Challenge Questions: Parse CSV, Validate, and Export to JSON

Objective

To design and implement a C# console application that parses a CSV, validates the format of the fields, and then exports a JSON representation of the data. Target time of this assignment is two hours or less.

Tasks

1. Read and parse the CSV data
2. Validate the data (ensure it matches the expected input) - Report invalid rows by printing an error message to the console that includes the row number and the reason for the error.
   a. customer_id is a positive integer
   b. first_name is a string
   c. last_name is a string
   d. email is a string, validate that it is a valid email format by utilizing regular expressions
   e. phone_number is in the format of XXX-XXX-XXXX (make sure it includes dashes and no parentheses or spaces)
   f. address is a string
   g. city is a string
   h. state is a string containing only two characters, validate that it is a valid United States abbreviation
   i. postal_code is a string with a maximum length of five digits and contains no nondigit characters
   j. car_make is a string
   k. car_model is a string
   l. car_year is an integer between 1900 and 2025
   m. license_plate is a string that has three consecutive characters or digits followed by the opposite characters or digits (ABC123 or 123ABC)
   n. purchase_date is a valid date within the year 2000 to present in format MM/DD/YYYY
   o. purchase_price is a valid decimal number
3. Transform the valid data into JSON format and save it to a file. The JSON output should be an array of objects, where each object represents a row of valid data. See the example below for the expected structure:

None

```
[
  {
    "customer_id": 1,
    "first_name": "John",
    // ... other fields ...
  },
  {
    // ... another valid row ...
  }
]
```

Deliverables

1. Source Code: A well-structured and commented codebase for the console application
2. README File: A clear and concise README file that includes:
    a. Instructions on how to build the project
    b. A brief description of the solution design and technologies used

(Optional) Multithreading: If you are familiar with multithreading techniques in C#, you can consider implementing them to optimize the performance of the application.