Chapter 1

Introduction

Nowadays, the life of an average person has become a lot more dynamic. Therefore, a lot more traveling is needed for daily work sometimes. And obviously not everyone is able to afford a car, while many other people probably have bigger cars that they use to travel between cities. So, if someone has 2 or 3 available places on his car and is traveling from city A to city B, why not take someone with you, who can not afford a car, and needs to travel in the same way. Basically, this is where ApoVjen application comes in handy.

1.1 Solution to the problem

I have lived in the capital city of my home country, Kosovo, in Prishtina, where I was doing my BSc. studies before I came in Hungary. The best faculties in Kosovo are located in capital city only, therefore the majority of people move there after finishing high school. It is also very common for students to visit their families on their cities every weekend or at least every other weekend, depending on their time availability. Obviously, most of these students can not afford a car, so they always use the public transport between the cities, or any other private service, if they can find one at the time they want. It is worth mentioning that public transport in Kosovo, is really not the best. Besides that, it goes a longer and older way, just to get more clients during the road, and does not use the International Highway, the prices for these trips are somewhat ridiculously expensive, especially for students. I’ve always been thinking, since there are other people that travel by car, most likely alone, it would not cost them anything to pick up 1 or 2 other people in his/her car if they are going the same way anyways. Since nowadays every single one of use has a smartphone, I thought that creating an app/website that would make carpooling possible, would be a great idea.

Chapter 2

User Documentation

To be able to use the application, users will first need to create an account. After successfully creating the account there will be user authentication which restricts the application to only those users who are authenticated, once verification is done the registered user will be automatically log in to the application with the registered credentials. User can reset their password in case they forget it or even change it to a different password if they wish so. Using this application, they will be able to read all the available trips, click on trip and see more detailed information about it. When will it departure, who is the trip owner, how many places are available to be booked, who else will be on that trip, and moreover if the user booked the trip, he can also join and participate on the trip group chat. The user is also able to create a new trip if he is interested to take some other people with him together.

2.1 System requirements

Ionic's earliest goal was to make it easy to develop mobile apps using web technologies like HTML, CSS, and JavaScript. Because of this foundation in web technologies, Ionic can run anywhere the web runs — iOS, Android, browsers, PWAs, and more.

*https://ionicframework.com/docs/reference/browser-support?\_gl=1\*y0gouc\*\_ga\*MTYyMDU2NTIxNC4xNjQ0MTU2Njk3\*\_ga\_REH9TJF6KF\*MTY1MTA5MjUzOC43NC4xLjE2NTEwOTI1NjUuMA..*

Mobile Browsers

Table

Description automatically generatedFor Mobile Browsers, depending on the ionic version of the app, the following Android and IOS versions can run the application.

Desktop Browsers

Table

Description automatically generatedBecause Ionic is based on web technologies, it works just as well on desktop browsers as it does on mobile devices.

2.2 Installing the application

The first thing we need to do is going to the GitHub repository which I will attach the URL at the end of the paper [1]. After, we need to clone the repository to the computer. After cloning, we need to install the latest ionic and Nest JS versions. After ionic is installed, we should open a new command prompt and change the execution of commands directory to the cloned repository directory and run the command: *ionic serve.*  
 After that a new browser page with a localhost URL will pop up and we can see the application up and running. It may not work properly yet because also backend server is not up and running yet. To make that work we follow the same steps as before. In another directory we clone the ApoVjen-server repository, and then on our computer, in that directory in command prompt, we can run the command: *npm start run: dev.* And now that we have both, the frontend and backend services, we can go ahead and use the application as expected.

2.3 Login Activity

The first-time user opens the already installed application, the login page will appear. There are two fields required to be filled in order to proceed inside the application, email address and a password. If the user is already registered the user can log in and it will redirect the user to the home page of the app. But in the case where the user is not registered, then registration is required beforehand.

Graphical user interface, application

Description automatically generated

2.3.1 Register New Account

To be able to create a new account, all the required fields should be filled, name, email address with any email domain, a password which should have complexity, contain at least one digit, at least one lowercase letter, at least one uppercase letter, at least one special character and length of at least 8 characters and a maximum of 20, and for the last required field user needs to confirm the password. After that the user should also fill the address information as well. These are just basic information, the street, number, zip code, city and state of the user. After clicking the register button, the form will be validated and checked in case every input is as expected. In the case where all the information is in the correct parameters, the user will be registered and automatically redirected to the home page of the application where he/she can proceed the normal use of it. In the case where any input parameter on the form is incorrect, the error message component will be triggered and shown below the wrong input.

A screenshot of a computer

Description automatically generated with low confidenceGraphical user interface, application

Description automatically generated

2.3.2 Reset password

User has the option to reset the password in case they forgot it, and they are not able to sign in into the account. The user should just write the email on the login form beforehand, and after clicking on the forgot password button, if the email is correct and it exists, the user will get a reset password link to their email, and a small pop up will show on the bottom of the screen notifying the user that the recovery email has been sent (Figure 1.3.1).

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated