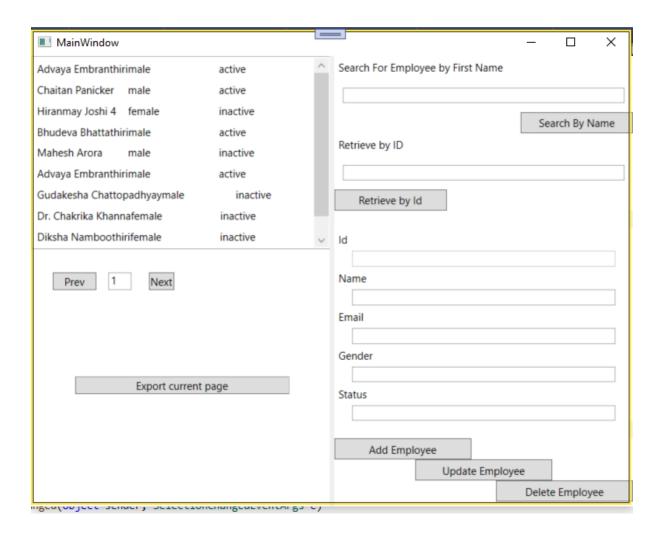
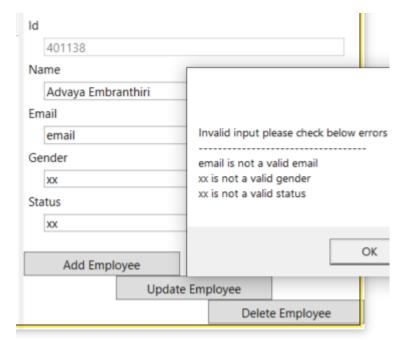
Solution has 5 projects explained below:

Client: (User interface WPF)

- Dependency
 - o Infrastructure
 - Automapper
- Content
 - o Search for employee using first name.
 - Enter search text and click Search by name button.
 - Retrieve Employee by Id
 - Enter employee Id and click Retrieve by Id button.
 - Details stack
 - Display for Id, Name, Email, Status and Gender
 - Add Employee
 - Add employee based on entered details (After validation).
 - Success message in case of success.
 - Failure message in case of failed.
 - Refresh the view to get the new entry displayed.
 - Update Employee
 - Update employee based on entered details (After validation).
 - Success message in case of success.
 - Failure message in case of failed.
 - Refresh the view to get the new entry displayed.
 - o Delete Employee
 - Delete the selected employee (either selected from the view or retrieved by Id).
 - Validation against empty selection.
 - View Employees
 - Display employees (Name, Gender and Status) by page number
 - o Page number controller
 - Export to CSV file
 - Export the current page to a csv file in the same location.
- Can be enhanced
 - o A better view for different flows instead of one single page.
 - o Using more appropriate visual toolbox match fields instead of blank test.
- Screenshots for reference







Id	
401138	
Name	
Advaya Embranthiri	
Email	
embranthiri_advaya@fadel-halvorson.info	
Gender	
male	
Status	
active	
Add Employee	
Update Employee	
Delete Employee	

Domain: (contains domain models)

- Dependency
- Content
 - o Employee.cs

Application: (contains the Repository contract)

- Dependency
 - o Domain
- Content
 - $\circ \quad \text{IEmplyeeService} \\$
 - o ItokenProvider

Infrastructure (Repositories with business logic to implement CRUD)

- Dependency
 - o Domain
 - o Application

• Content

- o Employee Repository
 - Implementation for all services
- Token Provider
 - Logic related to extract Token (To be extended later)
- Notes to Add
 - o Additional validation layer can be added here.
 - o Log mechanism for all request and exception from third parties
 - o Http Client repository to handle all requests

Testing

- Dependency
 - o Xnit
 - o Moq
- Content
 - Positive and Negative scenarios to cover all business logic operations on Employee repository.
- Screenshots

Type to sealer	
■ Tests (11 tests)	Success
▲ √ ⟨⟩ Tests (11 tests)	Success
■ ✓ EmployeeRepositoryTests (11 tests)	Success
√ AddEmployee_ReturnsError_WhenAddingWrongEmail	Success
√ AddEmployee_ReturnsError_WhenAddingWrongGender	Success
√ AddEmployee_ReturnsError_WhenAddingWrongStatus	Success
√ AddEmployee_ReturnsError_WhenAPIReturnsBadRequest	Success
√ AddEmployee_ReturnsPositiveInt	Success
√ DeleteEmployee_ReturnsFalse_WhenAPIReturnsNotFound	Success
√ DeleteEmployee_ReturnsTrue_WhenAPIReturnsSuccess	Success
√ GetEmployee_ReturnsEmployee_WhenAPIReturnsSuccess	Success
√ GetEmployee_ReturnsNull_WhenAPIReturnsNotFound	Success
√ GetEmployees_ReturnsEmployees_WhenAPIReturnsSuccess	Success
√ UpdateEmployee_ReturnsTrue	Success