid amount: ";

cin >> annual\_income;

}

cout << "Enter Your Deductions (for e.g: savings, medical): ";

cin >> deductions;

// Validate deductions

while (deductions < 0) {

cout << "Deductions cannot be negative. Please enter a valid amount: ";

cin >> deductions;

}

double tax\_amount = cal\_taxable\_income(annual\_income, deductions);

cout << "Tax Amount: " << tax\_amount << endl;

return 0;}

## 

## Chapter 5

### 

### PROJECT RESULTS

This chapter demonstrates the functionality of the Tax Bracket Calculator with sample inputs, outputs, and detailed calculations based on the implemented code. The program takes the user’s annual income and deductions as input, calculates the taxable income, determines the applicable tax bracket, and computes the tax amount using predefined rates.

### Input Example 1

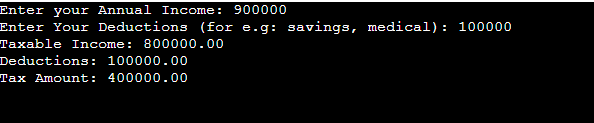
* **Annual Income:** PKR 900,000
* **Deductions:** PKR 100,000

#### Step-by-Step Calculation

1. **Taxable Income Calculation  
   Taxable Income = Annual Income - Deductions  
   Taxable Income = PKR 100,000 - PKR 20,000 = PKR 80,000**
2. **Tax Bracket Identification**
   * According to code:
     + If taxable income is less than or equal to PKR 1,000,000 → No tax is applied.
   * **Result:** Taxable Income (PKR 80,000) falls under the **No Tax** bracket.
3. **Tax Calculation**
   * **Tax Amount = PKR 0 (No tax payable)**

#### Output

* **Taxable Income:** PKR 100,000
* **Deductions:** PKR 20,000
* **Tax Amount:** PKR 0



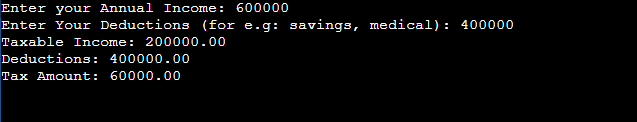
Input Example 2

* **Annual Income:** PKR 600,000
* **Deductions:** PKR 200,000

#### Step-by-Step Calculation

1. **Taxable Income Calculation  
   Taxable Income = Annual Income - Deductions  
   Taxable Income = PKR 600,000 - PKR 200,000 = PKR 400,000**
2. **Tax Bracket Identification**
   * According to your code:
     + Taxable Income (PKR 1,300,000) falls under **Bracket 2**, where the tax rate is 30%.
3. **Tax Calculation**
   * **Tax Amount = PKR 60,000**

#### Output

* **Taxable Income:** PKR 400,000
* **Deductions:** PKR 200,000
* **Tax Amount:** PKR 60,000

### Input Example 3

* **Annual Income:** PKR 1,500,000
* **Deductions:** PKR 200,000

#### Step-by-Step Calculation

1. **Taxable Income Calculation  
   Taxable Income = Annual Income - Deductions  
   Taxable Income = PKR 1,500,000 - PKR 200,000 = PKR 1,300,000**
2. **Tax Bracket Identification**
   * According to your code:
     + Taxable Income (PKR 1,300,000) falls under **Bracket 3**, where the tax rate is 50%.
3. **Tax Calculation**
   * **Tax Amount = Taxable Income × 50%  
     Tax Amount = PKR 1,300,000 × 0.5 = PKR 650,000**

#### 

#### 

#### Output

* **Taxable Income:** PKR 1,300,000
* **Deductions:** PKR 200,000
* **Tax Amount:** PKR 650,000

A black screen with white text

Description automatically generated

### 

### Summary of Corrected Results

1. **Input Example 1:**
   * Taxable Income: PKR 100,000
   * Deductions: PKR 20,000
   * Tax Amount: PKR 0
2. **Input Example 2:**
   * Taxable Income: PKR 400,000
   * Deductions: PKR 200,000
   * Tax Amount: PKR 60, 000
3. **Input Example 3:**
   * Taxable Income: PKR 1,300,000
   * Deductions: PKR 200,000
   * Tax Amount: PKR 650,000

**Observations from Results**

* **Validation of Inputs:** The program handles invalid inputs like negative values for annual income and deductions, ensuring only valid data is processed.
* **Tax Bracket Application:** The program correctly applies progressive tax rates based on the taxable income, ensuring accurate calculations.
* **Error-Free Output:** All results are displayed clearly, with appropriate formatting and precision.

