HomePageController.apxc

```
public class HomePageController {
    public String selectedConversionType { get; set; }

    public List<SelectOption> getItems() {
        List<SelectOption> options = new List<SelectOption>();
        options.add(new SelectOption('Speed Conversions', 'Speed Conversions'));
        options.add(new SelectOption('Weight Conversions', 'Weight Conversions'));
        options.add(new SelectOption('Data Conversions', 'Data Conversions'));
        options.add(new SelectOption('Matrix', 'Matrix'));
        return options;
    }

    public PageReference redirectToConversionPage() {
        if(selectedConversionType == 'Speed Conversions') {
            return Page.scon; // Assuming 'scon' is a Visualforce page for temperature conversion
        } else if(selectedConversionType == 'Weight Conversions') {
            return Page.wcon; // Assuming 'wcon' is a Visualforce page for weight conversion
        } else if(selectedConversionType == 'Matrix') {
            return Page.matrixHome; // Assuming 'lengthConversion' is a Visualforce page for length
conversion
        } else if(selectedConversionType == 'Data Conversions') {
            return Page.dcon; // Assuming 'lengthConversion' is a Visualforce page for length conversion
        }
        return null;
    }
}
```

HomePageController.vfp

```
<apex:page controller="HomePageController">
    <style>
```

```
body {
    font-family: Arial, sans-serif;
    background-color: grey;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
.container {
    width: 600px;
    padding: 20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.header {
    text-align: center;
    margin-bottom: 20px;
    color: purple;
      font-size: 20px;
}
.select-section {
    text-align: center;
    margin-bottom: 20px;
}
.btn-go {
    background-color: #007bff;
    color: #fff;
    border: none;
```

```
            padding: 10px 20px;
            border-radius: 5px;
            cursor: pointer;
            transition: background-color 0.3s ease;
            font-size: 16px;
            width: 100%;
        }
        .btn-go:hover {
            background-color: #0056b3;
        }
    </style>

    <apex:form >
        <div class="container">
            <div class="header">
                <h1 style="color: black; font-weight: bold; font-size: 30px">Calculator</h1> <!-- Change header color -->
                <p>Select Operation Type</p>
            </div>
            <div class="select-section">
                <apex:selectList value="{!selectedConversionType}" label="Conversion Type">
                    <apex:selectOptions value="{!items}"/>
                </apex:selectList>
            </div>
            <div style="text-align: center;">
                <apex:commandButton value="Proceed" styleClass="btn-go" action="{!redirectToConversionPage}"/>
            </div>
        </div>
    </apex:form>
</apex:page>
```

wcon.apxc

```
public class wcon{
    public String fromUnit { get; set; }
    public Decimal fromValue { get; set; }
    public String toUnit { get; set; }
    public Decimal convertedValue { get; set; }

    public List<SelectOption> getUnitOptions() {
        List<SelectOption> options = new List<SelectOption>();
        options.add(new SelectOption('Milligrams', 'Milligrams'));
        options.add(new SelectOption('Grams', 'Grams'));
        options.add(new SelectOption('Kilograms', 'Kilograms'));
        options.add(new SelectOption('Tonnes', 'Tonnes'));
        return options;
    }

    public void convert() {
        if (fromUnit == toUnit) {
            convertedValue = fromValue;
            return;
        }

        Decimal result = 0;
        if (fromUnit == 'Milligrams') {
            result = fromValue / 1000;
        } else if (fromUnit == 'Grams') {
            result = fromValue;
        } else if (fromUnit == 'Kilograms') {
            result = fromValue * 1000;
        } else if (fromUnit == 'Tonnes') {
            result = fromValue * 1000000;
```

```
        }

        if (toUnit == 'Milligrams') {
            convertedValue = result * 1000;
        } else if (toUnit == 'Grams') {
            convertedValue = result;
        } else if (toUnit == 'Kilograms') {
            convertedValue = result / 1000;
        } else if (toUnit == 'Tonnes') {
            convertedValue = result / 1000000;
        }
    }
}
```

wcon.vfp

```
<apex:page controller="wcon">

    <apex:form >
        <apex:pageBlock title="Weight Conversion">
            <apex:pageBlockSection columns="1">
                <div class="inputContainer">
                    <apex:selectList value="{!fromUnit}" size="1">
                        <apex:selectOptions value="{!unitOptions}"/>
                    </apex:selectList>

                    <apex:inputText value="{!fromValue}" label="Value"/>

                    <apex:selectList value="{!toUnit}" size="1">
                        <apex:selectOptions value="{!unitOptions}"/>
                    </apex:selectList>
                </div>
            </apex:pageBlockSection>
```

```
        <apex:pageBlockSection >
            <apex:commandButton value="Convert" action="{!convert}"/>
         </apex:pageBlockSection>

        <apex:pageBlockSection title="Converted Value" collapsible="false">
            <apex:outputPanel id="outputPanel">
                <apex:outputText value="{0, number, #,##0.00}">
                    <apex:param value="{!convertedValue}"/>
                </apex:outputText>
            </apex:outputPanel>
        </apex:pageBlockSection>
    </apex:pageBlock>
  </apex:form>
</apex:page>
```

