The project is has been build with .NET Core Web API .NET 6 .Has a web api project and some other class libraries. At the beginning when I saw the data from the json document that you sent I immediately thought that it would be better to use a NoSQL database to work on. But because I haven’t used MongoDB before but only Cosmos DB and this requires to pay it I decided to use SQL Server and Entity Framework Core to map my models and make the relationships between my entities. I used Onion Architecture with the use of Repository Pattern so I can create an abstraction layer between data access and the business logic layer of the application. By that we have a more loosely approach to access our data in the database. Also our code becomes cleaner, easier to maintain and reusable. Additionally, I created a service layer to extract all the business logic from Controllers. Also I created a repository manger and a service manger with the use of Lazy class to ensure the lazy initialization for the repositories and the services. This means that our instances is going to be created when we access them for the first time and not before that. For mapping between the Dtos and the models I used AutoMapper and also I used NLog library for my logging. For handling errors I created a middleware and put in the pipeline of the application and also action filters in the PUT/POST Actions of the controllers. All these are some of the best practices that I studied and read and I tried to implement them in my project.When I created my database I noticed that the date was in string format but not just simple format. So I saved the dates in the database with only dates and not with hours so by this would be easier to filter them. To calculate the end of day balances it was not so hard. The calculations was ok but I had to do a process to filtered the data by date and get the results for every day. I believe that in the mapping the logic maybe it’s not the right one because I didn’t know the business logic of the data so what properties I had to do nullable or not but I tried my best with the models and the relationships. Also I implemented for the transactions pagination, filtering and searching .I hope you will like what you are going to see and thank you for the opportunity because with this project I studied, searched and learned more