FIJI YEAR 12 CERTIFICATE COMPUTER PROJECT

2024

COMMON ASSESSMENT TASK (CAT)

NAME: DHRUV KUMAR

YEAR: 12A

TEACHER: MR. ASHNEEL KUMAR

PENANG SANGAM HIGH SCHOOL

INDEX: 857643

DEVELOPING A SIMPLE PAYROLL SYSTEM FOR BULA CARS RENTAL COMPANY USING VB.NET

AIM:

Students will use the knowledge from CE12.2.1 VisualBasic.net to develop a program on payroll and write a report on it.

OBJECTIVE:

- Discuss the difference between a software developer and a software engineer.
- Create the program on payroll calculation of Bula Cars rentals that should accept:
  - Employees name
  - Hourly wage rate
  - Hours worked
- The program should calculate the wages and show the employee's name, hours worked, hourly wage rate and the total wage for a week.
- Document each step to the creation of the program:
  1. Program specification
  2. Program design
  3. Program code
  4. Program test
  5. Program documentation
- Create a user interface using visual basic
- Discuss the benefits of using an automated payroll system compared to using a manual system.

ACKNOWLEDGEMENT:

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this project.

- First and foremost, I would like to thank God for giving me the strength, knowledge, ability and the opportunity to undertake this project and to complete it.
- I would like to express my special gratitude to my teacher, Mr Ashneel Kumar, for his invaluable guidance, encouragement, and support throughout this project. His knowledge and expertise in this field greatly enriched my understanding and learning.
- I am also immensely grateful to my parents, whose love, care, and sacrifices have been a constant source of motivation for me. Their unwavering faith in my abilities has always pushed me to strive for excellence.
- Lastly, I would like to thank my colleagues who contributed their time and effort in assisting me with this project. Their insights and solidarity made the project not only more manageable but also a truly enjoyable experience.

Thank you all for your immense support and encouragement.

I certify that all the information presented in this project is correct and relevant to the best of my knowledge.

Name: Dhruv Kumar

Signature: D. Kumar

Date: 12th July 2024

METHODOLOGY:

Information and data were obtained from various primary and secondary sources for the task commutation.

Primary Source:

- The web was used to gather information for this project: Microsoft Bing and Google Scholar.

Secondary source:

- Year 11 and year 12 textbooks were used to get the relevant information for the completion of this project.

TABLE OF CONTENT

CONTENTS

INTRODUCTION

Payroll is a critical part of the operation of an organisation in paying employees their salaries on time. The idea of taking control of employees' pay calculations is quite tedious if done manually and requires more effort and time mainly for big organisations. Hence, if this process is automated, it would be of great benefit as it would require less time to calculate the wages of the employees. The proposed payroll system is advantageous as it provides a user-friendly environment, increases security, and minimises human calculation errors.

To begin with, the current project deals with designing an automated payroll calculation system for Bula Cars rental company using visual basic programming for the calculation of wages for the employees weekly as they are currently using manual pay calculation techniques. According to **A. Kawale et al. (2018)**, the manual calculation of pay needs to be reassessed due to changes in need and changes in technology so an organisation needs a payroll system that is efficient and relevant to the organisation’s needs. Hence, the automated payroll system will reduce the time spent on pay calculation by employees.

Furthermore, in this project, a computerised payroll calculation system is designed using V.B .net for calculating pay, which shows the employee's name, hours worked, hourly pay rates, and the total weekly wages that need to be paid. According to **Palladan & Palladan (2018),** when employees' payroll information is entered and processed efficiently, the entire organisation will run smoothly.

To end now, the new computerised payroll calculation system will be made using the five stages of program development to accurately execute the payroll system that will assure the employees and the management that the pay calculation process is seamless and accurate, decreasing time spent on manual calculations and reducing calculation errors for better efficiency of the organisation.

SOFTWARE DEVELOPERS & SOFTWARE ENGINEERS

Software Developers

A software developer is a professional responsible for designing, coding, testing, and maintaining software applications or systems. According to **Tobin (2024),** they primarily build programs and apps for consumer markets based on user needs. Their role involves understanding user requirements, translating them into technical specifications, and implementing solutions using programming languages and development tools. A software developer's primary responsibility is to ensure that the software meets quality standards, is user-friendly, and functions efficiently. To qualify for this role, one typically needs a bachelor's degree in computer science, software engineering, or a related field, strong programming skills and knowledge of software development methodologies. Depending on factors such as experience, location, and industry, salaries for software developers can vary widely, generally ranging from $60,000 to $150,000 per year, with opportunities for advancement and higher earnings with experience and expertise.

Roles

1. **Coding:** Writing and testing code to develop software applications.
2. **Problem Solving:** Analysing user needs and designing software solutions to meet those requirements.
3. **Collaboration:** Working with team members (like designers, other developers, and testers) to ensure the successful deployment of software.
4. **Testing and Debugging:** Identifying and fixing issues within the codebase to ensure smooth functionality.
5. **Maintenance and Updates:** Updating software and addressing user feedback or issues post-launch to improve performance and usability.

Responsibilities

1. **Designing and Developing Software:** Creating robust and scalable software solutions based on user requirements and technical specifications.
2. **Testing and Debugging:** Ensuring the quality of the software by testing and debugging code to identify and fix errors and issues.
3. **Documentation:** Write technical documentation that explains how the software works, algorithms, and system architecture.
4. **Collaboration and Communication:** Working closely with other team members (such as designers project managers and testers) to coordinate effort and ensure smooth project execution.
5. **Continuous learning and Improvement:** Keeping up-to-date with the latest technologies, tools and best practices in software development to improve skills and contribute effectively to projects.

Software Engineers

Software engineers design, develop, and maintain software systems and applications. They play a critical role in identifying user needs, creating software solutions, and ensuring they are scalable, efficient, and maintainable. Responsibilities include coding, testing, debugging, and collaborating with other team members to deliver high-quality software. A bachelor's degree in computer science, software engineering, or a related field is typically required, along with proficiency in programming languages and software development methodologies. The salary range for software engineers varies widely based on experience, location, and industry, but generally falls between $70,000 and $150,000 annually.

Roles

1. **Developer:** Writing and testing code to create software applications and systems.
2. **System Architect:** designing the overall structure of software systems to ensure they meet requirements and are scalable and maintainable.
3. **Quality Assurance:** Ensuring software meets quality standards through rigorous testing and debugging processes.
4. **Project Manage:** Overseeing software projects, coordinating tasks, and ensuring timely delivery within budget.
5. **Support and Maintenance:** Providing ongoing support, troubleshooting issues, and implementing updates and improvements to existing software.

Responsibilities

1. **Coding and development:** writing clean, efficient, and maintainable code to build software applications and systems.
2. **Requirement analysis:** Collaborating with stakeholders to understand and define software requirements and specifications.
3. **Testing and debugging:** Conducting tests to identify and fix bugs to ensure the software functions correctly.
4. **Design and architecture:** Creating software architecture and design plans that outline the structure and components of the software.
5. **Documentation and maintenance:** Maintaining comprehensive documentation of code and systems, and providing ongoing maintenance and updates to ensure the software remains functional and up to date.

Difference Between Software Developer and Software Engineer

Difference between software developers and software engineers in a table form:

	Software developers	Software engineers
Definition	Software developers design specific computer systems and application software.	Software engineers work on a larger scale to design, develop, and test entire computer systems and application software for a company or organisation
Scope	Focus more on the actual coding and building of software applications. They work on specific parts of the software development process, primarily writing and implementing code.	Typically have a broader scope, focusing on the overall architecture and design of software systems. They are involved in the entire software development lifecycle, including planning, development, testing, deployment, and maintenance.

Role	Software developers may find themselves working in a more independent environment. - Focuses on the creation of software applications, including writing, debugging, and maintaining code. - Typically involved in the implementation phase of the software development life cycle.	Software engineers may work with many different people on a team, including other engineers and developers. - Takes a broader and more systematic approach to software development, considering the entire software development life cycle. - Often involved in the planning, designing, and architecture of software systems.
Responsibilities	- Writing clean, efficient, and maintainable code. - Debugging and troubleshooting software issues. - Collaborating with other team members like designers, testers, and other developers. - Ensuring that the software meets the specified requirements and functions correctly.	- Designing and creating complex software systems and frameworks. - Applying engineering principles to software development, ensuring scalability, maintainability, and robustness. - Conducting requirements analysis and feasibility studies. - Overseeing and guiding the development process, including coordinating with other engineering disciplines and project stakeholders. - Ensuring that software systems integrate well with existing systems and comply

		with industry standards.
Skills	• **Programming Languages:** proficient in languages such as Java, Python, C#, JavaScript, etc. • **Soft Skills:** Creativity, problem-solving, communication, and the ability to work in a team. • **Specific skills:** user interface design, and user experience design.	• **Programming Languages**: Similar to developers, with a deeper understanding of how these languages work. • **Soft Skills**: Analytical thinking, systematic problem-solving, project management, and strong communication skills. • **Specific Skills**: Systems design, algorithm optimisation, software architecture, and sometimes knowledge of hardware.
Qualification	• **Education**: Typically, a bachelor's degree in computer science, software development, or a related field. Some positions may accept boot camp certifications or self-taught individuals with strong portfolios. • **Certifications**: Optional but beneficial certifications in specific programming languages or frameworks.	• **Education**: A bachelor’s degree in computer science, software engineering, or a related engineering field is often required. Advanced degrees (master’s or PhD) can be advantageous. • **Certifications**: Professional certifications in software engineering principles (e.g., IEEE Certified Software Development Professional).
Salary	They have a slightly lower salary range, typically between $65,000 and $120,000 per year, corresponding to their focus on coding and application development.	Generally, they have a higher salary range, often between $70,000 and $150,000 per year, reflecting their broader scope and engineering responsibilities.

FIVE STEPS OF SOFTWARE DEVELOPMENT

STEP 1: PROGRAM SPECIFICATION

Program Specification is also called Program Definition or Program Analysis. In this step, the programmer has to determine the objectives, outputs, inputs, and processing requirements and document them. It requires the programmer to complete these tasks. They are:

- ❖ **Determining the program objectives:** the initial step is to characterise the issue to address the program objective. For this situation, designing a simple payroll calculation system for an effective and error-free pay calculation for Bula Cars Rental Company.

- ❖ **Determining the desired output:** while focusing on the desired outcome, consider the necessary inputs. Therefore, the employer needs to add the necessary details to get the pay calculated and after the calculation the employee's name, hours worked, hourly wage rate and total weekly wages will be displayed.

- ❖ **Determining the input data required:** in this step, the employee needs to enter the data that consists of the employee’s name, hourly wage rate, and the hours worked, ensuring accurate results.

- ❖ **Determining the processing requirements:** in this step, there is need to convert input into output must be identified. Through coding, the payroll calculator will determine the output with the use of data entered and display the total weekly wages.

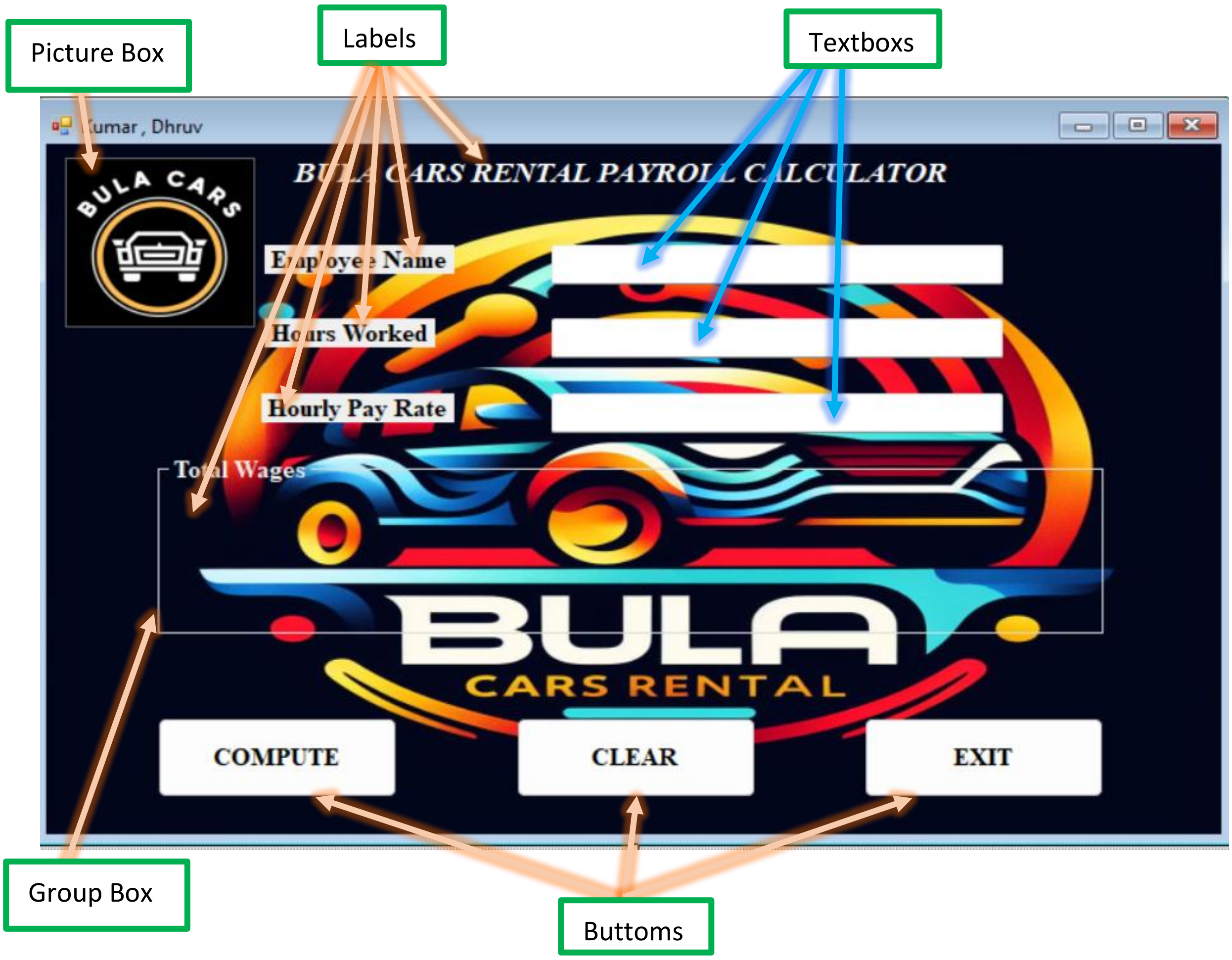
- ❖ **Documenting the program specifications:** The final task involves creating a comprehensive specification document that outlines the processes necessary for generating the objective outputs, detailing inputs, and outlining processing steps. This document will encompass all coding-based methods utilised in developing the payroll calculator.