scon.apxc

```
public class scon {
    public String inputValue { get; set; }
    public String selectedConversion { get; set; }
    public String result { get; set; }

    public void convertSpeed() {
        Decimal inputSpeed = Decimal.valueOf(inputValue);
        if (selectedConversion == 'kmphToMph') {
            result = String.valueOf(kmphToMph(inputSpeed)) + ' mph';
        } else if (selectedConversion == 'mphToKmph') {
            result = String.valueOf(mphToKmph(inputSpeed)) + ' km/h';
        } else if (selectedConversion == 'kmToM') {
            result = String.valueOf(kmToM(inputSpeed)) + ' m/s';
        } else if (selectedConversion == 'mToKm') {
```

```
            result = String.valueOf(mToKm(inputSpeed)) + ' km/s';
        }
    }

    private Decimal kmphToMph(Decimal kmph) {
        return kmph * 0.62137119;
    }

    private Decimal mphToKmph(Decimal mph) {
        return mph / 0.62137119;
    }

    private Decimal kmToM(Decimal km) {
        return km * 1000;
    }

    private Decimal mToKm(Decimal m) {
        return m / 1000;
    }
}
```

scon.vfp

```
<apex:page controller="scon">
    <apex:form >
        <apex:pageBlock title="Speed Converter">
            <apex:pageBlockSection >
                <apex:inputText value="{!inputValue}" label="Speed"/>
                <apex:selectList value="{!selectedConversion}" size="1" label="Conversion Type">
                    <apex:selectOption itemValue="kmphToMph" itemLabel="km/h to mph"/>
                    <apex:selectOption itemValue="mphToKmph" itemLabel="mph to km/h"/>
                    <apex:selectOption itemValue="kmToM" itemLabel="km/s to m/s"/>
                    <apex:selectOption itemValue="mToKm" itemLabel="m/s to km/s"/>
```

```
            </apex:selectList>
            <apex:commandButton action="{!convertSpeed}" value="Convert"
rerender="resultPanel"/>
        </apex:pageBlockSection>
        <apex:pageBlockSection title="Result" id="resultPanel">
            <apex:outputText value="{!result}"/>
        </apex:pageBlockSection>
    </apex:pageBlock>
  </apex:form>
</apex:page>
```

