Web-Based Income Tax Calculator: Design, Implementation, and Functionality

John Carlo J. Sinoy  
Computer Programming Student  
Red Deer PolytechnicRed Deer, Alberta  
johncarlo.sinoy@rdpolytech.ca

John Gerardo  
Computer Programming Student  
Red Deer PolytechnicRed Deer, Albertajohn.gerardo@rdpolytech.ca

*Abstract*— This project implements a web-based income tax calculator that enables users to input their annual income and calculate their tax liability based on specified tax brackets. The system offers additional functionalities, including support for tax deductions and a reset button. Built using HTML, CSS, and jQuery, the application features a simple, user-friendly interface.

Keywords—Income Tax Calculator, Tax Brackets, Tax Decutions, JavaScript, HTML, CSS

# introduction

Income tax calculation is essential for individuals and organizations to determine the amount of tax payable based on their earnings. This program simplifys the process of computing the taxes and optionally considers deductions.

ⅠⅠ. METHODOLOGY

## Technologies Used

* **HTML**: Used for creating the structure and layout of the web application.
* **CSS**: Used for styling the interface and providing a user-friendly design.
* **JQuery**: Used to handle dynamic content updates, event handling, and tax calculation logic.

## Design Approach

The income tax calculator is structured into three components:

1. **Input Section**: Users input their annual income and any tax deductions.
2. **Button**: A "Calculate Tax" button initiates the tax calculation process and a “Reset Button” to clear all input fields
3. **Result Display**: The computed tax is displayed in a separate section for the user.

## Tax Brackets

Tax is Calculated after subtracting any deductions from the income. The tax is calculated based on the following tax brackets:

* **0%** for income up to $10,000
* **10%** for income from $10,001 to $40,000
* **20%** for income from $40,001 to $100,000
* **30%** for income over $100,000

A screenshot of a tax calculator

Description automatically generated

1. *Tax Calculator preview*

## Functional Flow

1. The user enters their income and any deductions.
2. Upon clicking the "Calculate Tax" button, the input is validated.
3. The income after deductions is computed.
4. Based on the tax brackets, the tax is calculated and displayed.

ⅠⅠⅠ. IMPLEMENTATION

* 1. *HTML*

The HTML serves as the structural foundation for the web-based income tax calculator. It defines the layout, elements, and overall content that users interact with.

**Structure and Components:**

1. **Document Declaration and Head Section**:
   * The HTML document begins with a <!DOCTYPE html> declaration, ensuring that the page follows the HTML5 standard.
   * The <head> section contains metadata, such as the document's character set (UTF-8), viewport settings for responsiveness, and the link to the external CSS file (style.css). The title of the webpage is set using the <title> tag.
2. **Body Section**:
   * The main content of the tax calculator is enclosed within the <body> tag. The calculator interface is placed inside a <div> element with the class container. This container groups the various form elements for styling and layout purposes.
3. **Form Elements**:
   * **Input Fields**: The calculator includes two <input> elements where users enter their annual income and any deductions. These fields have attributes such as id, type="number", and placeholder, making it easy for users to input values.
     + id="income" allows the jQuery to target this specific element for capturing the user's income.
     + id="deductions"similarly captures any deductions.
   * **Button**: A <button> element with the id="calculateBtn" is used to trigger the tax calculation. The button allows the user to submit their input values and request a tax calculation. Another button with the id=”resetBtn” that allows the user to delete all their previous entries and the result.
4. **Result Display**:
   * The result of the tax calculation is shown in a <div> element with the id="result". This section will be dynamically updated by the jQuery code to display the calculated tax amount after the user clicks the button.
   1. *jQuery*

The jQuery script captures user input, applies the tax brackets based on the taxable income (after deductions), and displays the calculated tax.

The jQuery part of this project manages the interaction between the user’s input and the actual tax calculation. The entire calculation process is triggered by a user event, when the user clicks the “Calculate Tax” button, the program captures the input values, processes them, calculates the tax, and then displays the result on the screen.

**Event Handling with jQuery**

When the document is fully loaded ($(document).ready(function())), the program sets up an event listener for when the user clicks the "Calculate Tax" button ($('#calculateBtn').click(function())). This event listener is triggered every time the button is clicked.

**Capturing User Input**

Once the button is clicked, the program retrieves the values the user entered in the input fields for both income and deductions. jQuery uses the val() method to get the values from the input elements. Since input values are returned as strings, the parseFloat() function is used to convert them into numbers.

* **Income**: This is the gross income entered by the user.
* **Deductions**: These are the tax deductions, which are subtracted from the income to reduce the taxable amount. If the user doesn’t enter anything for deductions, it defaults to 0 using || 0 to ensure the program runs smoothly.

