**Application scope:  
  
Bussiness:**

For this task I worked on Code Challenge 1 – CSV Processing.  
It’s a ASP.NET MVC application that is intended to be used by business users in a Bank.It has the following functionalities:

* Import a CSV file in front end.
* Insert data in the CSV file in front end using Banking Model formular.
* Viewing the CSV data from file and the created data in a Table.
* Sorting, searching and filtering de data imported and created.
* Viewing the data in a Chart.The chart is dynamic,the user can choose between the 2 fileds provided which one to be represented in the Chart by clicking on the field.
* Downloading the modified CSV file.

The Banking Model Formular has the following fields:

* PIN : Personal identification Number;
* Name;
* Surname;
* LoanAmount : Is the total amount of a credit that has to be payed by a client;
* Interest Percent : The interest in procents that is used by the bank;
* IsVariable : Interest is variable or not.
* Interest Rate : Is the part of the loan that represents only the interest that will be payed to the Bank.
* Principal : Is the Principal from the total Loan that has to be payed to the Bank.

Flow of the application:  
 A user can upload a CSV document **that must be named TestReader** and to have a CSV extension. Also,the CSV file must provide **the same schema** that is provided with the example CSV file.After it hits the OK button,the data from the CSV file will be loaded in the front end in the Table.The user can search and filter the data.  
 The application provides a formular that can be used to insert data in the CSV file,data which can be seen in the Table.The application calculates Interest Rate and Principal **only when new data is inserted using the BankingModel.**  
 The formular uses the following formulas in order to calculate InterestRate and Principal:  
If Interest Percent is Variable:  
InterestRate = (EURIBOR + Fixed Rate) \* LoanAmount;   
If the InterestPercent is not variable:  
InterestRate = InterestPercent \* LoanAmount;  
Principal:  
Principal = LoanAmount – InterestRate;  
 The user can check the IsVariable box and the application applies automaticaly the EURIBOR interest rate + the fixed interest rate.EURIBOR interest rate and fixed interest rate are hardcoded in the application and cannot be manually chosed by a user.  
 If the interest rate is not variable,the InterestPercent provided in the BankingModel formular is used.  
 When the user hits the Create button,the application calculates the InterestRate and the Principal and inserts the data in the CSV file.It also displays the data in the Table.  
  
**Tehnical:**  
  
The MVC application uses the following types:  
In the *controller* there are 4 classes:

* **PathL**: class which stores the proprety **MapPath** which is used to store the location of the CSV file.
* **Matrix**: class which is used to store **CalculateInterestRate()** and **CalculatePrincipal()** methods which calculates the ***Principal*** and ***InterestRate***.I also stores the ***FixedRateE*** constant which is used for calculating InterestRate if the interest is not variable and ***EURIBOR*** rate.
* **IO**:class that stores the **Read()** method which reads the data from the CSV file.
* **MainController**:Controller class which holds 5 methods :  
  -***Graph()*** : method which reads data from the CSV file and sends it to the MainView in order to render the Graph.  
  -**Insert()** : method which inserts the data from the BankingModel formular in CSV file.  
  -**Upload()** : uploads the CSV file in the AppData/uploads folder.  
  -**Downlod()** : method used to download CSV file.  
  -**Main()** : Method which reads the data from CSV file and sends it to to View to be showed in the Table.

In the *model* is one class named **BankingModel** which implements the **IBankingModel** interface.