MatrixCalcController.apxc

```
public class MatrixCalcController {
  public List<List<Integer>> matrixA { get; set; }
  public List<List<Integer>> matrixB { get; set; }
  public List<List<Integer>> resultMatrix { get; set; }
  public MatrixDeterminantController determinantController { get; set; }

  public MatrixCalcController() {
    // Initialize matrices with default values
    matrixA = new List<List<Integer>>{
      new List<Integer>{0, 0},
      new List<Integer>{0, 0}
    };
    matrixB = new List<List<Integer>>{
      new List<Integer>{0, 0},
      new List<Integer>{0, 0}
    };
      determinantController = new MatrixDeterminantController();
  }
```

```
    public void addMatrices() {
        resultMatrix = addOrSubtractMatrices(matrixA, matrixB, true);
    }

    public void subtractMatrices() {
        resultMatrix = addOrSubtractMatrices(matrixA, matrixB, false);
    }

    public void multiplyMatrices() {
        if (matrixA[0].size() != matrixB.size()) {
            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Number of columns in Matrix A must be equal to the number of rows in Matrix B.'));
            return;
        }

        resultMatrix = new List<List<Integer>>();
        for (Integer i = 0; i < matrixA.size(); i++) {
            resultMatrix.add(new List<Integer>());
            for (Integer j = 0; j < matrixB[0].size(); j++) {
                Integer sum = 0;
                for (Integer k = 0; k < matrixB.size(); k++) {
                    sum += matrixA[i][k] * matrixB[k][j];
                }
                resultMatrix[i].add(sum);
            }
        }
    }

    public void transposeMatrixA() {
        resultMatrix = transpose(matrixA);
    }
```

```
    public void transposeMatrixB() {
        resultMatrix = transpose(matrixB);
    }

    public void calculateDeterminant() {
        determinantController.matrix = matrixA; // Use matrixA or matrixB as per your requirement
        determinantController.calculate();
    }

    public void inverseMatrixA() {
        resultMatrix = inverse(matrixA);
    }

    public void inverseMatrixB() {
        resultMatrix = inverse(matrixB);
    }

    private List<List<Integer>> addOrSubtractMatrices(List<List<Integer>> a, List<List<Integer>> b,
Boolean isAddition) {
        List<List<Integer>> result = new List<List<Integer>>();
        for (Integer i = 0; i < a.size(); i++) {
            result.add(new List<Integer>());
            for (Integer j = 0; j < a[0].size(); j++) {
                Integer val = isAddition ? a[i][j] + b[i][j] : a[i][j] - b[i][j];
                result[i].add(val);
            }
        }
        return result;
    }
```

```
    private List<List<Integer>> transpose(List<List<Integer>> matrix) {
        List<List<Integer>> transposedMatrix = new List<List<Integer>>();
        for (Integer j = 0; j < matrix[0].size(); j++) {
            transposedMatrix.add(new List<Integer>());
            for (Integer i = 0; i < matrix.size(); i++) {
                transposedMatrix[j].add(matrix[i][j]);
            }
        }
        return transposedMatrix;
    }

    private List<List<Integer>> inverse(List<List<Integer>> matrix) {
        // Check if the matrix is 2x2
        if (matrix.size() != 2 || matrix[0].size() != 2) {
            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Inverse can only be calculated for 2x2 matrices.'));
            return null;
        }

        Integer determinant = matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0];

        // Check if determinant is zero
        if (determinant == 0) {
            ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Matrix is singular, inverse does not exist.'));
            return null;
        }

        Integer factor = 1 / determinant;

        List<List<Integer>> result = new List<List<Integer>>();
```

```
        result.add(new List<Integer>{
            matrix[1][1] * factor,
            -matrix[0][1] * factor
        });

        result.add(new List<Integer>{
            -matrix[1][0] * factor,
            matrix[0][0] * factor
        });

        return result;
    }
}
```

MatrixCalcController.vfp

```
<apex:page controller="MatrixCalcController">
    <apex:form >
        <apex:pageMessages />

        <apex:pageBlock title="Matrix Calculator">
            <apex:pageBlockSection title="Matrix A" columns="2">
                <apex:inputText value="{!matrixA[0][0]}" style="width: 50px;"/>
                <apex:inputText value="{!matrixA[0][1]}" style="width: 50px;"/>

                <apex:inputText value="{!matrixA[1][0]}" style="width: 50px;"/>
                <apex:inputText value="{!matrixA[1][1]}" style="width: 50px;"/>
            </apex:pageBlockSection>

            <apex:pageBlockSection title="Matrix B" columns="2">
                <apex:inputText value="{!matrixB[0][0]}" style="width: 50px;"/>
                <apex:inputText value="{!matrixB[0][1]}" style="width: 50px;"/>
```

```
            <apex:inputText value="{!matrixB[1][0]}" style="width: 50px;"/>
            <apex:inputText value="{!matrixB[1][1]}" style="width: 50px;"/>
        </apex:pageBlockSection>

        <apex:pageBlockButtons location="bottom">
            <apex:commandButton value="Add" action="{!addMatrices}" style="margin-right: 5px;"/>
            <apex:commandButton value="Subtract" action="{!subtractMatrices}" style="margin-right: 5px;"/>
            <apex:commandButton value="Multiply" action="{!multiplyMatrices}" style="margin-right: 5px;"/>

            <apex:commandButton value="Transpose Matrix A" action="{!transposeMatrixA}" style="margin-right: 5px;"/>
            <apex:commandButton value="Transpose Matrix B" action="{!transposeMatrixB}" style="margin-right: 5px;"/>

            <apex:commandButton value="Inverse Matrix A" action="{!inverseMatrixA}" style="margin-right: 5px;"/>
            <apex:commandButton value="Inverse Matrix B" action="{!inverseMatrixB}" style="margin-right: 5px;"/>

            <apex:commandButton value="Calculate Determinant" action="{!calculateDeterminant}" style="margin-right: 5px;"/>
        </apex:pageBlockButtons>

        <apex:pageBlockSection title="Result" rendered="{!resultMatrix != null}">
            <apex:repeat value="{!resultMatrix}" var="row">
                <apex:repeat value="{!row}" var="cell">
                    <apex:outputText value="{!cell}" style="width: 50px;"/>
                </apex:repeat>
                <br/>
            </apex:repeat>
        </apex:pageBlockSection>
```

```
        <apex:pageBlockSection title="Determinant"
rendered="{!determinantController.showResult}">
            <apex:outputText value="Determinant: {!determinantController.determinant}"/>
        </apex:pageBlockSection>
    </apex:pageBlock>
  </apex:form>
</apex:page>
```