STEP 2: PROGRAM DESIGN

In the process of program design, a structured approach is crucial for creating effective solutions. This involves careful planning before commencing with the actual coding in a specific programming language. Well-documented procedures in this phase ensure that future maintenance of the program is straightforward for future developers.

Recently, I was tasked with developing a payroll calculator to streamline wage calculations for Bula Cars Rental company. This venture began with creating a comprehensive diagram illustrating the calculator's operational flow. Pseudo code was then drafted to outline the logic and operations of the program. Subsequently, the user interface was meticulously designed using VB.NET, using essential controls such as labels, textboxes, picture boxes, buttons, and group boxes to structure the payroll calculator's layout. This structured approach ensures clarity and efficiency during implementation but also minimises errors, facilitating accurate computation of the weekly wages for employees.

INTERFACE DESIGN FOR BULA CARS RENTAL COMPANY PAYROLL CALCULATOR



PROPERTIES TABLE

Objects	Properties	Settings
Form1	Name BackgroundImage BackgroundImageLayout BackColor Font FontStyle FontSize ForeColor Text	Form1 System.Drawing.Bitmap Stretch Control Microsoft Sans Serif Regular 8 ControlText Kumar , Dhruv
Label1	Name BackColor Font FontStyle FontSize ForeColor Text	Label1 Transparent Times New Roman Bold Italic 14 White BULA CARS RENTAL PAYROLL CALCULATOR
PictureBox1	Name BoderStyle Background image layout Image SizeMode	PictureBox1 FixedSingle Tile System.Drawing.Bitmap StretchImage
Label2 (Employee Name)	Name BackColor Font FontStyle FontSize ForeColor Text	Label2 Control Times New Roman Bold 12 ControlText Employee Name

Text box (Employee Name)	Name	txtEmployeeName
	BackColor	Window
	Font	Times New Roman
	FontStyle	Regular
	FontSize	12
	ForeColor	WindowText
	Text	
Label3 (Hours Worked)	Name	Label3
	BackColor	Control
	Font	Times New Roman
	FontStyle	Bold
	FontSize	12
	ForeColor	ControlText
	Text	Hours Worked
Text box (Hours Worked)	Name	txtHoursworked
	BackColor	Window
	Font	Times New Roman
	FontStyle	Regular
	FontSize	12
	ForeColor	WindowText
	Text	
Label4 (Hourly Pay Rate)	Name	Label4
	BackColor	Control
	Font	Times New Roman
	FontStyle	Bold
	FontSize	12
	ForeColor	ControlText
	Text	Hourly Pay Rate

Text box (Hourly Pay Rate)	Name BackColor Font FontStyle FontSize ForeColor Text	txtHourlyPayRate Window Times New Roman Regular 12 WindowText
Display Total Pay (Label)	Name BackColor Font FontStyle FontSize ForeColor Text	lblDisplayTotalPay LemonChiffon Times New Roman Bold 12 ActiveCaptionText
GropuBox1	Name BackColor Font FontStyle FontSize ForeColor Text	GroupBox1 Transparent Times New Roman Bold 12 ButtonFace Total Wages
Compute Button	Name BackColor Font FontStyle FontSize ForeColor Text	btnCompute Control Times New Roman Bold 12 ControlText COMPUTE

Clear Button	Name	btnClear
	BackColor	Control
	Font	Times New Roman
	FontStyle	Bold
	FontSize	12
	ForeColor	ControlText
	Text	CLEAR
Exit Button	Name	btnExit
	BackColor	Control
	Font	Times New Roman
	FontStyle	Bold
	FontSize	12
	ForeColor	ControlText
	Text	EXIT

PSEUDOCODE

Pseudocode serves as an informal, high-level description of a computer program or an algorithm's functionality. According to **Sheldon (2023),** pseudocode is a detailed readable description of what a program should do. It adheres to the structural conventions of a standard programming language but is designed primarily for human understanding rather than machine execution.

The primary objective of employing pseudo-code is to render programming logic intelligible to human readers. It functions akin to a preliminary program skeleton that can be accurately translated into a specific programming language. Pseudocode offers a straightforward method of outlining program logic using plain language, facilitating the formulation of executable statements that yield desired outcomes.

It's important to note that pseudo-code is not a programming language but serves as a precursor to actual coding in a specific language. Once the program's purpose and operational flow are comprehended, pseudo code aids in structuring the logic and sequence of operations.

Advantages of pseudo-code include its ability to focus developers exclusively on refining algorithmic aspects during the initial stages of code development. Moreover, being language-independent, pseudo code enables clear communication of program design using accessible, everyday language. This characteristic simplifies modifications to the program's structure as needed.

However, pseudo-code does have drawbacks. Its lack of standardised style or format can lead to variations across different instances of pseudo-code, potentially complicating its interpretation. Additionally, readers may find it challenging to follow the logical flow at each step due to its inherently unstructured nature. Lastly, the process of crafting pseudo-code can be time-consuming.

In conclusion, pseudo-code is a fundamental tool in program design, sketching out program structures before the actual coding begins. Its universal clarity makes it accessible to developers across various programming backgrounds, facilitating effective communication and design modification.

PSEUDOCODE WRITING

```
Start Procedure
    Input Employee Name, Hours Worked, Hourly Pay Rate
    If (Hours Worked<=40) then
    Total Wages = (Hourly Pay Rate * Hours Worked)
    Else
    Total Wages = (Hourly Pay Rate*40) + ((Hours Worked-40) *(Hourly Pay Rate*1.5))
    Display Employee Name, Hours Worked, Hourly Pay Rate, Total Wages
End Procedure
```

FLOWCHART

Flowcharts are essential tools in computer science program design, providing visual representations that outline the logical flow of processes and algorithms. They serve to break down complex procedures into manageable steps, offering a structured approach to designing software. By visually mapping out how data moves through different stages of computation and decision-making, flowcharts help programmers identify potential issues early in the design phase. This pre-emptive insight allows for iterative refinement and optimisation, ensuring the eventual software is robust and efficient.

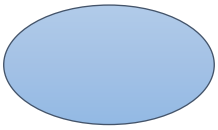
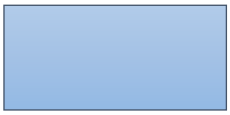
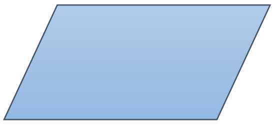
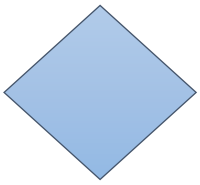
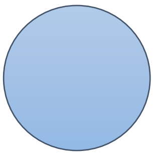
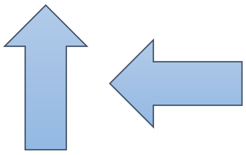
The primary function of flowcharts is to enhance clarity and coherence in program development. They facilitate effective communication among team members and stakeholders by providing a clear overview of the program's structure and functionality. This visual clarity aids in troubleshooting, maintenance, and future updates to the software, serving as valuable documentation throughout the program's lifecycle.

Flowcharts offer several advantages, including their universal accessibility. They can be understood by developers and stakeholders regardless of their programming expertise or background, fostering collaboration across multidisciplinary teams. Additionally, flowcharts streamline the design process by enabling programmers to visualise and refine algorithms before implementation. This process reduces errors and ensures the program efficiently meets its intended objectives.

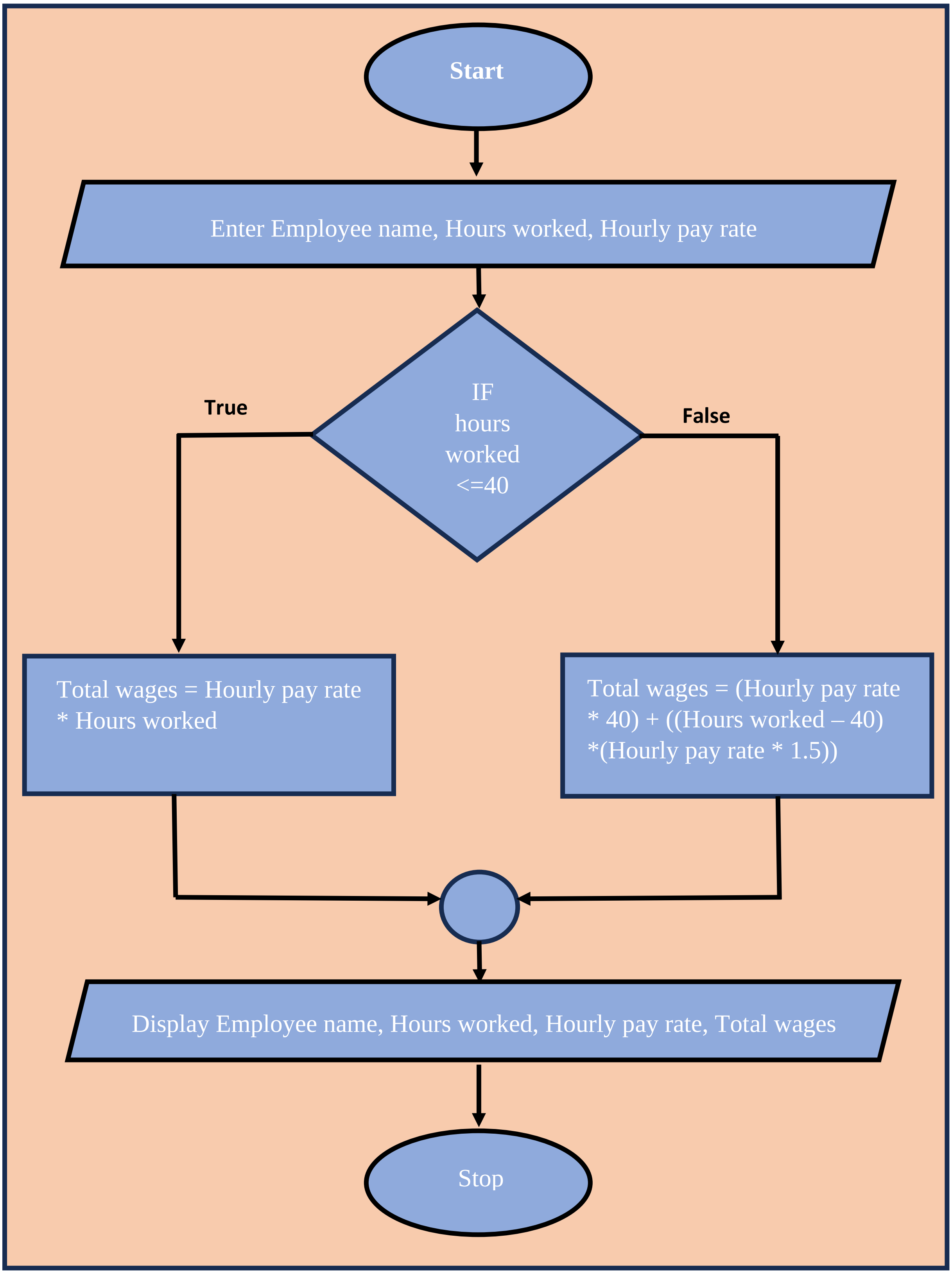
Despite their benefits, flowcharts have limitations. Constructing comprehensive flowcharts can be time-consuming, especially for complex programs. Moreover, clarity and precision in designing flowcharts are crucial; ambiguous or overly complex diagrams can lead to misinterpretations or errors during implementation.

