Radancy Coding Test

Develop code to implement the following requirements.

## Basic Bank system requirements

* A user can have as many accounts as they want.
* A user can create and delete accounts.
* A user can deposit and withdraw from accounts.
* An account cannot have less than $100 at any time in an account.
* A user cannot withdraw more than 90% of their total balance from an account in a single transaction.
* A user cannot deposit more than $10,000 in a single transaction.

## For Full Stack Roles

* Develop an HTTP based API to fulfill the requirements.
  + A real database is not required and can be faked with in-memory data structures.
* Develop a front-end UI to call the API and give feedback to the user if business rules are violated.

## For Front end only roles

* You can build all the requirements directly into a front-end application.
* The UI should give the user feedback if business rules are violated.

## For All Roles

* Write coded tests to cover business rule requirements.

## Focus for QA roles –

There is no need to develop a GUI, please provide a Web API instead.

We will focus our review on coding style, organization, testability and test coverage.

Don't worry about a real database. Feel free to fake it with in-memory data structures.

The completed work can be returned to us in a zipped/compressed package that we can extract, build and run; or through a public repository such as GitHub.  Please setup the project so that we can run the application locally in a container via Docker Compose.

At a minimum the solution should have an API that functions written in C#

The tests should be written in a way that they deliver results against that API.