DataConversionController.apxc

```
public class DataConversionController {
  public List<SelectOption> unitOptions { get; set; }
  public String selectedFromUnit { get; set; }
  public String selectedToUnit { get; set; }
  public Double inputValue { get; set; }
  public Double convertedValue { get; set; }

  public DataConversionController() {
    unitOptions = new List<SelectOption>();
    unitOptions.add(new SelectOption('bit', 'Bit'));
    unitOptions.add(new SelectOption('byte', 'Byte'));
    unitOptions.add(new SelectOption('kb', 'Kilobyte'));
    unitOptions.add(new SelectOption('mb', 'Megabyte'));
    unitOptions.add(new SelectOption('gb', 'Gigabyte'));
    unitOptions.add(new SelectOption('tb', 'Terabyte'));
  }

  public void convertData() {
    if (selectedFromUnit == null || selectedToUnit == null || inputValue == null) {
        ApexPages.addMessage(new ApexPages.Message(ApexPages.Severity.ERROR, 'Please fill out
all fields.'));
        return;
```

```
    }

    if (selectedFromUnit.equalsIgnoreCase(selectedToUnit)) {
        convertedValue = inputValue;
        return;
    }

    Double result;
    if (selectedFromUnit.equals('bit')) {
        result = convertFromBit(inputValue, selectedToUnit);
    } else if (selectedToUnit.equals('bit')) {
        result = convertToBit(inputValue, selectedFromUnit);
    } else {
        // Convert to bytes first
        Double bytes = convertToBytes(inputValue, selectedFromUnit);
        // Then convert from bytes to the target unit
        result = convertFromBytes(bytes, selectedToUnit);
    }

    convertedValue = result;
}

private Double convertToBytes(Double value, String unit) {
    if (unit.equals('byte')) {
        return value;
    } else if (unit.equals('kb')) {
        return value * 1024;
    } else if (unit.equals('mb')) {
        return value * 1024 * 1024;
    } else if (unit.equals('gb')) {
        return value * 1024 * 1024 * 1024;
```

```
    } else if (unit.equals('tb')) {
        return value * 1024 * 1024 * 1024 * 1024;
    }
    return 0.0;
}

private Double convertFromBytes(Double bytes, String unit) {
    if (unit.equals('byte')) {
        return bytes;
    } else if (unit.equals('kb')) {
        return bytes / 1024;
    } else if (unit.equals('mb')) {
        return bytes / (1024 * 1024);
    } else if (unit.equals('gb')) {
        return bytes / (1024 * 1024 * 1024);
    } else if (unit.equals('tb')) {
        return bytes / (1024 * 1024 * 1024 * 1024);
    }
    return 0.0;
}

private Double convertFromBit(Double value, String unit) {
    // Convert from bit to bytes first, then to the target unit
    Double bytes = value / 8;
    return convertFromBytes(bytes, unit);
}

private Double convertToBit(Double value, String unit) {
    // Convert to bytes first, then to bit
    Double bytes = convertToBytes(value, unit);
    return bytes * 8;
```

```
    }
}
```

DataConversionController.vfp

```
<apex:page controller="DataConversionController" docType="html-5.0">
    <style>
        body {
            Arial, sans-serif;
            background-color: #f2f2f2;
            0;
            padding: 0;
        }
        .container {
            width: 50%;
            50px auto;
            background-color: #fff;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        .form-group {
            20px;
        }
        .form-group label {
            font-weight: bold;
            display: block;
            5px;
            color: #333;
        }
        .form-group select {
            width: 100%;
```

```
        padding: 10px;
        border: 1px solid #ccc;
        border-radius: 5px;
        background-color: #fff;
        color: #333;
    }
    .btn {
        background-color: #4CAF50;
        color: #fff;
        border: none;
        border-radius: 5px;
        padding: 10px 20px;
        cursor: pointer;
        font-size: 16px;
    }
    .btn:hover {
        background-color: #45a049;
    }
    .result {
        20px;
        padding: 10px;
        border: 1px solid #ccc;
        border-radius: 5px;
        background-color: #f9f9f9;
    }
</style>

<div class="container">
    <h2>Data Conversion</h2>
    <apex:form >
        <div class="form-group">
```

```
            <label for="inputValue">Input Value:</label>
            <apex:input type="number" id="inputValue" value="{!inputValue}" required="true"/>
        </div>
        <div class="form-group">
            <label for="selectedFromUnit">From:</label>
            <apex:selectList id="selectedFromUnit" value="{!selectedFromUnit}" required="true">
                <apex:selectOptions value="{!unitOptions}"/>
            </apex:selectList>
        </div>
        <div class="form-group">
            <label for="selectedToUnit">To:</label>
            <apex:selectList id="selectedToUnit" value="{!selectedToUnit}" required="true">
                <apex:selectOptions value="{!unitOptions}"/>
            </apex:selectList>
        </div>
        <div class="form-group">
            <apex:commandButton value="Convert" action="{!convertData}" styleClass="btn"/>
        </div>
        <div class="result">
            <apex:outputPanel rendered="{!convertedValue != null}">
                <p>Value: {!convertedValue}</p>
            </apex:outputPanel>
        </div>
    </apex:form>
  </div>
</apex:page>
```