In conclusion, flowcharts are indispensable in computer science program design, providing a visual roadmap that simplifies complex concepts and promotes effective communication. They play a pivotal role in designing robust and efficient software solutions, contributing to the overall success and usability of computer programs. However, their effectiveness depends on clear construction and ongoing maintenance to accurately reflect the program's logic and functionality throughout its lifecycle.

FLOW CHART SYMBOL DESCRIPTION

Symbols	Name	Description
	Terminal	Depicts the beginning and ending steps of the sequence.
	Process	A rectangle is used to represent arithmetic and memory portion of the computer.
	Input/ Output	A parallelogram is used to indicate the input and output steps of the sequence.
	Decision	A diamond is used to represent the use of the computers logic and control unit which must decide whether a statement is true or false.
	Connector	A connector indicates the ending of a loop; it is also used to combine two flow lines.
	Flow lines	A flow line shows the sequence in which direction to be flowed.

PROGRAM DESIGN: FLOWCHART FOR BULA CARS RENTAL COMPANIES PAYROLL CALCULATOR


Start
Enter Employee name, Hours worked, Hourly pay rate
IF hours worked <=40
True
False
Total wages = Hourly pay rate * Hours worked
Total wages = (Hourly pay rate * 40) + ((Hours worked – 40) *(Hourly pay rate * 1.5))
Display Employee name, Hours worked, Hourly pay rate, Total wages
Stop

STEP 3: PROGRAM CODE

Programming code consists of precise instructions designed for creating computer programs. Despite operating within the same domain, they serve distinct purposes. Source code, also known as program code, is the tangible display of these instructions. Conversely, coding denotes the process of translating program designs into reality using a programming language. This phase involves either writing with traditional tools or typing on a keyboard, composing the alphanumeric characters and symbols that constitute the program.

Several attributes characterise a well-crafted program:

- **Reliability:** Ensuring consistent functionality across various scenarios.

- **Error Handling:** Addressing common input errors to enhance robustness.

- **Documentation:** Thoroughly documenting code to facilitate comprehension by other programmers.

Employing structured programming and diverse logical constructs is pivotal for developing efficient programs. Learning computer programming confers numerous advantages. Notably, it is highly valued by employers, imparting invaluable problem-solving skills applicable across diverse career paths.

PROGRAM CODE FOR BULA CARS RENTAL COMPANY PAY ROLL CALCULATOR

COMPUTE BUTTON CODE

```
Private Sub btnCompute_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCompute.Click
        Dim Name As String
        Dim HoursWorked As Double
        Dim HourlyPayRate As Double
        Dim Wages As Double

        Name = txtEmployeeName.Text
        HoursWorked = Double.Parse(txtHoursWorked.Text)
        HourlyPayRate = Double.Parse(txtHourlyPayRate.Text)

        If HoursWorked <= 40 Then
            Wages = HoursWorked * HourlyPayRate
        Else
            Wages = (HourlyPayRate * 40) + ((HoursWorked - 40) *
(HourlyPayRate * 1.5))

        End If
        lblDisplayPay.Text = Name & " has worked for " & HoursWorked & "
hrs with an hourly payrate of $ " & HourlyPayRate & "." &
Environment.NewLine & Name & "s Total Wages is $" & Wages

    End Sub
```

CLEAR BUTTON CODE

```
Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
        txtEmployeeName.Clear()
        txtHourlyPayRate.Clear()
        txtHoursWorked.Clear()
        lblDisplayPay.Text = ""

    End Sub
```

EXIT BUTTON CODE

```
Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        Me.Close()
    End Sub
```

STEP 4: PROGRAM TEST

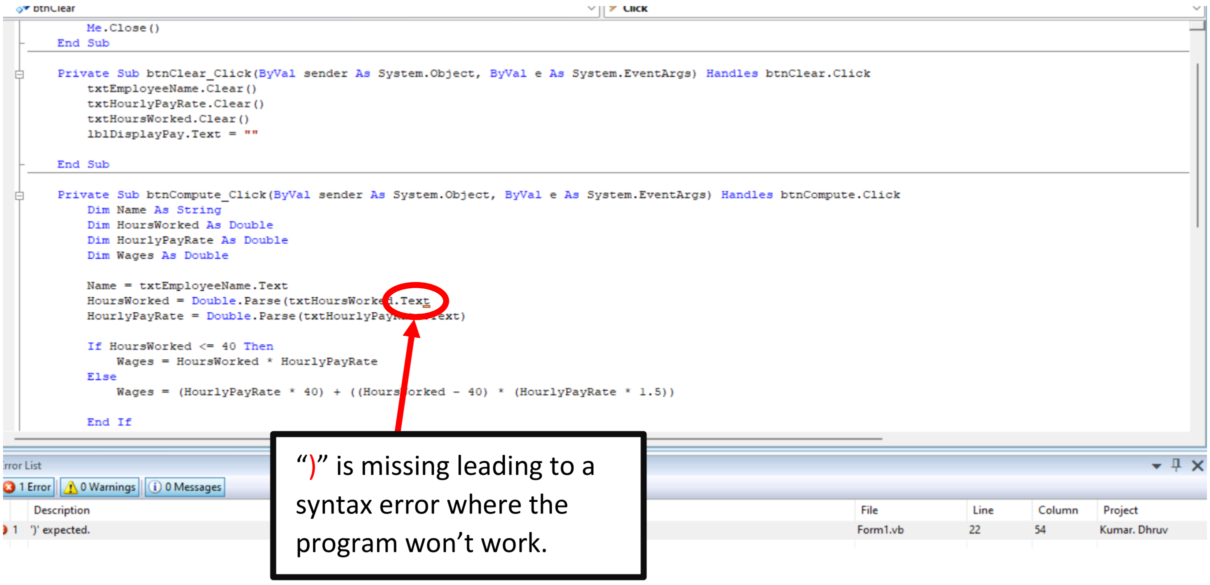
Program testing is crucial in software development, ensuring that programs function correctly and reliably. It involves systematically executing the software with various inputs to uncover errors or unexpected behaviours. By validating the program against its specifications, testing helps mitigate risks associated with defects, enhances user satisfaction, and optimises resource allocation by catching issues early in the development cycle.

In computer science, programmers play a pivotal role in program testing. Beyond coding, they design and implement test cases to thoroughly evaluate their programs. Programmers utilise testing frameworks and methodologies to assess different aspects of the code, from individual components (unit testing) to interactions between modules (integration testing). Their expertise in debugging and interpreting test results enables them to identify and rectify issues efficiently, ensuring the software meets quality standards and performs reliably. Thus, programmers' involvement in testing is essential for delivering high-quality software solutions that meet user expectations.

In addition, I was required to test the payroll calculator to ensure that the correct output was according to the given relevant data and that the program functioned as expected and was free from any error to ensure that the program was operable and user-friendly.

(A bug is an error that prevents the program code from functioning as expected.)

- **Syntax errors-** A syntax error is a mistake in the structure of a sentence or code that violates the rules of the language, making it unable to be parsed or understood correctly by the interpreter or compiler.

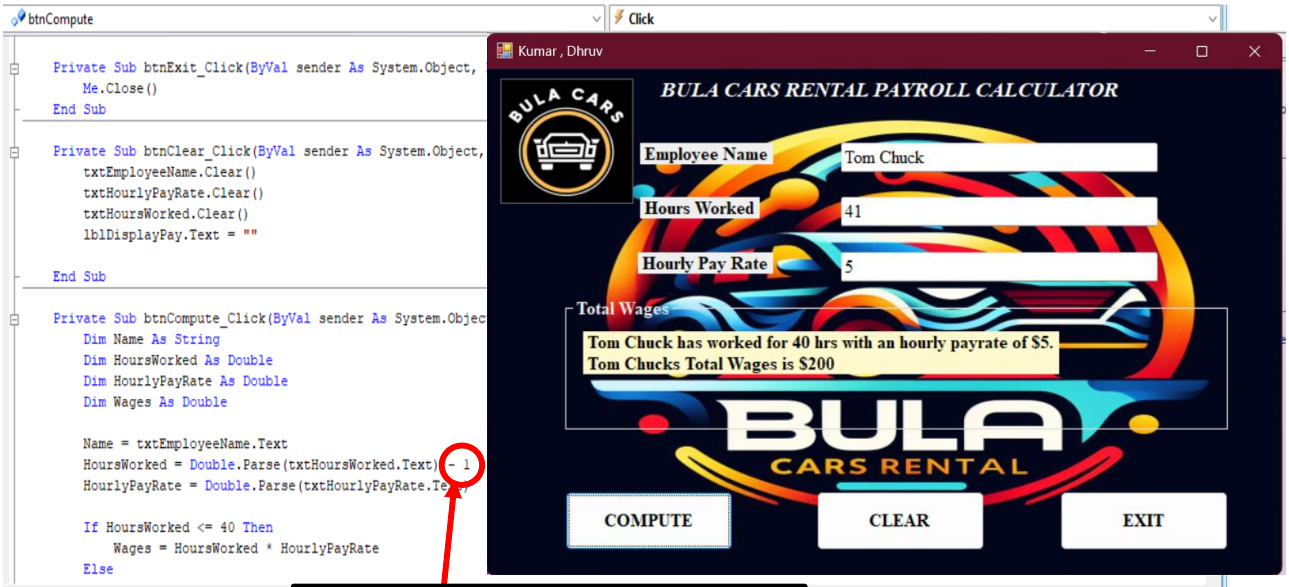


```
        Me.Close()
    End Sub

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
        txtEmployeeName.Clear()
        txtHourlyPayRate.Clear()
        txtHoursWorked.Clear()
        lblDisplayPay.Text = ""

    End Sub

    Private Sub btnCompute_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCompute.Click
        Dim Name As String
        Dim HoursWorked As Double
        Dim HourlyPayRate As Double
        Dim Wages As Double

        Name = txtEmployeeName.Text
        HoursWorked = Double.Parse(txtHoursWorked.Text
        HourlyPayRate = Double.Parse(txtHourlyPayRate.Text)

        If HoursWorked <= 40 Then
            Wages = HoursWorked * HourlyPayRate
        Else
            Wages = (HourlyPayRate * 40) + ((HoursWorked - 40) * (HourlyPayRate * 1.5))

        End If
```

	Description	File	Line	Column	Project
1	')' expected.	Form1.vb	22	54	Kumar. Dhruv

"**)**" is missing leading to a syntax error where the program won't work.

- **A logic error-** refers to a mistake in the algorithm or reasoning used to solve a problem, leading to unintended or incorrect results in the program's output.



```
    Private Sub btnExit_Click(ByVal sender As System.Object,
        Me.Close()
    End Sub

    Private Sub btnClear_Click(ByVal sender As System.Object,
        txtEmployeeName.Clear()
        txtHourlyPayRate.Clear()
        txtHoursWorked.Clear()
        lblDisplayPay.Text = ""

    End Sub

    Private Sub btnCompute_Click(ByVal sender As System.Objec
        Dim Name As String
        Dim HoursWorked As Double
        Dim HourlyPayRate As Double
        Dim Wages As Double

        Name = txtEmployeeName.Text
        HoursWorked = Double.Parse(txtHoursWorked.Text) - 1
        HourlyPayRate = Double.Parse(txtHourlyPayRate.Text)

        If HoursWorked <= 40 Then
            Wages = HoursWorked * HourlyPayRate
        Else
```

Kumar , Dhruv

BULA CARS RENTAL PAYROLL CALCULATOR

Employee Name: Tom Chuck

Hours Worked: 41

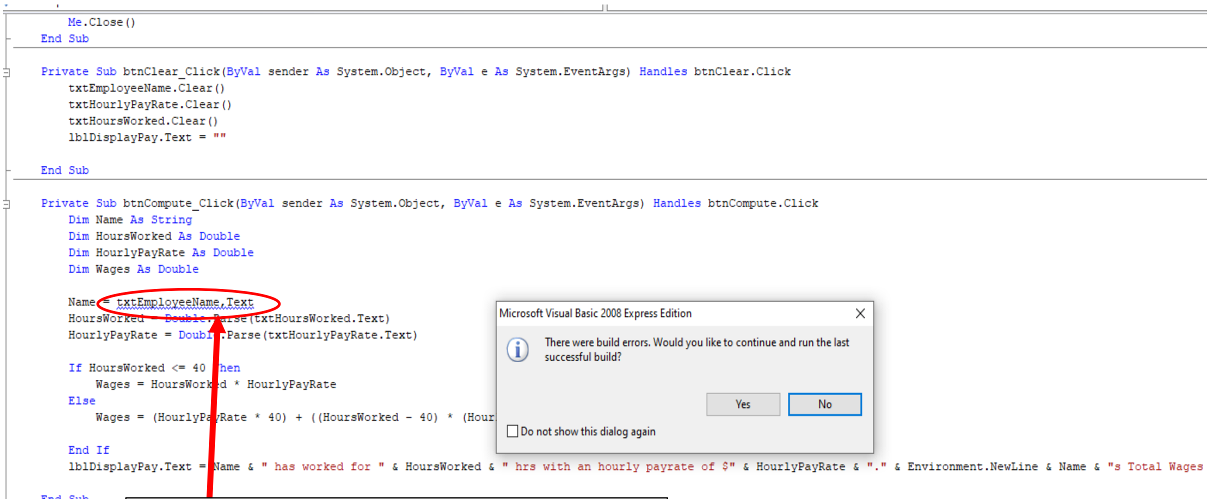
Hourly Pay Rate: 5

Total Wages

Tom Chuck has worked for 40 hrs with an hourly payrate of $5.
Tom Chucks Total Wages is $200

The simple logic error is introduced by always subtracting 1 from “HoursWorked”, which will lead to incorrect wage calculations.