**Input Validation**

Input validation is crucial to ensure that users enter valid data. If the income is not a valid number (left empty or set to 0), the program will display an error message instead of attempting to calculate the tax. This is handled using a simple conditional check:

if (isNaN(income) || income <= 0) {

$('#result').html('Please enter a valid income.');

return;

If the deduction is set to a negative number an error message will also occur that will prevent any further calculations

**Calculating Taxable Income**

Once the income and deductions are retrieved, the program calculates the taxable income by subtracting the deductions from the income. Taxable income is the portion of the income that will be subjected to tax, after deductions have been accounted for.

If Income subtracted by the Deductions is 0 or any number less than zero, the message “Your tax is $0 after deductions” will display as a result.

**Tax Calculation Based on Brackets**

The next step is to determine how much tax is owed based on the taxable income. This is done by applying tax rates for different income brackets using a series of conditional statements (if-else).

**Displaying the Result**

Once the tax is calculated, the result is displayed dynamically on the webpage using jQuery’s html() method. This method updates the content of the #result div with the computed tax amount.

The toFixed(2) method ensures that the tax amount is always displayed with two decimal places, which is standard for representing currency values.

* 1. *CSS*

The file is used to style the HTML content, giving the income tax calculator a simple, clean, and user-friendly appearance. It ensures that the web page is visually appealing and easy to navigate.

**Key Styling Elements:**

1. **Body Styling**:

The body section uses display: flex, justify-content: center, and align-items: center to center the calculator on the screen, creating a clean, minimalist design. This approach also makes the page responsive across different screen sizes. The background-color of the body is set to a neutral light grey (#f0f0f0) to provide a subtle contrast against the white container.

1. **Container Styling**:

The .container class is applied to the main block that houses the input fields and button. It has a white background with padding (15px), a small border-radius (5px), and a light box-shadow to add a slight 3D effect, making the calculator stand out against the page background.

1. **Input Fields**:

The input fields are styled to span 100% of the width of the container. Padding and margin adjustments ensure that the inputs are user-friendly, with enough spacing between the text and the input borders. The border of the input fields is a light grey to match the overall minimalist theme.

1. **Button Styling**:

The “Calculate” button uses a solid blue background color (#007bff), with white text for high contrast. The button is designed to span 100% of the container width, ensuring it is easy to click and noticeable. On hover, the button changes color to a slightly darker blue (#0056b3), providing visual feedback to the user when they interact with it. Same design principles apply to the “Reset” button.

1. **Result Display**:
   * The result section is kept minimal with larger font sizes to ensure the calculated tax is clear and easy to read. The margin above the result ensures there is enough space between the input fields and the result display.

#### **Design Choices**:

* **Minimalist Approach**: The design is intentionally simple to avoid overwhelming the user. The interface uses neutral colors and basic styling for a professional and clean look.
* **Responsiveness**: Using percentages for widths ensures that the calculator adapts to various screen sizes. The CSS ensures that the application remains usable on both desktop and mobile devices.
* **User Feedback**: The button hover effect provides a subtle but important visual indication that the button is clickable, enhancing the user experience.

Ⅳ. *Conclusion*

The income tax calculator provides an efficient solution for quickly calculating tax liability based on income and deductions. With its simple interface and clear results, users can easily understand how much tax they owe. The tool is extendable and can incorporate additional tax rules or features such as saving past calculations or providing detailed tax breakdowns.

Ⅴ. *Future Work*

Possible future enhancements could include:

* Implementing local storage to save previous calculations.
* Adding tax breakdowns for visual representation.

A screenshot of a tax calculator

Description automatically generatedA screenshot of a tax calculator

Description automatically generatedA screenshot of a tax calculator

Description automatically generatedA screenshot of a calculator

Description automatically generatedA screenshot of a tax calculator

Description automatically generatedA screenshot of a tax calculator

Description automatically generated

Fig. 7. *Calculation for Income in 30% tax bracket*

Fig. 6. *Calculation for Income in 20% tax bracket*

Fig. 5. *Calculation for Income in 10% tax bracket*

Fig. 4. *Calculation for Income in 0% tax bracket*

Fig. 3. *Tax Calculator with Inputs*

Fig. 2. *Tax Calculator without Inputs*

A screen shot of a tax calculator

Description automatically generatedA screen shot of a tax calculator

Description automatically generatedA screenshot of a tax calculator

Description automatically generatedA screenshot of a tax calculator

Description automatically generatedA screenshot of a tax calculator

Description automatically generated

Fig. 11.2. *Income is less than/equal to tax deductions.*

Fig. 11. *Income is less than/equal to tax deductions.*

Fig. 10. *Negative number for Deduction input box*

Fig. 9. *Negative number for Income input box*

Fig. 8 . *Calculation for Income in 30% tax bracket with deductions that drops the income to the 20% tax bracket*

*(See Fig. 7 for comparison)*