# Veerendra Chippala

A position with a progressive company that will fully utilize my skills and offers opportunities for continuous professional development.

12-182/1, Thirdline Hanuman Nagar,
Ramavarappadu,
Vijayawada, Andhra Pradesh 521108
**(+91) 9515375868**
**veerendrachippala0207@gmal.com**

## EDUCATION

### **Ramachandra College of Engineering,**Eluru —B.Tech

2020-2023

Completed my bachelors in Electrical and Electronics Engineering in the year 2023 with CGPA-7.1(till date).

### **Govt. Polytechnic,** Gannavaram— *Diploma*

2017-2020

Completed my Diploma in Electrical and Electronics Engineering in the year 2020 with CGPA of 7.6.

### **St. Ann's EM High School,** Prasadampadu— *SSC*

2016-2017

Completed my Matriculation in the year 2017 with CGPA of 9.0.

## PROJECTS

### **Registration and SignUp page**— *Java, HTML, CSS, MySQL*

A dynamic web page which allows the user to register in a webpage and login after registration.

### **Diabetes prediction using Mechine learning through Python**— *Machine learning through python.*

By using the data collected from various test results like BMI, BP, Glucose level etc.., is stored in dataset is given to the program as input to predict whether the person having diabetes or not by using SVM,KNN and Random Forest algorithm.

### **Basic Online shopping software**—Python

A basic online shopping software using only if and else blocks, which works according to user choices like amazon and flipkart.

## SKILLS

**Front end:** HTML5 ,CSS3

**BackEnd**: Java and Python.

**Database**: Mysql.

## CERTIFICATIONS

**Internship at Kodnest:**
Received a certificate for successfully completing the java fullStack internship and also enriching my skills and expertised in wed app development.

**Co-curricular Activites:**

Won second prize in Circuit debugging held at Ramachandra College of Engineering, Eluru

Received participation certificate in paper presentation at Sasi Institute of Technology, Tadepalligudem.

## LANGUAGES

English

Telugu

# BITWISE OPERATOR

Bitwise operators are used to perform operations on individual bits of binary numbers. These operators manipulate the bits at the binary level, providing a way to perform low-level operations on data. Bitwise operators are commonly used in programming languages, particularly in systems programming and embedded systems.

There are several bitwise operators available in most programming languages, including:

1. **AND (&):** The bitwise AND operator compares the corresponding bits of two numbers and produces a new number where each bit is set to 1 if both bits are 1, otherwise, it is set to 0. The truth table for the AND operator is as follows:

   0 & 0 = 0

   0 & 1 = 0

   1 & 0 = 0

   1 & 1 = 1

   **Example:**

```
class LogAnd{
        public static void main(String[] args){
                int a = 5;    // Binary: 0101
                int b = 3;    // Binary: 0011
                int result = a & b;
                // result = 0101 & 0011 = 0001 (Decimal: 1)
                System.out.println(result);
        }
}
```

2. **OR (|):** The bitwise OR operator compares the corresponding bits of two numbers and produces a new number where each bit is set to 1 if either of the bits is 1, otherwise, it is set to 0. The truth table for the OR operator is as follows:

   0 | 0 = 0

   0 | 1 = 1

   1 | 0 = 1

   1 | 1 = 1

   **Example:**

```
Class LogOr{
        Public static void main(String[] args){
                int a = 5;    // Binary: 0101
                int b = 3;    // Binary: 0011
                int result = a | b;
```

```
        // result = 0101 | 0011 = 0111 (Decimal: 7)
        System.out,println(result);
    }
}
```

3. XOR (^): The bitwise XOR (exclusive OR) operator compares the corresponding bits of two numbers and produces a new number where each bit is set to 1 if the bits are different, otherwise, it is set to 0. The truth table for the XOR operator is as follows:

   0 ^ 0 = 0

   0 ^ 1 = 1

   1 ^ 0 = 1

   1 ^ 1 = 0

   **Example:**

```
Class LogXor{
    Public static void main(String[] args){
        int a = 5;    // Binary: 0101
        int b = 3;    // Binary: 0011
        int result = a ^ b;
        // result = 0101 ^ 0011 = 0110 (Decimal: 6)
        System.out,println(result);
    }
}
```

4. NOT (~): The bitwise NOT operator flips each bit of a number, converting 0s to 1s and 1s to 0s. It is a unary operator, meaning it operates on a single operand. The truth table for the NOT operator is as follows:

   ~0 = 1

   ~1 = 0

   **Example:**

```
Class LogNot{
    Public static void main(String[] args){
        int a = 5;    // Binary: 0101
        int result = ~a;
        // result = ~0101 = 1010 (Decimal: -6)
        System.out,println(result);
    }
}
```

**Note**: That the result of the bitwise NOT operation depends on the number of bits used to represent the number (usually the size of the data type). In the example above, assuming we are using a 4-bit integer, the result is **-6** because the leftmost bit represents the sign (1 for negative numbers).

These bitwise operators can be useful in various scenarios, such as manipulating individual bits in binary data, creating bit masks, or performing optimizations in certain algorithms.