Web Programming

Lab #4

UNIVERSITATEA TEHNICĂ A MOLDOVEI

Vangheli Dumitru

15 April , 2022

TABLE OF CONTENTS

Free Download

https://github.com/omenvng/PWlab4

SECTION I

Short story#: *Imagine, you're a developer at Sunny Entertainment GK, Tokyo. Ni hao. The company is struggling to find customers in Tokyo, so the marketing department proposes to launch a proof-of-concept app for running quizzes. The marketing campaign starts soon and you have to develop the app itself.*

The good part is that there is a Quiz API that can be consumed by your application to fetch data about the quizzes.

Tasks:

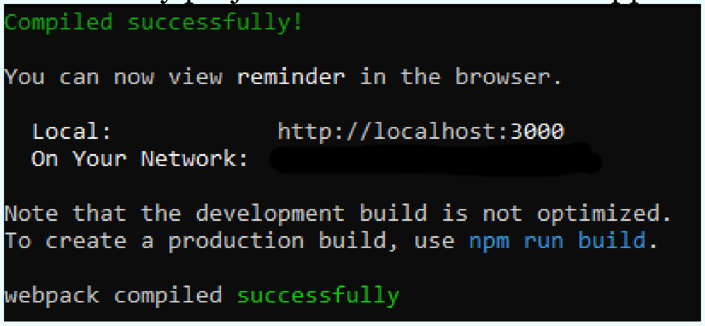
1. *Pick a frontend framework;*
2. *Create a web app that has the following function*
3. *It shows a landing page with different quizzes category*
4. *The user can pick a quiz and play it*
5. *After the game has ended, the user can see their score.*
6. *The app should have attractive UI;*
7. *Consume Quiz API to fetch data from backend server.*

Quiz:

Well actually i created a multiple choice quiz with React , well frankly speaking and being honest this is now possible thanks to the Create React App project which was create by Facebook.
The quiz is created in a style from " The Witcher " universe thus, the design and content is intended for that specific univers of books and videogames.
The main feature of this Quiz App is that its fetching the quizzes from the API(https://pure-caverns-82881.herokuapp.com/api/v54/quizzes) based on user selection of category and number of questions.
So by using " Create React App " I got a moder workflow with Webpack , Babel ... all already configured for work.
All this allowed me to jump into writing my code straight away.

Setup:

