# Vector Analyzer Project Report

## 1. Introduction

The Vector Analyzer is a command-line application designed to assist in performing vector-related computations. The project is developed in C# and leverages object-oriented programming principles. It offers a robust set of functionalities to work with 3D vectors, including addition, subtraction, scalar multiplication, dot product, cross product, and angle calculations.

## 2. Features and Functionalities

### 2.1 Vector Operations:

- **Addition and Subtraction:** Perform arithmetic operations on vectors.
- **Scalar Multiplication:** Multiply vectors by scalar values.
- **Dot Product and Cross Product:** Calculate the dot product and cross product between two vectors.
- **Projection:** Determine the projection of one vector over another.

### 2.2 Vector Store:

- **Add and Remove Vectors:** Store up to 20 vectors with unique names.
- **Modify Vectors:** Update the properties of existing vectors.
- **Explore Vectors:** View details, including magnitude and components.
- **Unit Vectors:** Calculate and display the unit vector of a given vector.

### 2.3 Angle Calculation:

- Calculate the angle between two vectors in degrees.

## 3. System Design

### 3.1 Class Structure:

- **Vector Class:** Encapsulates vector data and operations such as parsing, magnitude calculation, and unit vector computation.
- **Store Class:** Manages the collection of vectors, offering functionalities to add, remove, modify, and explore vectors.
- **VectorOperations Class:** Implements static methods for vector arithmetic and advanced computations like dot and cross products.
- **VectorAnalyzer Class:** Serves as the main entry point, handling user interaction and integrating all functionalities.

### 3.2 Key Methods:

- **Parse:** Converts string input into a Vector object.
- **GetMagnitude:** Calculates the vector's magnitude.
- **AddVectors/SubtractVectors:** Perform vector arithmetic.

- **DotProduct/CrossProduct:** Compute vector algebra operations.
- **Projection:** Determines the projection of one vector onto another.

# 4. Implementation Highlights

#### 4.1 User Interaction:

The application employs a menu-driven approach to guide users through available options. Separate menus are implemented for the main application and the Vector Store.

#### 4.2 Robust Parsing Mechanism:

The Vector class's static Parse method efficiently converts user input into a vector object, handling common input errors gracefully.

#### 4.3 Comprehensive Error Handling:

From invalid input to operations on zero vectors, the application ensures robust error detection and clear feedback to users.

# 5. Example Workflow

1. **Start the Application:** The main menu offers options such as exploring the Vector Store, performing vector arithmetic, or exiting the application.
2. **Add a Vector:** Users input vectors in the format "v = 2i + 3j + 4k." The application parses the input and adds it to the store.
3. **Perform Operations:** Select options for addition, subtraction, or scalar multiplication. Input vector names as prompted.
4. **Explore Vectors:** View vector details or compute the unit vector and angles between vectors.
5. **Exit the Application:** Close the program after completing tasks.

# 6. Future Enhancements

- Extend to n-dimensional vectors.
- Implement a graphical user interface (GUI) for visualization.
- Support for saving/loading vectors from files.

# 7. Conclusion

The Vector Analyzer is a comprehensive tool for 3D vector manipulation, suitable for students and professionals working on vector-related problems. With its intuitive interface and robust backend, the application demonstrates the effectiveness of C# in implementing mathematical tools.