- **Run-time error-** occurs while trying to run the program.



```
        Me.Close()
    End Sub

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
        txtEmployeeName.Clear()
        txtHourlyPayRate.Clear()
        txtHoursWorked.Clear()
        lblDisplayPay.Text = ""

    End Sub

    Private Sub btnCompute_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCompute.Click
        Dim Name As String
        Dim HoursWorked As Double
        Dim HourlyPayRate As Double
        Dim Wages As Double

        Name = txtEmployeeName,Text
        HoursWorked = Double.Parse(txtHoursWorked.Text)
        HourlyPayRate = Double.Parse(txtHourlyPayRate.Text)

        If HoursWorked <= 40 Then
            Wages = HoursWorked * HourlyPayRate
        Else
            Wages = (HourlyPayRate * 40) + ((HoursWorked - 40) * (Hour

        End If
        lblDisplayPay.Text = Name & " has worked for " & HoursWorked & " hrs with an hourly payrate of $" & HourlyPayRate & "." & Environment.NewLine & Name & "s Total Wages

    End Sub
End Class
```

Microsoft Visual Basic 2008 Express Edition

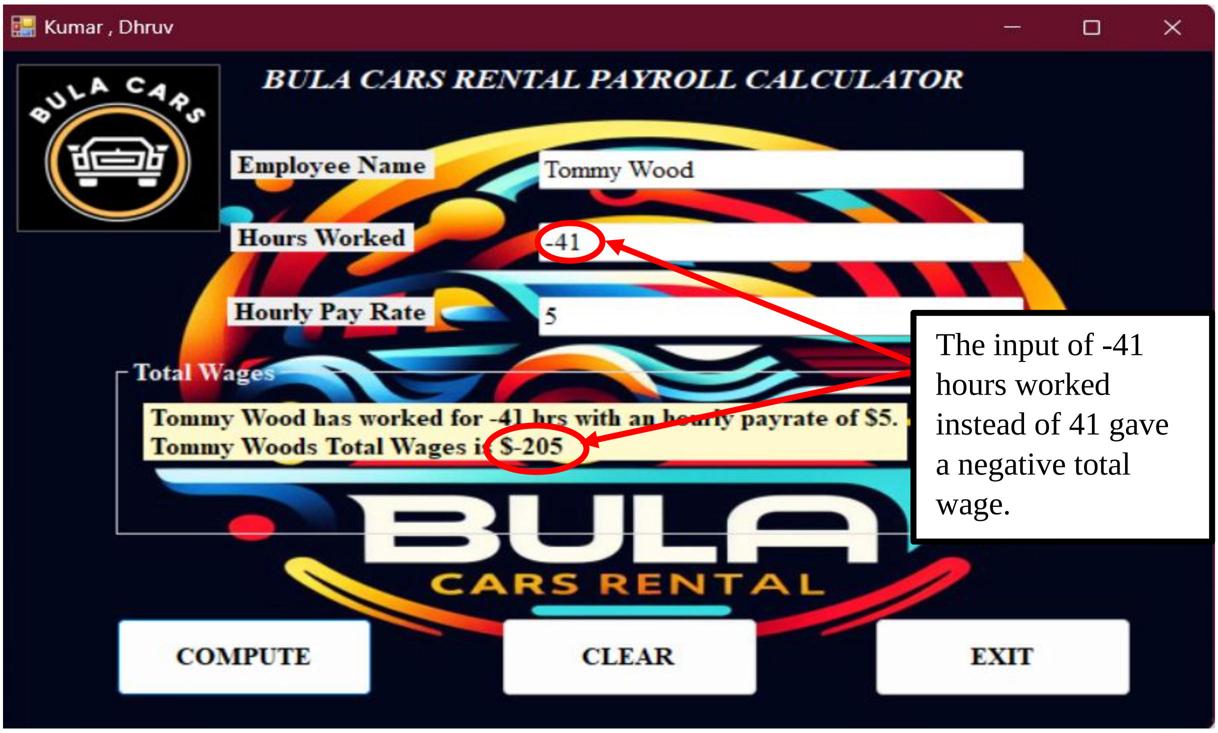
There were build errors. Would you like to continue and run the last successful build?

Yes No

□ Do not show this dialog again

A runtime error occurred as the above should be “txtEmployeeName. Text” where the “,” should have been “.”

- **Invalid input error-** occurs when provided input does not meet the expected criteria such as type, format, range, or constraints.



Employee Name: Tommy Wood

Hours Worked: -41

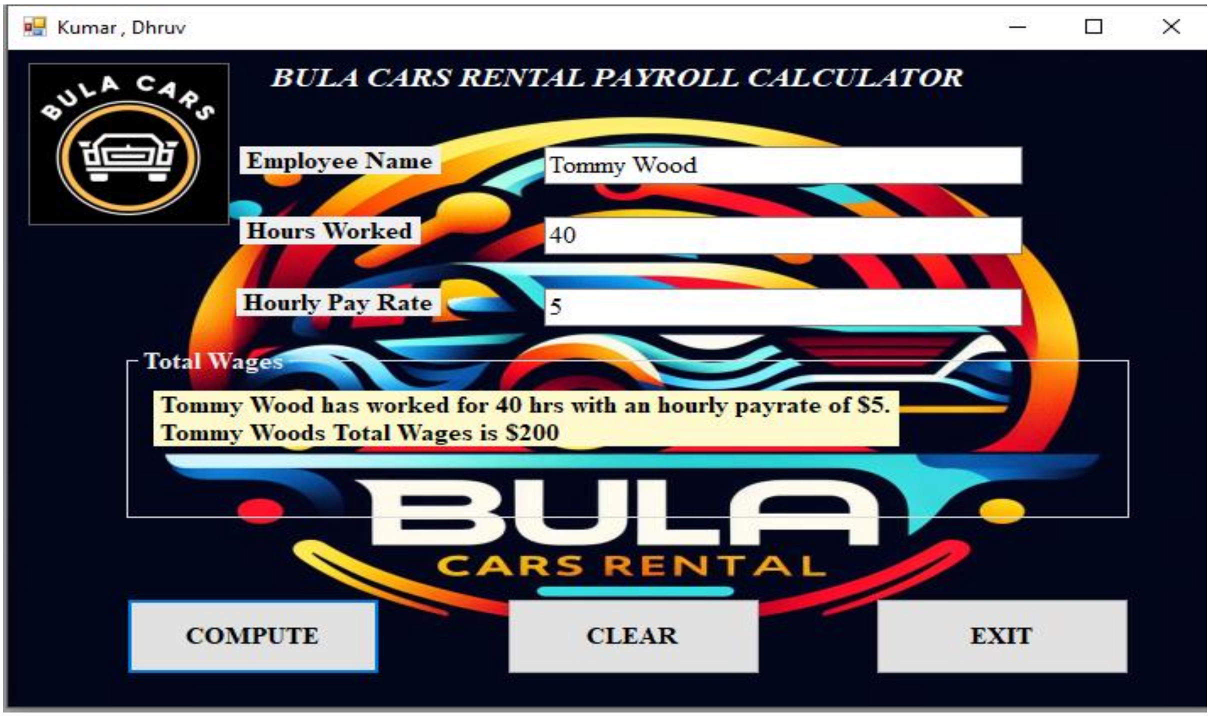
Hourly Pay Rate: 5

Tommy Wood has worked for -41 hrs with an hourly payrate of $5.
Tommy Woods Total Wages is $-205

The input of -41 hours worked instead of 41 gave a negative total wage.

PROGRAM TEST OF PAYROLL CALCULATION FOR BULA CARS RENTAL COMPANY:

Test 1 (Total wages<=40)



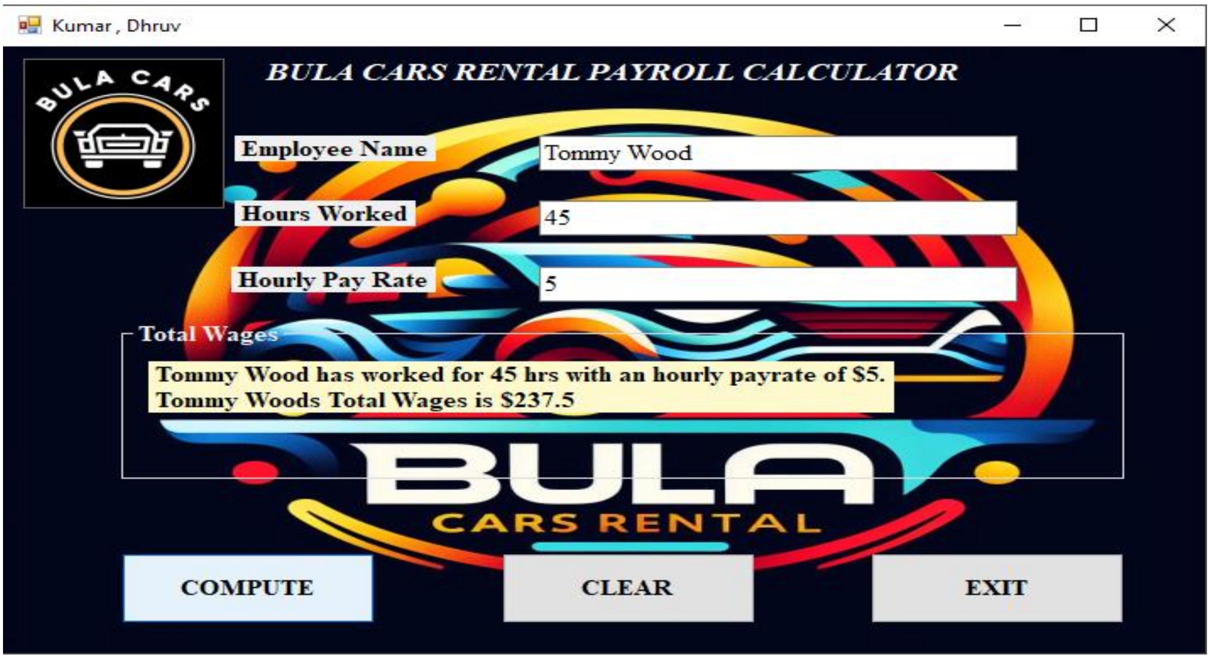
Employee Name: Tommy Wood

Hours Worked: 40

Hourly Pay Rate: 5

Tommy Wood has worked for 40 hrs with an hourly payrate of $5.
Tommy Woods Total Wages is $200

Test 2 (Total wages>40)



