**Technical Requirements**

**Project Overview:** Implement a full-stack solution that displays a report of the top-performing sellers by month, which can be filtered by branch. The frontend will be a React application that interacts with a .NET Core backend to read data from a provided CSV file and display the report. Authentication is not required at this stage.

**Backend Requirements (.NET Core)**

1. **Architecture and Technology:**
   * Use **.NET 8** for building the API.
   * The project should follow **clean architecture principles** (separation of concerns, clear data flow, etc.).
   * Implement the necessary service layer to handle the business logic.
   * The data source is a CSV file (provided). Use appropriate libraries to read and process this data (e.g., **CsvHelper**).
   * The backend should expose RESTful APIs to retrieve the list of branches and the top-performing sellers by month for the selected branch.
2. **Data Processing Logic:**
   * **CSV Data Reading:**
     + Read data from the provided CSV file (orders.csv located in the root of API project), which contains information about sellers, products, price, order dates, and branch.
     + Implement services to process and extract the relevant information (best-performing sellers by month) from the CSV file.
     + Use an efficient method to aggregate and calculate data, keeping performance in mind as the dataset could grow.
   * **Branch Filtering:**
     + Implement functionality to filter sellers based on the selected branch.
     + The branch list should be dynamically derived from the CSV file to ensure accuracy and avoid hardcoding.
3. **Unit Testing:**
   * Ensure that critical business logic is covered by **unit tests**. Several test cases would be enough to cover most common negative and positive test case scenarios.

**Frontend Requirements (React)**

1. **Technology Stack:**
   * Use **React** (v16.8+) to build the frontend.
   * Use **TypeScript** for static typing and better maintainability.
   * You can use third-party libraries if needed. Specify the reason for using a particular library.
   * Implement a responsive design using **CSS-in-JS** solutions or SCSS.
2. **User Interface:**
   * Implement a **dropdown list** to allow users to select a branch from the available branches.
   * Once a branch is selected, fetch the data from the backend and display a table showing the top-performing seller for each month (from January to December).
   * Each row in the table should include:
     + **Month** (e.g., January)
     + **Seller Name**
     + **Count of Total Orders**
     + **Total Price**
   * The layout should be clean and intuitive, ensuring users can easily select a branch and view the results.
3. **Testing:**
   * Write **unit tests** for key components, such as:
     + The branch dropdown list.
     + The table that displays the top sellers.
     + Proper rendering of seller data after branch selection.
   * Use a testing framework like **Jest** and **React Testing Library** to cover critical UI functionality.