TrigonometryCalculatorController.apxc

```
public class TrigonometryCalculatorController {
    public Double inputValue { get; set; }
    public Double sinResult { get; set; }
```

```
    public Double cosResult { get; set; }
    public Double tanResult { get; set; }
    public Boolean showSinResult { get; set; }
    public Boolean showCosResult { get; set; }
    public Boolean showTanResult { get; set; }

    public TrigonometryCalculatorController() {
        inputValue = 0.0;
        sinResult = 0.0;
        cosResult = 0.0;
        tanResult = 0.0;
        showSinResult = false;
        showCosResult = false;
        showTanResult = false;
    }

    public void computeSin() {
        sinResult = Math.sin(inputValue);
        showSinResult = true;
    }

    public void computeCos() {
        cosResult = Math.cos(inputValue);
        showCosResult = true;
    }

    public void computeTan() {
        tanResult = Math.tan(inputValue);
        showTanResult = true;
    }
}
```

TrigonometryCalculatorController.vfp

```
<apex:page controller="TrigonometryCalculatorController">
  <apex:form >
    <apex:pageBlock title="Trigonometry Calculation">
      <apex:pageBlockSection title="Input Value">
        <apex:inputText value="{!inputValue}" label="Value" styleClass="matrix-input"/>
      </apex:pageBlockSection>
      <div style="text-align: center;" class="calculate-button">
        <center>
          <apex:commandButton action="{!computeSin}" value="Calculate Sin"
rerender="sinResultPanel" styleClass="calculate-button"/>
          <apex:commandButton action="{!computeCos}" value="Calculate Cos"
rerender="cosResultPanel" styleClass="calculate-button"/>
          <apex:commandButton action="{!computeTan}" value="Calculate Tan"
rerender="tanResultPanel" styleClass="calculate-button"/>
        </center>
      </div>
    </apex:pageBlock>

    <apex:outputPanel id="sinResultPanel">
      <apex:outputPanel rendered="{!showSinResult}">
        <apex:pageBlock title="Sin Result">
          <apex:outputPanel layout="block" style=" 10px;">
            <div class="matrix-row">
              <div class="matrix-cell">
                Sin: {!sinResult}
              </div>
            </div>
          </apex:outputPanel>
        </apex:pageBlock>
      </apex:outputPanel>
    </apex:outputPanel>
```