**Conclusion of Results**

The Tax Bracket Calculator successfully processes various cases, calculating taxable income and tax amounts accurately. The program’s logic correctly handles different tax brackets and edge cases, making it a reliable tool for tax calculations.

**Chapter 6**

### CONCLUSION AND FUTURE WORK

#### Conclusion

The Tax Bracket Calculator project showcases the practical application of programming concepts in addressing real-world challenges. By leveraging the simplicity and power of the C++ programming language, the tool provides a user-friendly interface for calculating taxes based on progressive tax brackets. It enables users to determine their taxable income, identify their applicable tax bracket, and compute their total tax accurately.

The project highlights key areas of learning, such as input validation, conditional logic, and numerical computations. It also underlines the importance of financial literacy by giving users a clear understanding of how their income and deductions affect their tax obligations.

Through this project, we gained valuable experience in designing, implementing, and testing a real-world application, enhancing our coding and problem-solving skills. Additionally, the project demonstrates how technology can simplify complex tasks, making it accessible to individuals without prior knowledge of tax systems.

Overall, the Tax Bracket Calculator serves as a reliable and efficient solution for personal tax computation and offers a strong foundation for future enhancements.

#### Future Work

While the current version of the Tax Bracket Calculator meets its initial objectives, there are several avenues for improvement and expansion to increase its utility and scope:

1. **Integration with Mobile and Web Applications**:
   * Developing mobile and web-based versions of the calculator to make it more accessible to users on various platforms.
   * Adding user-friendly graphical interfaces for a better user experience.
2. **Inclusion of Advanced Tax Laws**:
   * Incorporating additional features to support complex tax scenarios, such as:
     + Tax reliefs for specific investments.
     + Tax exemptions for senior citizens or certain income groups.
     + Handling tax rebates for charitable donations or education expenses.
3. **Support for Multiple Users or Small Businesses**:
   * Expanding the system to support multiple user profiles, allowing families or small businesses to calculate taxes for all members or employees in one place.
   * Adding functionality for business-related deductions and expenses.
4. **Currency and Region Customization**:
   * Allowing the program to adapt to different countries’ tax systems by providing customizable tax brackets and rates.
   * Supporting multiple currencies with automatic conversions based on real-time exchange rates.
5. **Integration with Financial Tools**:
   * Linking the calculator with personal finance management tools to provide a comprehensive overview of income, expenses, and taxes.
   * Offering insights into savings strategies to reduce taxable income.
6. **Advanced User Features**:
   * Introducing a feature to generate detailed tax reports that can be saved or printed for record-keeping.
   * Adding tax forecasting capabilities to help users plan their finances for the upcoming year.
7. **AI-Based Recommendations**:
   * Utilizing artificial intelligence to analyze user input and provide recommendations to optimize tax savings.
   * Suggesting deductions or investments that could lower taxable income.

#### 

#### Impact of Future Enhancements

The proposed improvements would significantly enhance the usability, versatility, and scope of the Tax Bracket Calculator. By integrating modern features like AI, mobile compatibility, and advanced tax rules, the tool can cater to a wider audience, including professionals, small business owners, and international users.

The project represents a stepping stone toward bridging the gap between technology and personal finance, demonstrating how software solutions can simplify intricate processes and empower users to make informed financial decisions.

## REFERENCES

1. **Federal Board of Revenue (FBR) Pakistan**
   * Official tax brackets and rates can be found on the Federal Board of Revenue's website.
   * **Website:**<https://www.fbr.gov.pk>
2. **GeeksforGeeks C++ Tutorials**
3. **Programming Basics – W3Schools**