Employee Name: Tommy Wood

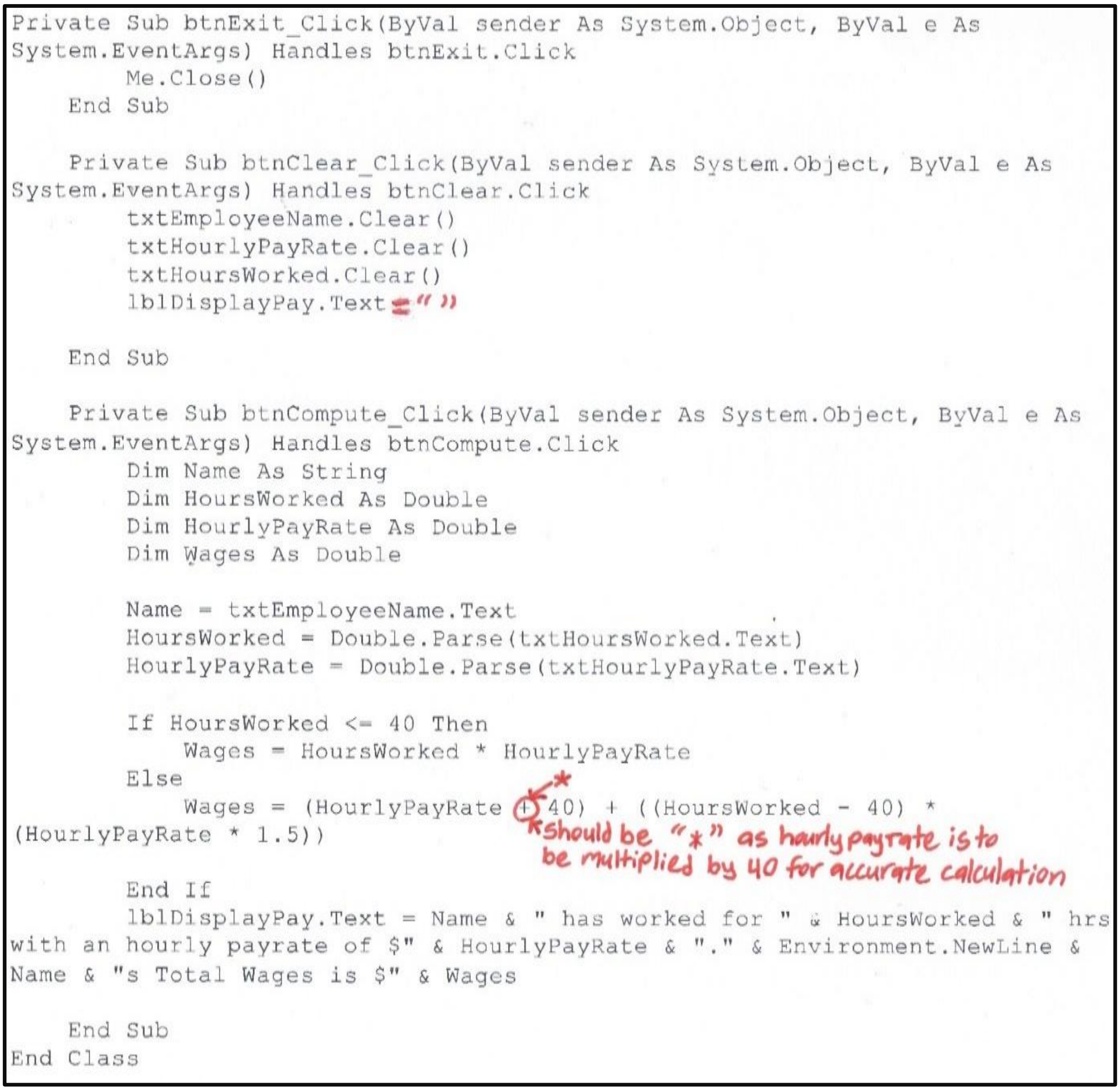
Hours Worked: 45

Hourly Pay Rate: 5

Tommy Wood has worked for 45 hrs with an hourly payrate of $5.
Tommy Woods Total Wages is $237.5

Desk checking test

Desk checking is the manual process of reviewing and tracing through code or algorithms step-by-step on paper to identify errors or logical flaws before running the program.



```
Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        Me.Close()
    End Sub

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
        txtEmployeeName.Clear()
        txtHourlyPayRate.Clear()
        txtHoursWorked.Clear()
        lblDisplayPay.Text= ""

    End Sub

    Private Sub btnCompute_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCompute.Click
        Dim Name As String
        Dim HoursWorked As Double
        Dim HourlyPayRate As Double
        Dim Wages As Double

        Name = txtEmployeeName.Text
        HoursWorked = Double.Parse(txtHoursWorked.Text)
        HourlyPayRate = Double.Parse(txtHourlyPayRate.Text)

        If HoursWorked <= 40 Then
            Wages = HoursWorked * HourlyPayRate
        Else
            Wages = (HourlyPayRate + 40) + ((HoursWorked - 40) *
(HourlyPayRate * 1.5))

        End If
        lblDisplayPay.Text = Name & " has worked for " & HoursWorked & " hrs
with an hourly payrate of $" & HourlyPayRate & "." & Environment.NewLine &
Name & "s Total Wages is $" & Wages

    End Sub
End Class
```

*

should be "*" as hourly payrate is to be multiplied by 40 for accurate calculation

Manual test with sample data

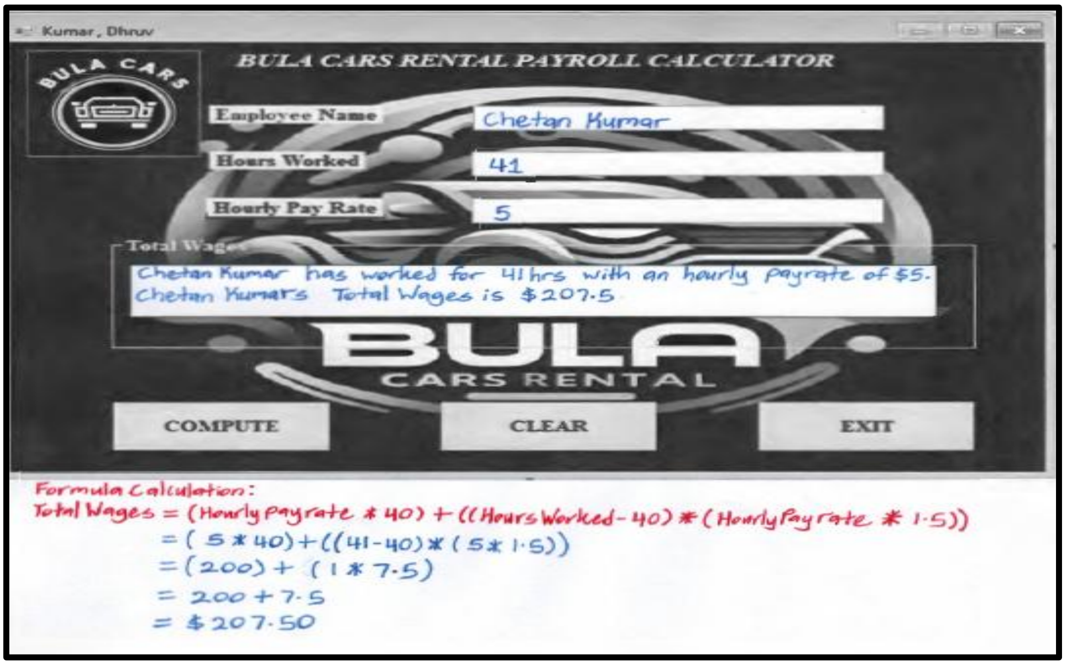
Manual testing with sample data in computer programming is manually executing test cases and using predefined data sets to verify that the software behaves as expected.

Employee Name: Chetan Kumar

Hours Worked: 41hrs

Hourly Pay Rate: $ 5.00

Total Wages: ?



Employee Name: Chetan Kumar

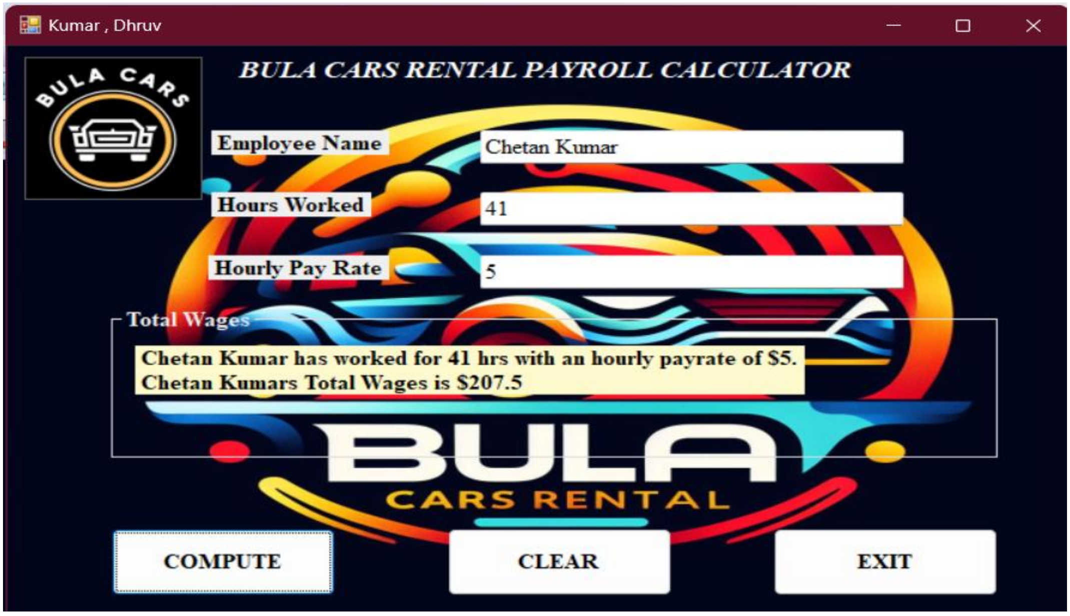
Hours Worked: 41

Hourly Pay Rate: 5

Total Wages: Chetan Kumar has worked for 41hrs with an hourly payrate of $5. Chetan Kumars Total Wages is $207.5

Formula Calculation:

Total Wages = (Hourly Payrate * 40) + ((Hours Worked - 40) * (Hourly Payrate * 1·5))
= (5 * 40)+((41-40) * (5 * 1·5))
= (200) + (1 * 7·5)
= 200 + 7·5
= $ 207·50



Employee Name: Chetan Kumar

Hours Worked: 41

Hourly Pay Rate: 5

Total Wages:
Chetan Kumar has worked for 41 hrs with an hourly payrate of $5.
Chetan Kumars Total Wages is $207.5

STEP 5: PROGRAM DOCUMENTATION

The Bula Cars Rental Company Payroll Calculator documentation provides comprehensive guidance for users, developers, and operators, ensuring successful interaction with the program across various roles. User documentation includes clear instructions for operating the payroll calculator, with step-by-step directions for running the program, entering data (employee name, hours worked, and hourly pay rate), and understanding the output (employee name, hours worked, hourly pay rate, and total weekly pay). Additionally, troubleshooting advice and examples help users navigate common issues and ensure accurate payroll calculations.

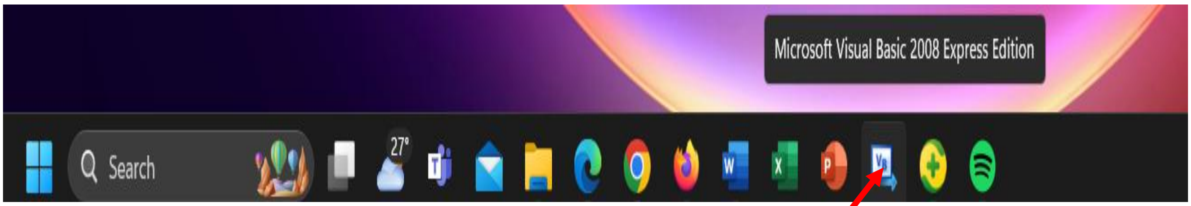
For developers, the documentation offers resources to streamline development tasks, including program examples, descriptions of design tools, and detailed debugging and testing instructions. Developers receive guidance on maintaining and updating the payroll calculator, ensuring its long-term functionality and reliability. Sample code snippets demonstrate key functionalities, aiding in the efficient development and enhancement of the program. This section supports designers, writers, testers, and maintainers by providing a technical overview and procedural descriptions essential for future modifications.

Operator documentation assists machine operators in configuring and managing the payroll calculator hardware. It includes instructions for initial setup, configuration settings, and monitoring hardware performance. Operators can refer to this section to understand hardware behaviour, interpret indicators and alerts, and troubleshoot hardware-related issues. By offering detailed procedural descriptions and hardware-specific guidance, this documentation ensures that operators can effectively manage the hardware, identify potential problems, and maintain optimal performance, thereby supporting the overall functionality of the payroll calculator.