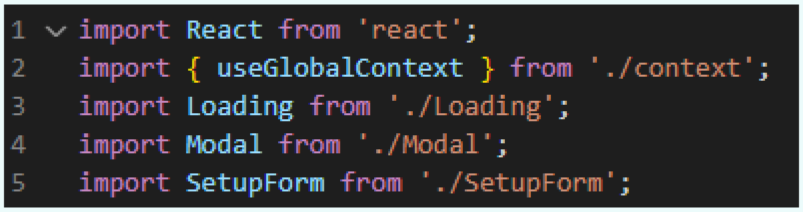
As I said my project was created on React app with a Node Backend.



```
Compiled successfully!

You can now view reminder in the browser.

  Local:            http://localhost:3000
  On Your Network:  [redacted]

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

The app components:

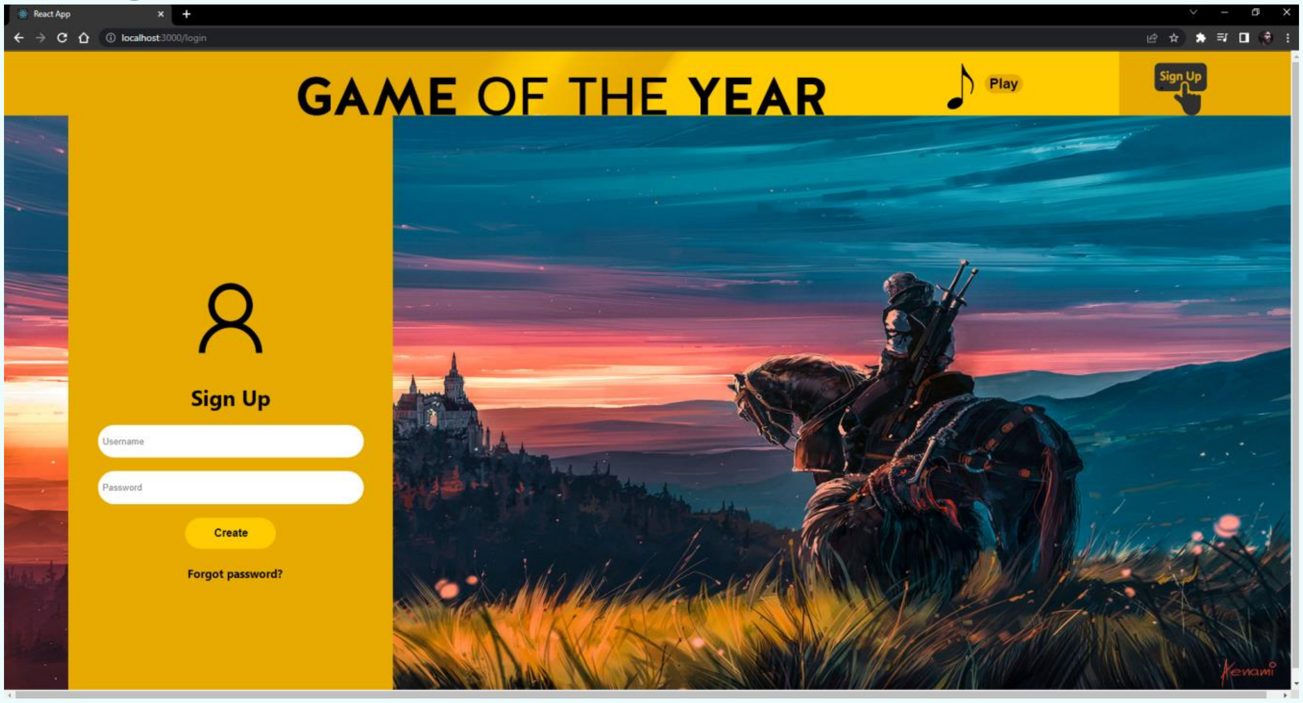
Well creating this app involves adding small components to build up your app.



