# CSV Report Console Application

## 1. Introduction

The CSV Report Application is a C# console based program that:  
- Reads book data from a CSV file.  
- Generates a formatted report either Plain Text or HTML.  
- Sends the report via email of user’s choice with the CSV file attached.  
- Supports dynamic columns and HTML pagination for large CSV files.  
- Runs in a loop so the user can process multiple reports without restarting.

2. Main Components

|  |  |
| --- | --- |
| Component | Description |
| BookCsvReader | Reads and parses the CSV file into Book objects. Supports dynamic columns by storing extra fields in a dictionary. |
| BookReportFormatter | Formats book data into Plain Text or HTML. HTML format supports dynamic columns and paginated tables for large datasets. |
| SmtpEmailService | Sends the report via SMTP email. Attaches the original CSV file and embeds the HTML report in the email body. |
| ReportApplication | Orchestrates reading CSV data, formatting the report, and sending email if the user chooses to. Handles user input for format selection, recipient email, and whether to continue or quit. |
| Program.cs | Entry point. Configures dependency injection, reads settings from appsettings.json, and loops application execution until the user quits. |
|  |  |

## 3. Workflow

1. Startup  
 - Reads configuration from appsettings.json (CSV path, email settings, etc.).  
 - Initializes services using dependency injection.

2. CSV Reading  
 - BookCsvReader opens the CSV file.  
 - First row = headers.  
 - Maps fixed fields (Title, Author, PublicationYear, etc…) to properties.  
 - Extra columns are stored in ExtraFields.  
  
3. Report Formatting  
 - BookReportFormatter detects all columns dynamically.  
 - Creates either:  
 • Plain text table with aligned columns.  
 • HTML table with pagination (default: 50 rows per page).  
  
4. Email Sending  
 - SmtpEmailService sends the HTML report.  
 - Attaches the original CSV file.  
 - Prompts user for recipient email if not set in config.  
  
5. Loop Control  
 - After each run, user can choose to:  
 • Continue (start again).  
 • Quit (exit program).

## 4. Features Handled

Reads any CSV file (dynamic columns & thousands of rows).  
 HTML pagination for large datasets.  
 Attaches CSV to email.  
 Interactive prompts for:  
 - Output format  
 - Email sending  
 - Recipient email  
 - Continue/Quit  
 Configurable via appsettings.json.  
 Logging for all key actions and errors.

## 5. File & Method Overview

### BookCsvReader.cs

• ReadAsync(string filePath, CancellationToken): Reads the CSV file and returns a list of Book objects. Skips invalid rows, logs warnings, and supports cancellation.  
• TryParseYear(string input): Parses an integer year from a string, returning 0 if invalid.  
• ParseBookFromLine(string line): Parses a CSV line into a Book object.  
• ParseCsvLine(string line): Splits a CSV line into fields, handling quoted values.

### BookReportFormatter.cs

• Format(IReadOnlyList<Book> books, ReportFormat format): Chooses the correct format method based on user selection.  
• FormatAsHtml(IReadOnlyList<Book> books): Builds an HTML table with dynamic columns and pagination.  
• FormatAsPlainText(IReadOnlyList<Book> books): Builds a plain text table with aligned columns.  
• CreateHtmlWrapper(string content): Wraps HTML content with full HTML boilerplate and styles.  
• CreateTableSeparator(int, int, int): Builds a text separator row for plain text.  
• TruncateIfNeeded(string text, int maxLength): Shortens text with ellipsis if it exceeds max length.

### SmtpEmailService.cs

• SendAsync(EmailReport report, CancellationToken): Sends an email with the report content, optionally with HTML formatting.  
• CreateSmtpClient(): Configures and returns an SMTP client.  
• CreateMailMessage(EmailReport report): Builds an email message with subject, body, and recipients.  
• ValidateEmailSettings(): Validates that email settings are correctly configured.

### ReportApplication.cs

• RunAsync(CancellationToken): Main workflow to read CSV, format report, display or email it, and log outcomes.  
• GetReportFormat(): Prompts the user for output format selection.  
• GetEmailPreference(): Asks the user whether to send the report via email.  
• SendEmailReport(string content, bool isHtml, CancellationToken): Prepares and sends an email with the report attached.

### Program.cs

• Main(string[] args): Application entry point. Builds host, retrieves IReportApplication, and loops until user quits.  
• CreateHostBuilder(string[] args): Configures app settings, dependency injection, and logging.

### AppSettings.cs & EmailSettings.cs

• AppSettings: Holds CSV file path, default output format, and auto-send email flag.  
• EmailSettings: Holds SMTP server configuration, credentials, recipient, and SSL setting.