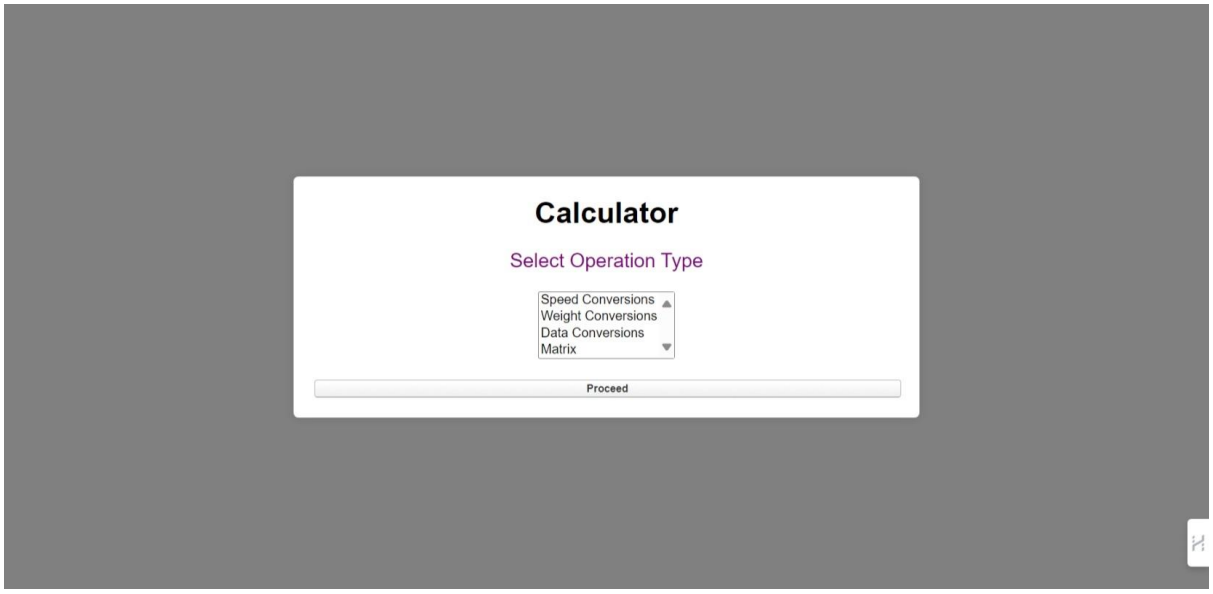
```
    <apex:outputPanel id="cosResultPanel">
        <apex:outputPanel rendered="{!showCosResult}">
            <apex:pageBlock title="Cos Result">
                <apex:outputPanel layout="block" style=" 10px;">
                    <div class="matrix-row">
                        <div class="matrix-cell">
                            Cos: {!cosResult}
                        </div>
                    </div>
                </apex:outputPanel>
            </apex:pageBlock>
        </apex:outputPanel>
    </apex:outputPanel>

    <apex:outputPanel id="tanResultPanel">
        <apex:outputPanel rendered="{!showTanResult}">
            <apex:pageBlock title="Tan Result">
                <apex:outputPanel layout="block" style=" 10px;">
                    <div class="matrix-row">
                        <div class="matrix-cell">
                            Tan: {!tanResult}
                        </div>
                    </div>
                </apex:outputPanel>
            </apex:pageBlock>
        </apex:outputPanel>
    </apex:outputPanel>
</apex:form>

<style>
```

```
        .matrix-input {
            width: 30px;
        }
        .matrix-row {
            display: flex;
            justify-content: center;
            align-items: center;
        }
        .matrix-cell {
            padding: 5px 10px;
            border: 1px solid #ccc;
            5px;
        }
    </style>
</apex:page>
```

