**Documentation assignment 2**

**--Frontend—**

Name: Marcu Horea-Radu

Group: 30433

**1. Architecture**

For the development of this part of the assignment, I have used the Angular framework. Angular uses a declarative approach to building user interfaces, which means developers can specify what they want the interface to look like, and Angular will take care of the underlying code needed to create it. It uses two-way data binding, which means that any changes made to the data on the user interface are immediately reflected in the model and vice versa. Angular also includes a powerful set of features for creating reusable components, managing application state, handling forms, routing, and more. It uses TypeScript, a superset of JavaScript, which adds features such as static typing and class-based object-oriented programming to the language.

For creating the actual parts of the frontend, such as buttons and divs, I have used HTML and CSS. In order to display actual questions on the screen, I have generated some mock ones in typescript, which will be replaced with the real ones by using the http.get<> method, which I have only tested to see that it works as intended. The same will be changed for the answers of each question, them needing to be taken with a similar request. The main component of the application is the app.component page, which creates the navigation part and the content part. The content is then taken with the help of router-outlet, which allows us to show just the wanted content, which has been associated with the link on which we are currently on (either by typing it in or by going through different buttons).

As for the structure of the project, I have arranged the components in different packages, so that the parts which are connected are close to one another and a bit separated from the rest.

**2. Routing**

The actual routing of the application has been done in the app.module file, where all the components are declared and all the paths are defined for each component. In an Angular app, routing is managed by the Angular Router module. The router reads the URL in the browser's address bar and matches it to a defined route in the application's route configuration. When a user navigates to a particular URL, the corresponding component or view is loaded and displayed.

As for my application the current routes are the following:

const appRoute: Routes = [  
 {path: '', component: HomePageComponent},  
 {path: 'profile', component: UserPageComponent},  
 {path: 'questions', component: QuestionPageComponent},  
 {path: 'answers', component: AnswerPageComponent}  
]

So, if the link does not have anything after it, the home page is displayed, while if it has the defined words after it, the user accesses the different components which have been specified.