**Sales Forecasting Application**

**Overview:**

This application provides a sales forecasting tool based on historical sales data. It allows users to query sales for a specific year, apply percentage increases to forecast future sales, and visualize the results.

**Technologies Used**

* ASP.NET Core Web App( Razor Pages)
* C#
* SQL Server
* Chart.js for data visualization

**1)Clone the Repository**:

git clone https://github.com/your-username/SalesForecastingApp.git

cd SalesForecastingApp

**2)Database Setup**:

* Create a new SQL Server database.
* Import Data
* Create Stored Procedures

3)**Configure Connection String**:

* Update the connection string in ‘appsettings.json’ to point to your SQL Server database.

4) **Run the Application**:

* Open the solution in Visual Studio
* Build the project and run.

**Project Structure:**

Sales Forecasting :

pages:

* Sales.cshtml
* Sales.cshtml.cs

Models:

* Orders.cs
* Product.cs
* Return.cs

Data:

* SalesForecastingContext.cs

Appsettings.json

Program.cs

**Database Scripts:**

StoredProcedures.sql:

CREATE TYPE StateIncrement AS TABLE (

State NVARCHAR(100),

IncrementPercentage DECIMAL(5, 2)

);

GO

CREATE PROCEDURE GetYearlySalesWithIncrement

@Year INT,

@OverallIncrementPercentage DECIMAL(5, 2),

@StateIncrements StateIncrement READONLY

AS

BEGIN

DECLARE @Results TABLE (

State NVARCHAR(100),

TotalSales DECIMAL(18, 2),

IncrementPercentage DECIMAL(5, 2),

IncrementedSales DECIMAL(18, 2)

);

INSERT INTO @Results (State, TotalSales, IncrementPercentage, IncrementedSales)

SELECT

O.State,

SUM(P.Sales) - COALESCE(SUM(CASE WHEN ORR.OrderId IS NOT NULL THEN P.Sales ELSE 0 END), 0) AS TotalSales,

COALESCE(SI.IncrementPercentage, @OverallIncrementPercentage) AS IncrementPercentage,

(SUM(P.Sales) - COALESCE(SUM(CASE WHEN ORR.OrderId IS NOT NULL THEN P.Sales ELSE 0 END), 0)) \*

(1 + COALESCE(SI.IncrementPercentage, @OverallIncrementPercentage) / 100) AS IncrementedSales

FROM

Orders$ O

INNER JOIN

Products$ P ON O.OrderId = P.OrderId

LEFT JOIN

OrdersReturns$ ORR ON O.OrderId = ORR.OrderId

LEFT JOIN

@StateIncrements SI ON O.State = SI.State

WHERE

YEAR(O.OrderDate) = @Year

GROUP BY

O.State, SI.IncrementPercentage;

SELECT

State,

TotalSales,

IncrementPercentage,

IncrementedSales

FROM

@Results;

END;

GO

## Features

* Query sales data for a specific year.
* Apply percentage increase to forecast sales.
* Download forecasted data as a CSV file.
* Visualize sales data in charts.