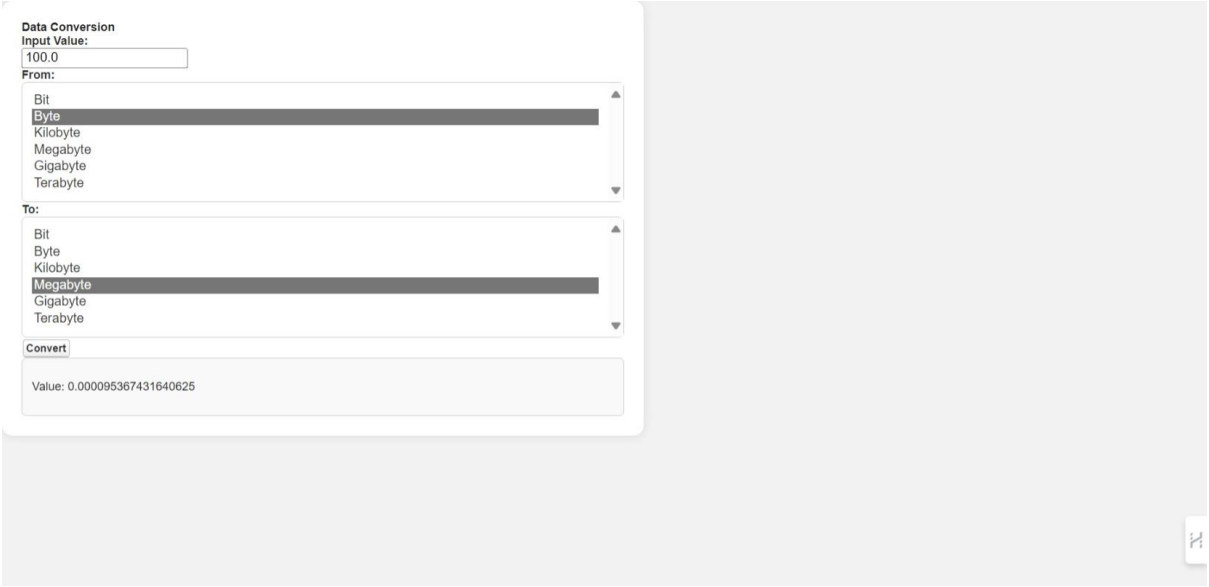
KAUSHAL BORKAR, SHIHSIR NARKHEDE



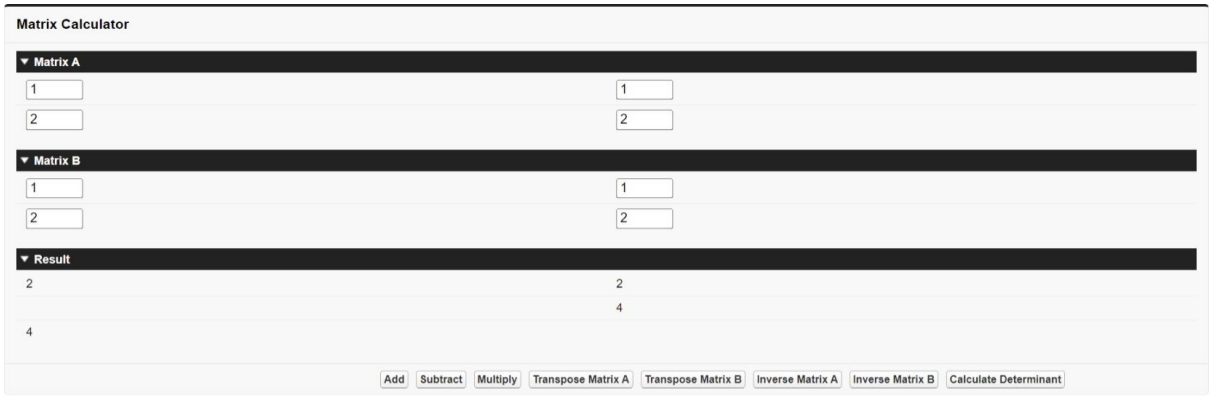
ADITYA SHINGOTE, ASHISH PANDEY, PRATIK MANJARE

PIYUSH TAMBE, SANYAM KUDALE

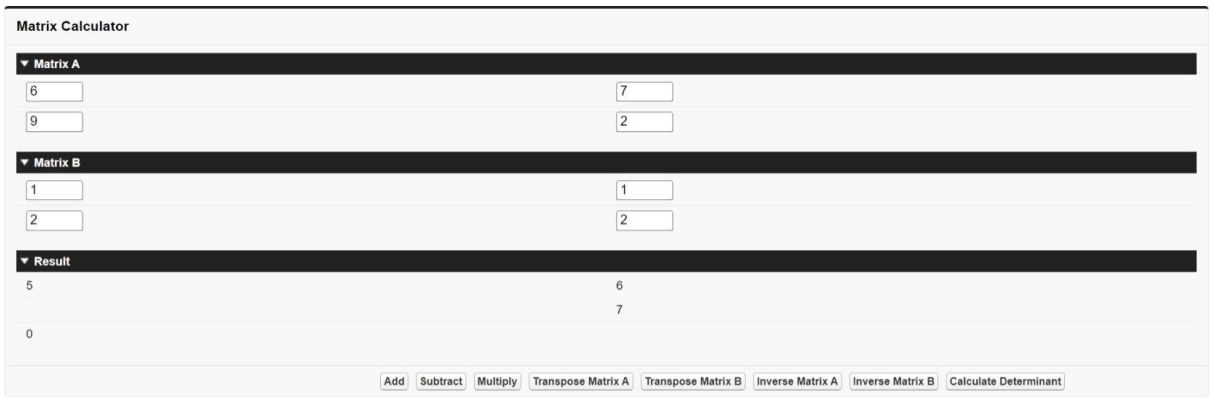
DHRUV GAGARE, GUNJAN NARKHEDE, DNYANESHWAR ALGULE



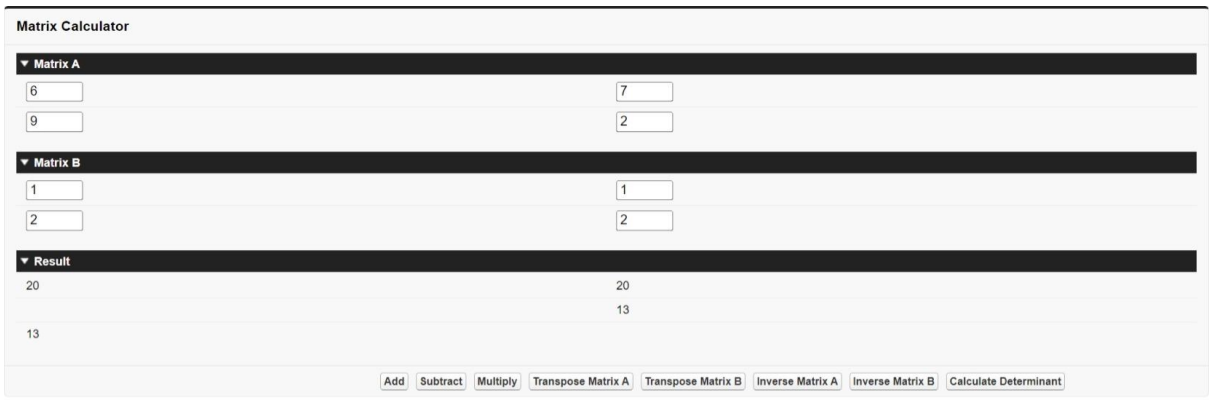
MALASHA RATHOD



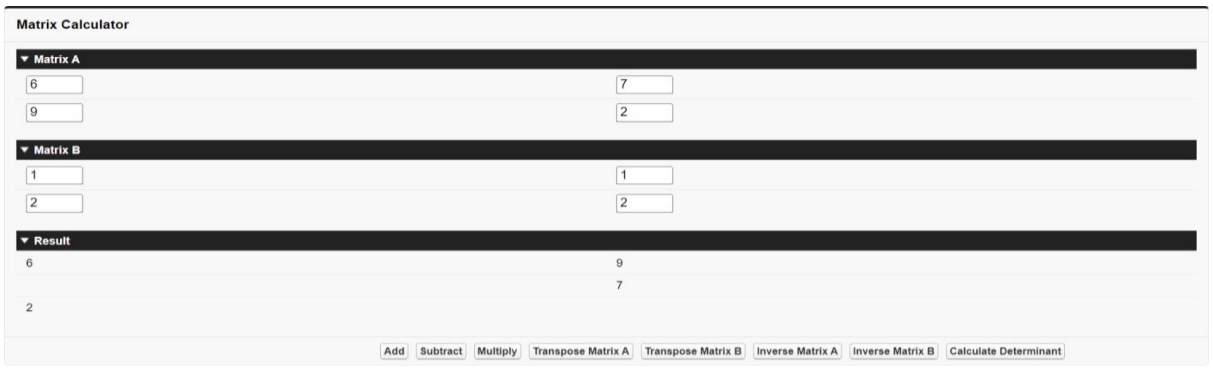
SANIKA CHAVAN



DHANASHREE THOMBARE



SAKSHI GULAVE



AKANSHA THEURKAR

Matrix Calculator

Matrix A

6	7
9	2

Matrix B

1	1
2	2

Result

1	2
	1
2	