Steps *to prepare* the payroll calculator for Bula Cars Rental Company:

Step 1

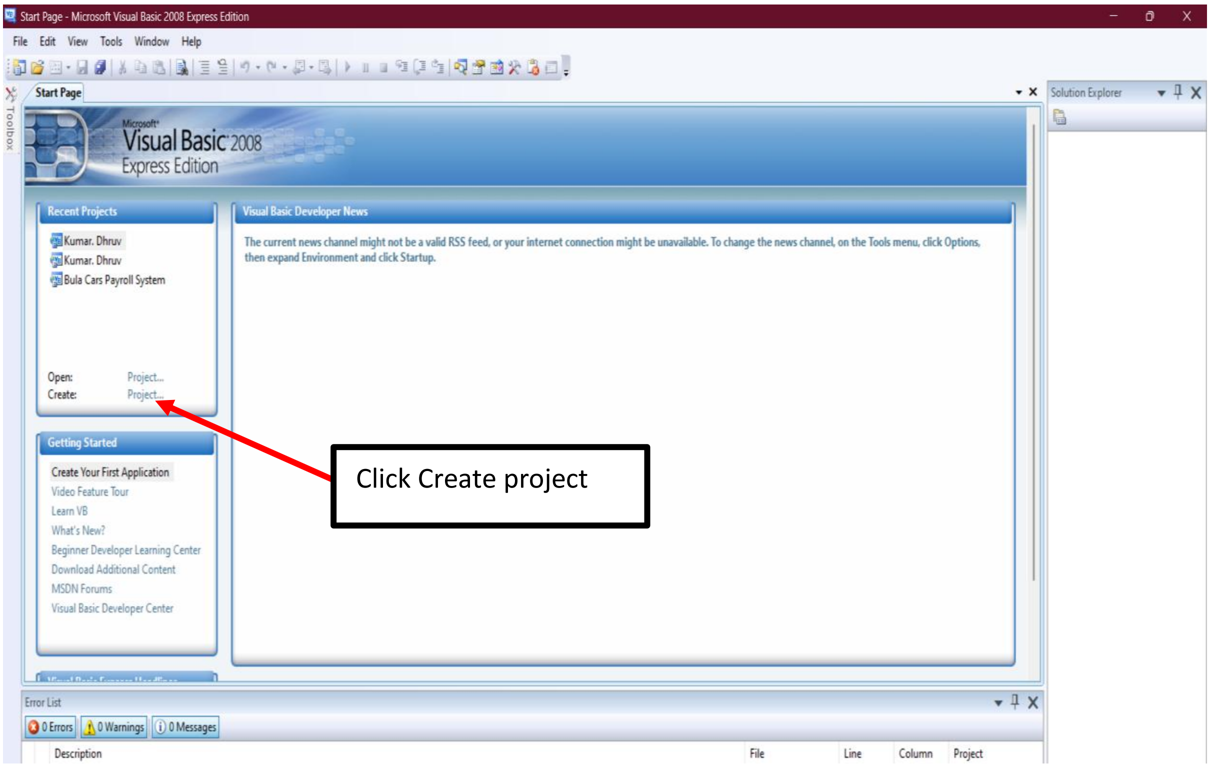
Open VB.net program.



Select VB.net and open

Step 2

A start page will open where the create project is to be selected.

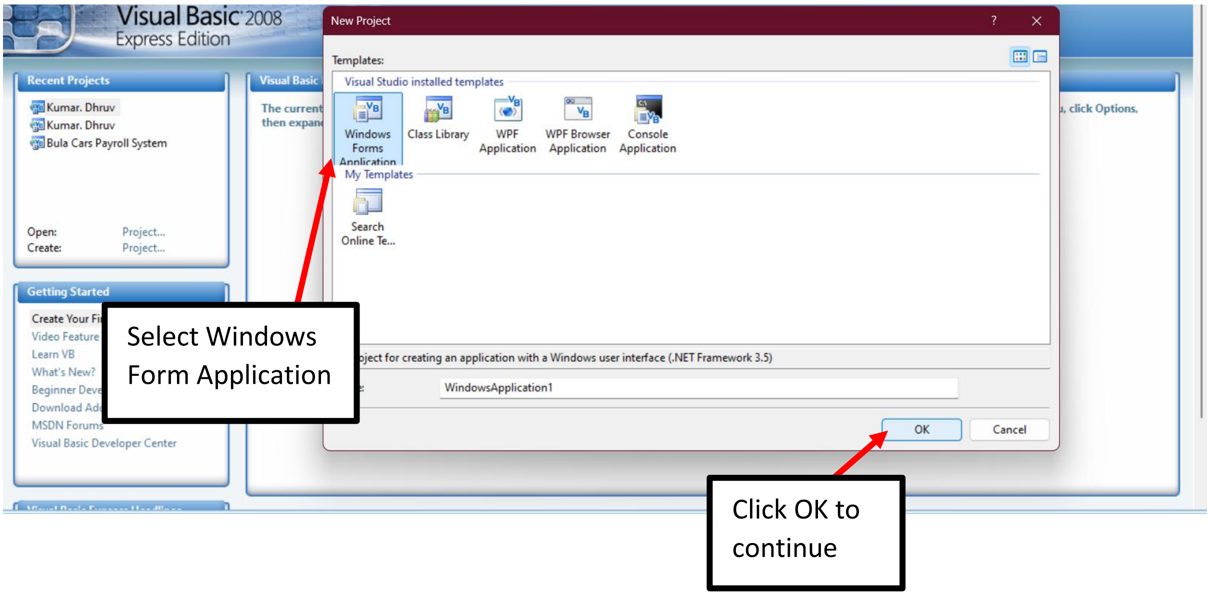

Start Page - Microsoft Visual Basic 2008 Express Edition
File Edit View Tools Window Help
Start Page
Solution Explorer
Toolbox
Microsoft® Visual Basic 2008 Express Edition
Recent Projects
Kumar. Dhruv
Kumar. Dhruv
Bula Cars Payroll System
Open: Project...
Create: Project...
Getting Started
Create Your First Application
Video Feature Tour
Learn VB
What's New?
Beginner Developer Learning Center
Download Additional Content
MSDN Forums
Visual Basic Developer Center
Visual Basic Developer News
Error List
0 Errors
0 Warnings
0 Messages
Description
File
Line
Column
Project

The current news channel might not be a valid RSS feed, or your internet connection might be unavailable. To change the news channel, on the Tools menu, click Options, then expand Environment and click Startup.

Click Create project

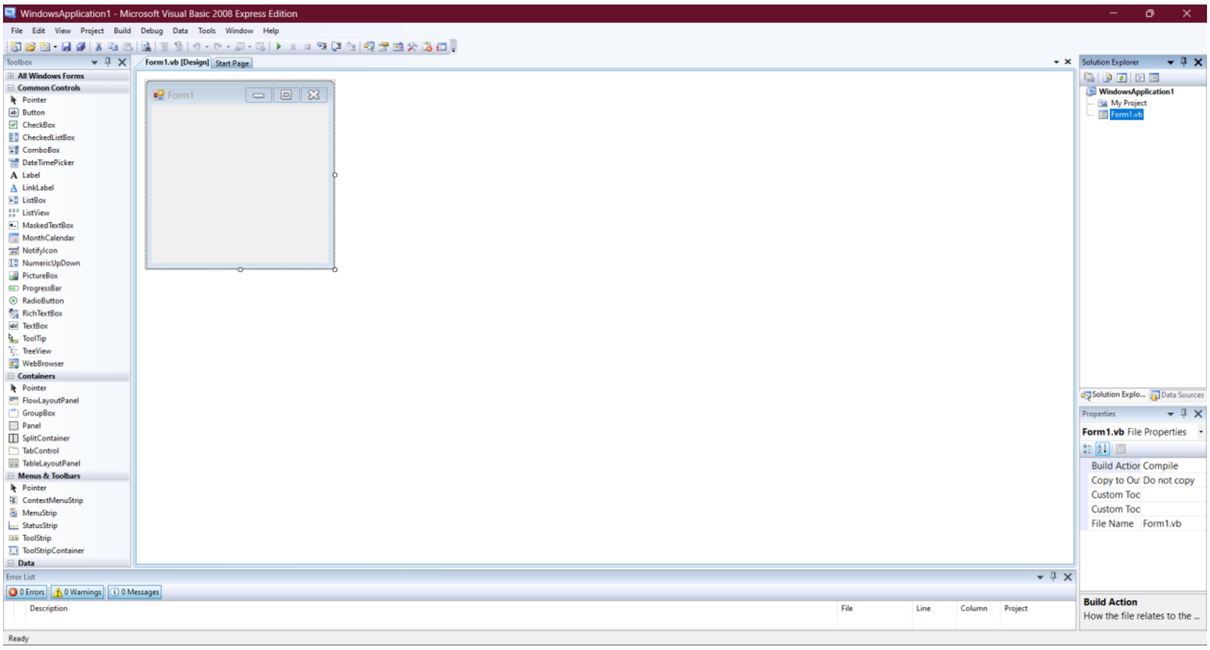
Step 3

A new project page will open where the Windows form application is selected and OK is pressed.



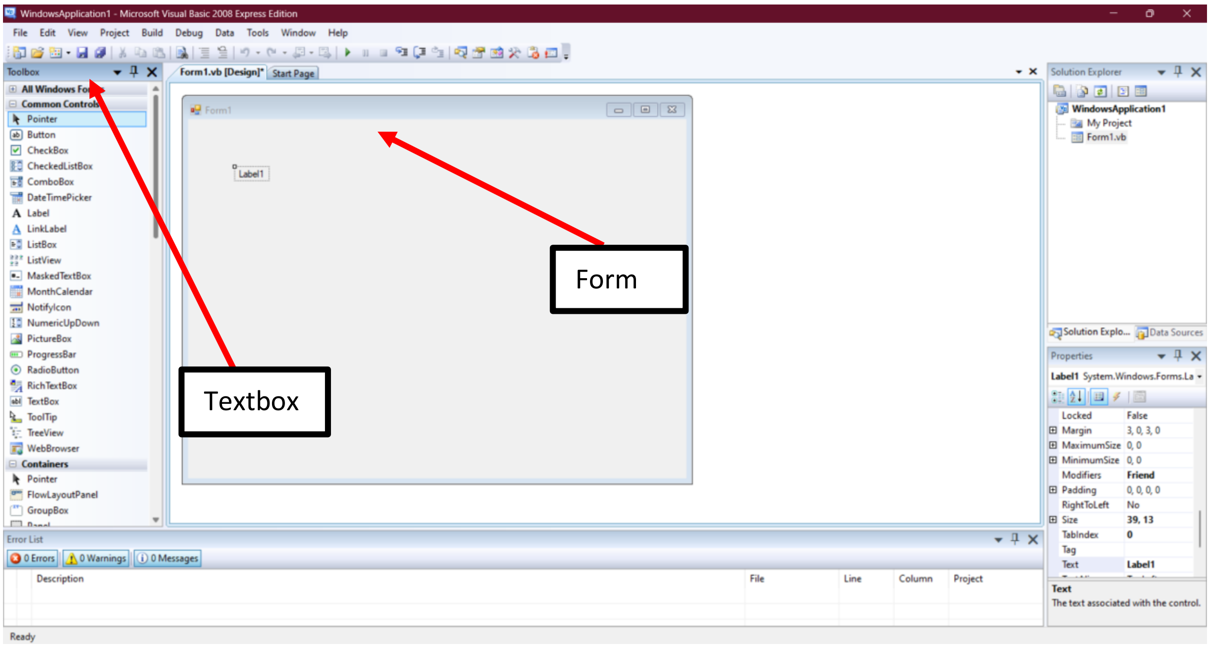
Step 4

A form will open


WindowsApplication1 - Microsoft Visual Basic 2008 Express Edition
File Edit View Project Build Debug Data Tools Window Help
Toolbox
Form1.vb [Design]
Start Page
Form1
Solution Explorer
WindowsApplication1
My Project
Form1.vb
Properties
Error List
Ready

Step 5

Drag and drop controls to the form from the toolbox.



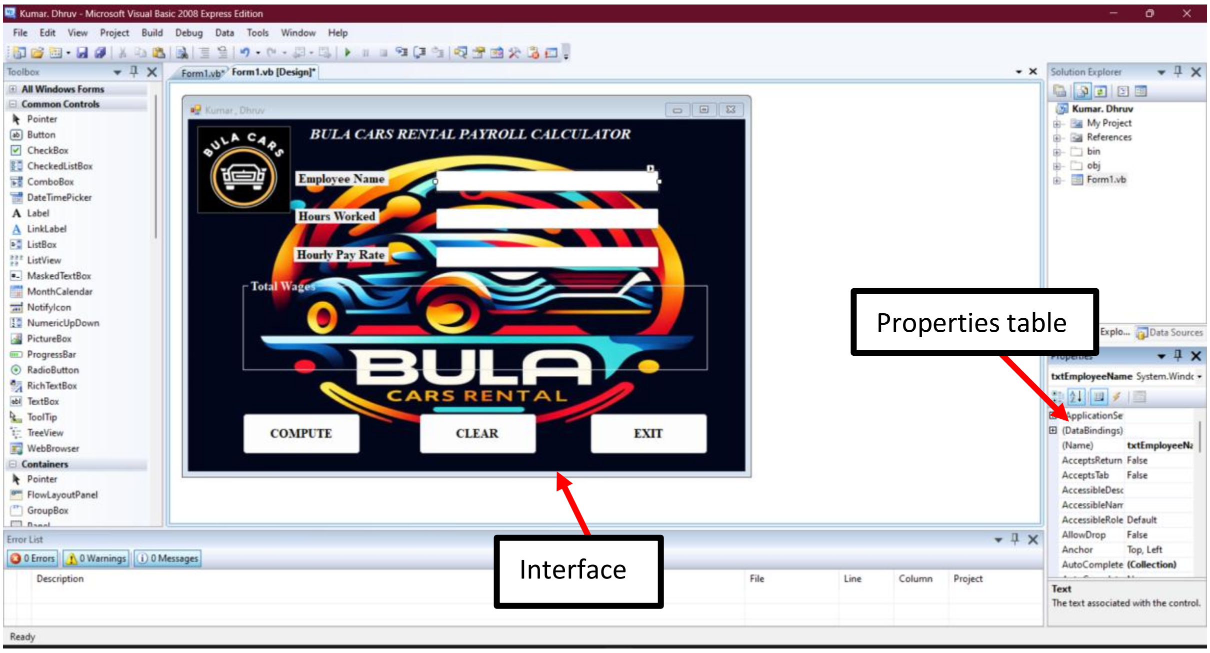
Label1 System.Windows.Forms.La

Locked False
Margin 3, 0, 3, 0
MaximumSize 0, 0
MinimumSize 0, 0
Modifiers Friend
Padding 0, 0, 0, 0
RightToLeft No
Size 39, 13
TabIndex 0
Tag
Text Label1

Text
The text associated with the control.

