**Application Details**

**Application is built on below given frameworks, and these are required to run the application.**

1. .Net Framework 4.6.1
2. Entity Framework 6.2.0
3. MS SQL

**Instruction on how to install, configure and execute solution.**

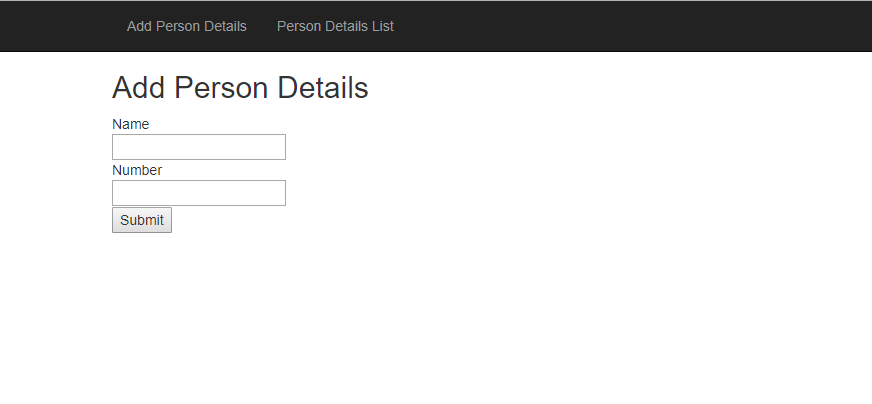
1. Download the solution from GitHub public repository.
2. Open the solution in VS 2015 or VS 2017.
3. Edit the **Web.config** file and set the DB connection string

<connectionStrings>

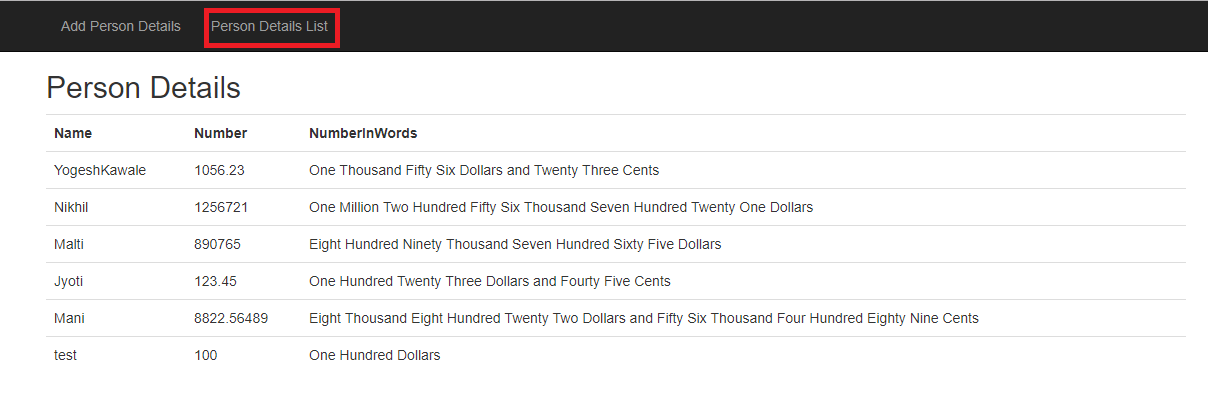
<add name="SQLConnection" providerName="System.Data.SqlClient" connectionString=" user id=###;password=######;Data Source=########;Database=assignment" />

</connectionStrings>

1. Re-build the solution.
2. Run the application from VS, or else you can deploy to local Site instance in IIS.
3. After running the application, code first migration approach will create the Database with name “**assignment**” and entities based on models we have in solution.
4. Application first page is “Add-Person-Details-Page” and it looks like this.



1. And second page is “Person-Details-List-Page” and it looks like this.



1. This pages also can be accessible through navigation on header, you can see in snaps.

**Technical details:**

1. Application is built for recording Person Details (includes his/ her Name and Number as amount) and listing Person Details with Name, Number, and Number in Words
2. Implementation is Code First approach migration in Entity Framework, in which AutomaticMigrationsEnabled = true; is configured.
3. Application has a controller called “Home” in which we have **Index** action method which is responsible for adding Person Details record.
4. Another action method “**PersonDetailsList**” which has a responsibility of getting records from **PersonDetails** entity and translate the numbers to words, and render the records after translation.
5. Application has a “**PersonDetailsRepository.cs**” inheriting **BaseEFRepository** abstract class contains all business and logic for all DB transactions.
6. For number translation to words, I have used the approach of identifying the number type (ones, tens, hundreds and thousands), and based on number type translation is done.
7. First identified the Whole Number and translated it
8. Then identified the Digit Number and translated it.