Add | Subtract | Multiply | Transpose Matrix A | Transpose Matrix B | Inverse Matrix A | Inverse Matrix B | Calculate Determinant

Matrix Calculator

Matrix A

6	7
9	2

Matrix B

1	1
2	2

Result

0	0
	0
0	

Add | Subtract | Multiply | Transpose Matrix A | Transpose Matrix B | Inverse Matrix A | Inverse Matrix B | Calculate Determinant

Error:
Matrix is singular, inverse does not exist.

Matrix Calculator

Matrix A

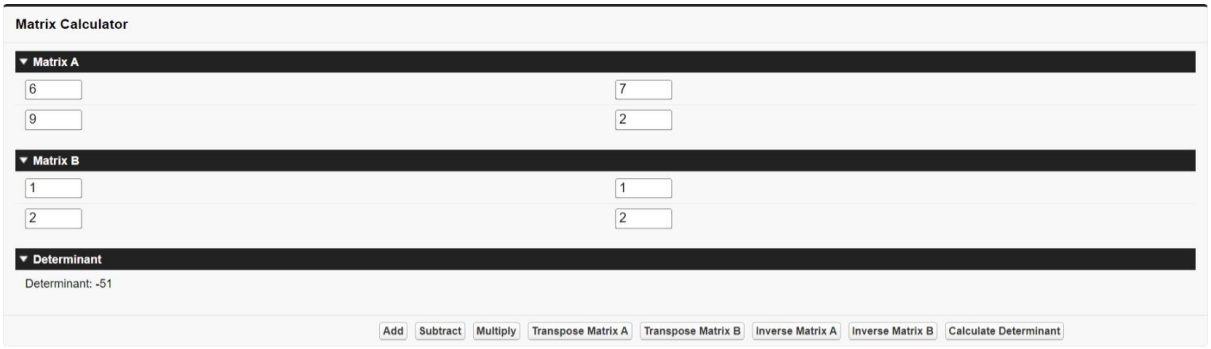
6	7
9	2

Matrix B

1	1
2	2

Add | Subtract | Multiply | Transpose Matrix A | Transpose Matrix B | Inverse Matrix A | Inverse Matrix B | Calculate Determinant

MENKA KHANDARE



SWARALI DESHPANDE