Step 6

After dragging all the controls from the textbox set your interface and change the name and other properties of the controls and the form in the properties table.



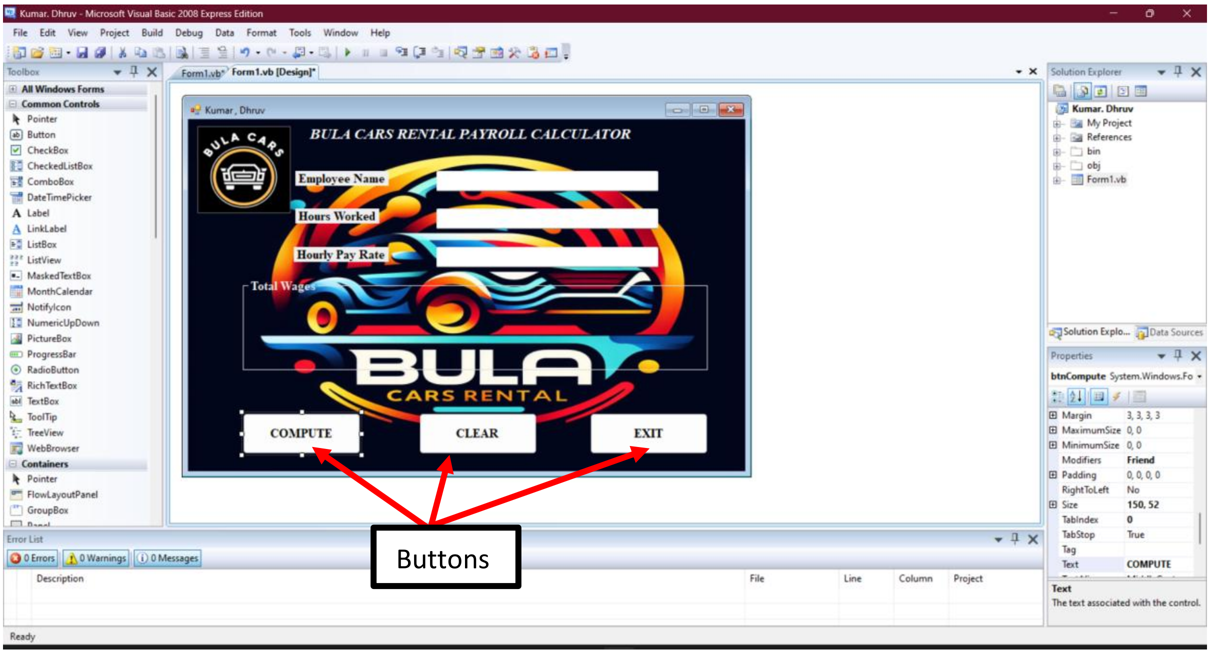
txtEmployeeName System.Windc

ApplicationSe
(DataBindings)
(Name) txtEmployeeNa
AcceptsReturn False
AcceptsTab False
AccessibleDesc
AccessibleNam
AccessibleRole Default
AllowDrop False
Anchor Top, Left
AutoComplete (Collection)

Text
The text associated with the control.

Step 7

Double-tap the buttons to get the coding place.



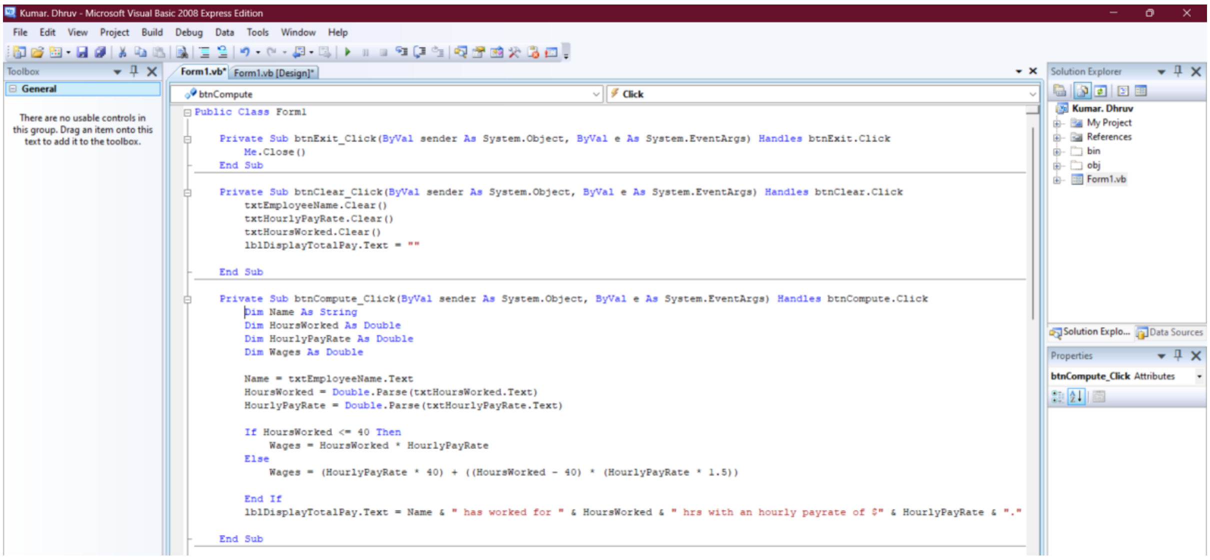
Properties: btnCompute System.Windows.Fo

Margin 3, 3, 3, 3
MaximumSize 0, 0
MinimumSize 0, 0
Modifiers Friend
Padding 0, 0, 0, 0
RightToLeft No
Size 150, 52
TabIndex 0
TabStop True
Tag
Text COMPUTE

Text
The text associated with the control.

Step 8

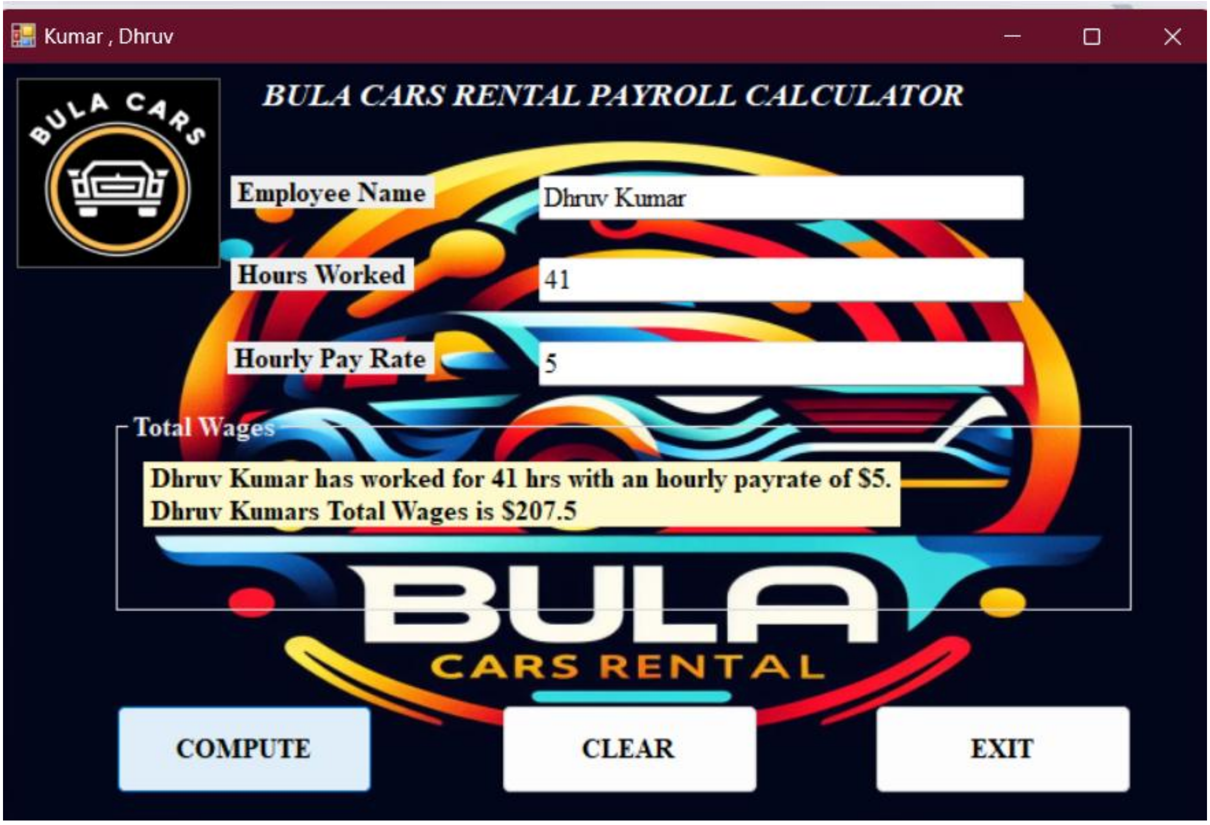
Enter the coding to the controls.


Kumar. Dhruv - Microsoft Visual Basic 2008 Express Edition
File Edit View Project Build Debug Data Tools Window Help
Toolbox
General
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
Form1.vb* Form1.vb [Design]*
btnCompute
Click
Solution Explorer
Kumar. Dhruv
My Project
References
bin
obj
Form1.vb
Properties
btnCompute_Click Attributes

```
Public Class Form1

    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
        Me.Close()
    End Sub

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
        txtEmployeeName.Clear()
        txtHourlyPayRate.Clear()
        txtHoursWorked.Clear()
        lblDisplayTotalPay.Text = ""

    End Sub

    Private Sub btnCompute_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCompute.Click
        Dim Name As String
        Dim HoursWorked As Double
        Dim HourlyPayRate As Double
        Dim Wages As Double

        Name = txtEmployeeName.Text
        HoursWorked = Double.Parse(txtHoursWorked.Text)
        HourlyPayRate = Double.Parse(txtHourlyPayRate.Text)

        If HoursWorked <= 40 Then
            Wages = HoursWorked * HourlyPayRate
        Else
            Wages = (HourlyPayRate * 40) + ((HoursWorked - 40) * (HourlyPayRate * 1.5))

        End If
        lblDisplayTotalPay.Text = Name & " has worked for " & HoursWorked & " hrs with an hourly payrate of $" & HourlyPayRate & "."

    End Sub
```

Step 9

After coding run and test the program.



Kumar , Dhruv

BULA CARS RENTAL PAYROLL CALCULATOR

Employee Name: Dhruv Kumar

Hours Worked: 41

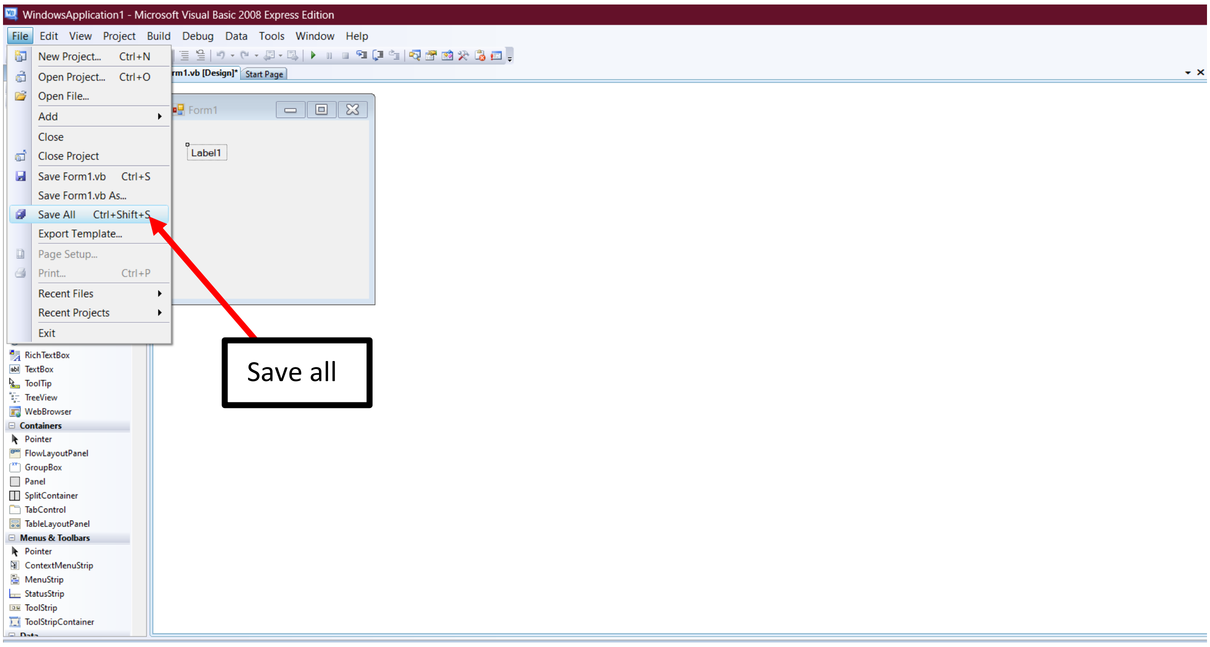
Hourly Pay Rate: 5

Total Wages

Dhruv Kumar has worked for 41 hrs with an hourly payrate of $5.
Dhruv Kumars Total Wages is $207.5

Step 10

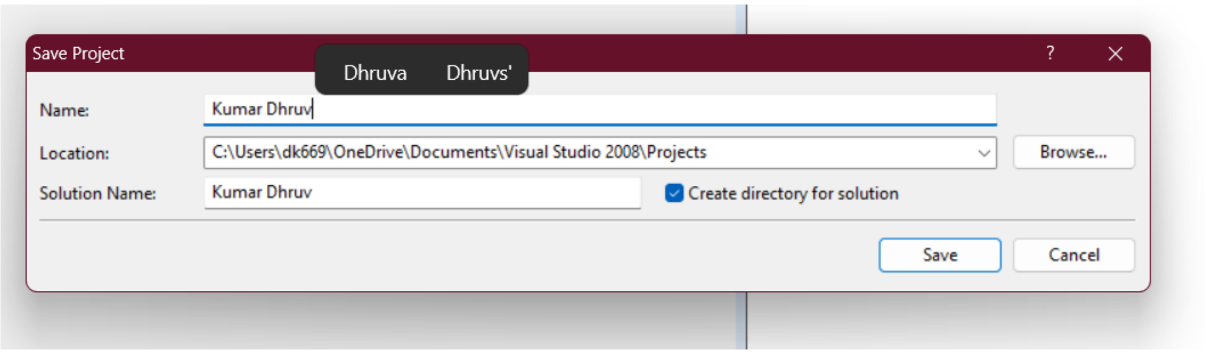
To save the program, go to the file and select Save All.



WindowsApplication1 - Microsoft Visual Basic 2008 Express Edition

Step 11

After clicking save all a save project tab will open where the name of the project can be written and then click save.



Name: Kumar Dhruv

Location: C:\Users\dk669\OneDrive\Documents\Visual Studio 2008\Projects

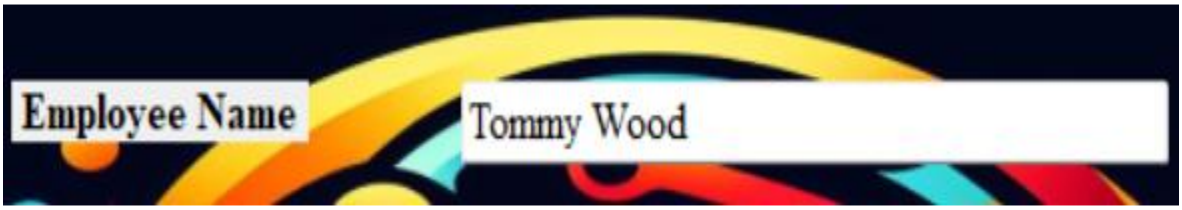
Solution Name: Kumar Dhruv

☑ Create directory for solution

Steps *to be followed while using* the program:

Step 1

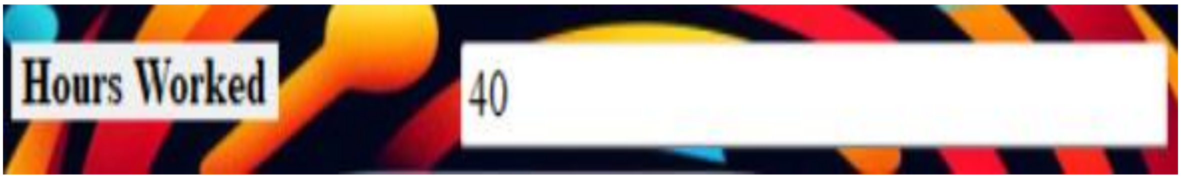
User to enter employee name



Employee Name Tommy Wood

Step 2

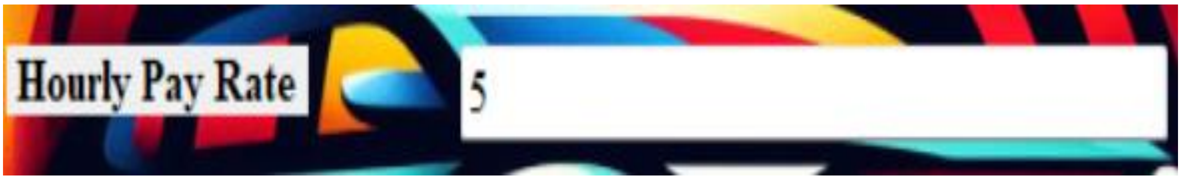
user to enter the hours worked by the employee



Hours Worked 40

Step 3

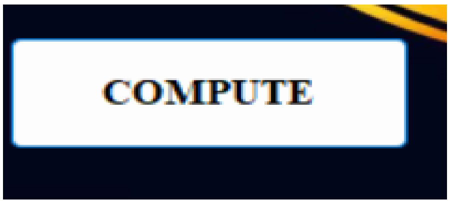
user to enter the hourly pay rate for the employee



Hourly Pay Rate 5

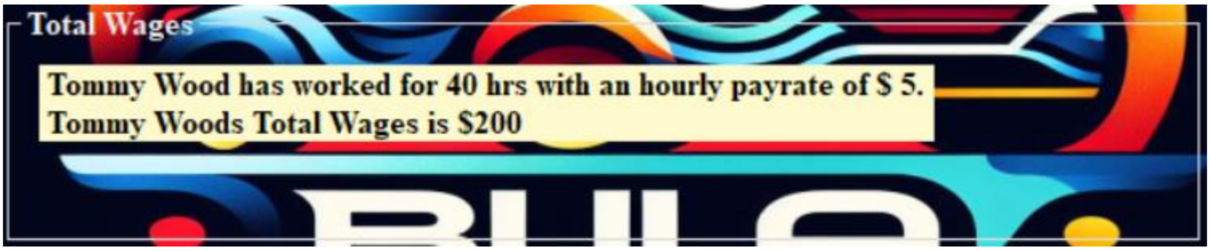
Step 4

The user clicks on the compute button to get the output of total wages



Step 5

After clicking on the compute button, the interface shows the payroll display (summary of the payroll)



Tommy Wood has worked for 40 hrs with an hourly payrate of $ 5.
Tommy Woods Total Wages is $200

However, if the user mistakenly enters incorrect inputs then a clear button is present to erase errors.

Step 6

The user can finally click the exit button to leave the interface

THE BENEFITS OF USING AN AUTOMATED PAYROLL SYSTEM OVER A MANUAL SYSTEM

In the modern business landscape, the adoption of computerised systems for payroll calculation has revolutionised how companies manage their employee compensation. The technological shift at Bula Cars Rental Company offers numerous advantages over traditional manual methods, which were time-consuming and prone to errors. The benefits of computerised payroll systems are accuracy and efficiency, cost-effectiveness, and saving time for the business.

One of the most significant benefits of computerised payroll calculation is the **enhancement of accuracy and efficiency**. Manual payroll processes are inherently susceptible to human error, whether it's in data entry, calculation, or record-keeping. A small mistake can lead to significant discrepancies, affecting employees' pay and potentially causing dissatisfaction and disputes. According to **Cronquist (2024),** a person's pay isn't something you want to mess up, payroll mistakes can cost you and your employees. In contrast, computerised payroll systems minimise these risks by automating calculations and ensuring that data is consistently and accurately processed. Furthermore, computerised systems can process payroll for large numbers of employees in a fraction of the time it would take manually, freeing up valuable human resources to focus on more strategic tasks. The use of automation in payroll also ensures that payments are made on time, enhancing employee satisfaction and trust in the organisation.

Another critical advantage of computerised payroll systems is their **cost-effectiveness**. According to **Bartram (2024),** by reducing the need for manual data entry and minimising errors, businesses can save on labour costs and avoid costly payroll mistakes. While there is an initial investment required for purchasing software and possibly training staff, the long-term savings are substantial. Manual payroll processing requires significant manpower, which translates to higher labour costs. Additionally, the likelihood of errors in manual payroll can lead to costly penalties. Computerised systems reduce these risks by ensuring compliance with tax laws and regulations through automatic updates and accurate calculations. They also decrease the need for extensive administrative staff dedicated to payroll processing, allowing businesses to reallocate resources more efficiently. Furthermore, the maintenance costs of computerised systems are generally lower compared to the ongoing expenses associated with manual processes, such as printing, storage, and handling of physical documents.

One of the most apparent benefits of computerised payroll systems is the **substantial amount of time saved** for the business. Processing payroll manually is an arduous and time-intensive task, involving numerous steps from calculating hours worked and deductions to generating pay cheques and maintaining records. This process can take days or even weeks, especially for larger organisations. Computerised payroll systems streamline these tasks by automating calculations with just a few clicks. This automation drastically reduces the time required to complete payroll, allowing HR personnel and accountants to focus on other important duties. Additionally, the integration of computerised payroll systems can accelerate the process, ensuring seamless data flow. This efficiency not only enhances productivity but also allows businesses to respond more swiftly to payroll-related inquiries and adjustments.

In conclusion, the shift from manual to computerised payroll calculation offers significant advantages in terms of accuracy and efficiency, cost-effectiveness and time-saving. By adopting computerised systems, businesses can ensure that their payroll processes are more reliable, efficient, and compliant, ultimately contributing to better financial management and employee satisfaction. As technology continues to advance, the benefits of computerised payroll systems are likely to become even more pronounced, making them an indispensable tool for modern businesses.

CONCLUSION

The development of Bula Cars' automated payroll system marked a significant technological advancement, streamlining payroll operations and enhancing efficiency. This comprehensive process began with analysing existing procedures to identify inefficiencies and setting clear objectives such as improving accuracy, reducing processing time, and enhancing user experience. Software engineers and developers played pivotal roles: engineers designed a ascendable, secure system architecture and created detailed technical specifications, while developers translate these specifications into functional software, focusing on user-friendly interfaces and robust performance. Their collaborative efforts resulted in a system that drastically reduced payroll processing time, minimized errors, and ensured timely and accurate employee payments. This transformation not only streamlined payroll processing but also improved financial management and employee satisfaction, highlighting the importance of leveraging technology and skilled professionals in modern business operations.

REFERENCES

Bartram, F. (2024). *7 Benefits Of An Automated Payroll System*. Retrieved on 27th June 2024 from:

https://peoplemanagingpeople.com/strategy-operations/payroll-finance/benefits-automated-payroll-system/

Cronquist, D. (2024). *8 Ways Payroll Automation Saves Time and Money*. Retrieved on 25th June 2024 from:

https://www.bamboohr.com/blog/benefits-of-automated-payroll

Kawale, A., Muthmare, H., Devghare, P., Rajbhoj, Y., Sonkusare, R., & Khadse, D. B. (2018). Automatic payroll processing system. *International Journal of Scientific Research in Science, Engineering and Technology*, *4*(4), 567-570. Retrieved on 28th February 2024 from:

https://www.academia.edu/37077251/Automatic_Payroll_Processing_System

Palladan, A. A., & Palladan, N. Y. (2018). Employees views on payroll computerization and its impact on their productivity: A grounded theory approach. *Arabian Journal of Business and Management Review*, *8*(2), 340. Retrieved on 29th February 2024 from:

https://www.researchgate.net/publication/325737417_Employees_Views_on_Payroll_Computerization_and_Its_Impact_on_Their_Productivity_A_Grounded_Theory_Approach

Sheldon, R. (2023). What is pseudocode?: Definition from TechTarget. Retrieved on 7th June 2024 from:

https://www.techtarget.com/whatis/definition/pseudocode

Tobin, J. M. (2024). *The difference between a software developer and a software engineer*. ComputerScience.org. Retrieved on 16th March 2024 from:

https://www.computerscience.org/resources/software-developer-vs-software-engineer/