```
import React from 'react';
import { useGlobalContext } from './context';
import Loading from './Loading';
import Modal from './Modal';
import SetupForm from './SetupForm';
```

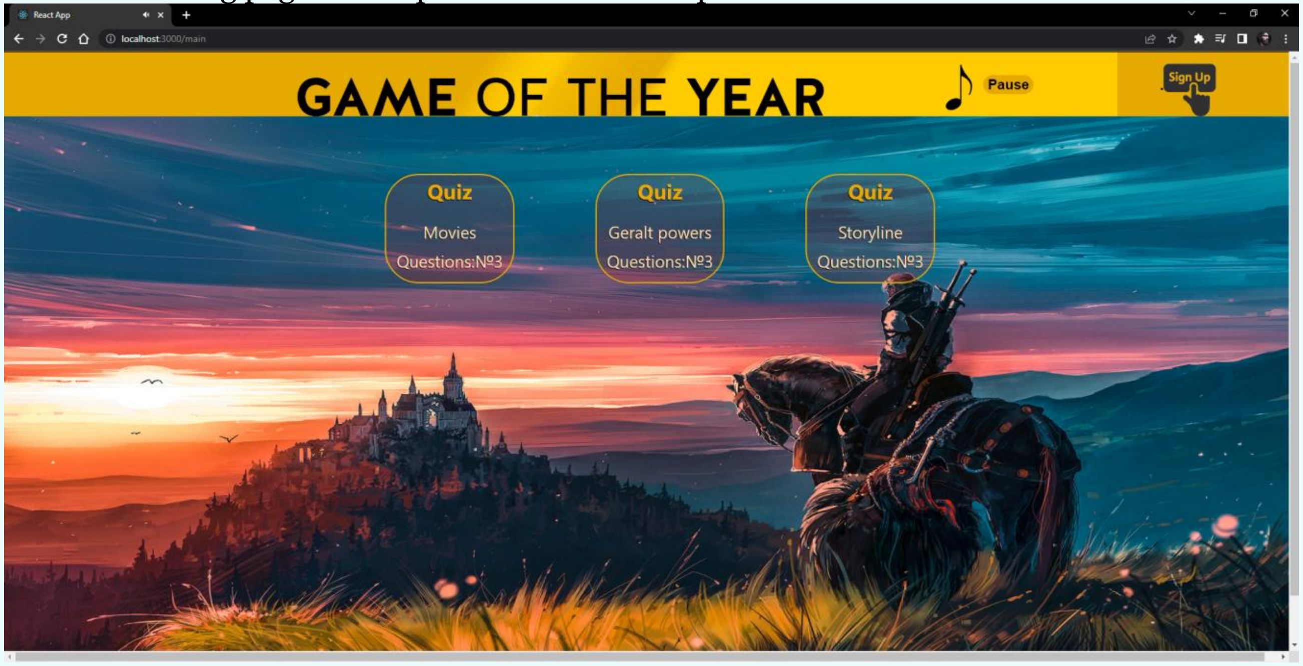
These components will be composed together through a container component to build my quiz.

Design of the App:



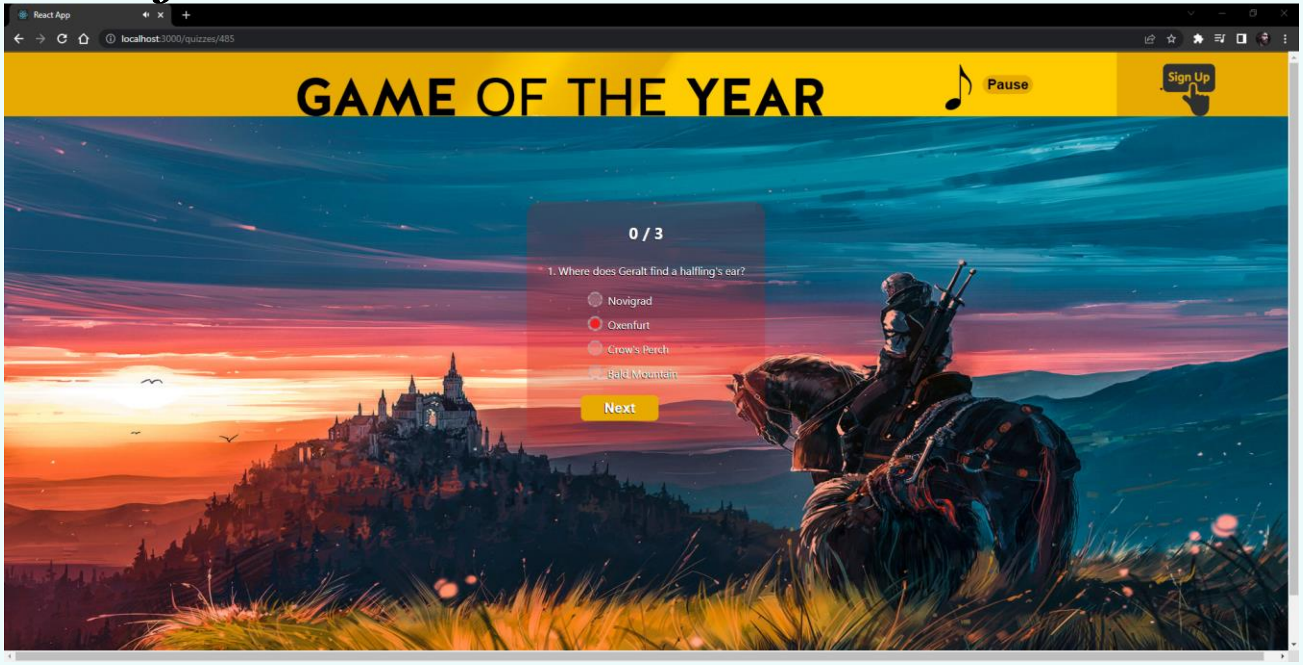
The login process is fetched with User API which are created using /api/v54/users endpoint. Where are stored user IDs for submitting responses.

Well the landing page of the quiz consists from 2 parts :



1. It consists of the number of questions selected, the maximum value of which can be set manually.
2 . And the category part ,which you can choose before starting the quiz game.

The game itself consists of a question and four possible answers and only one correct one.

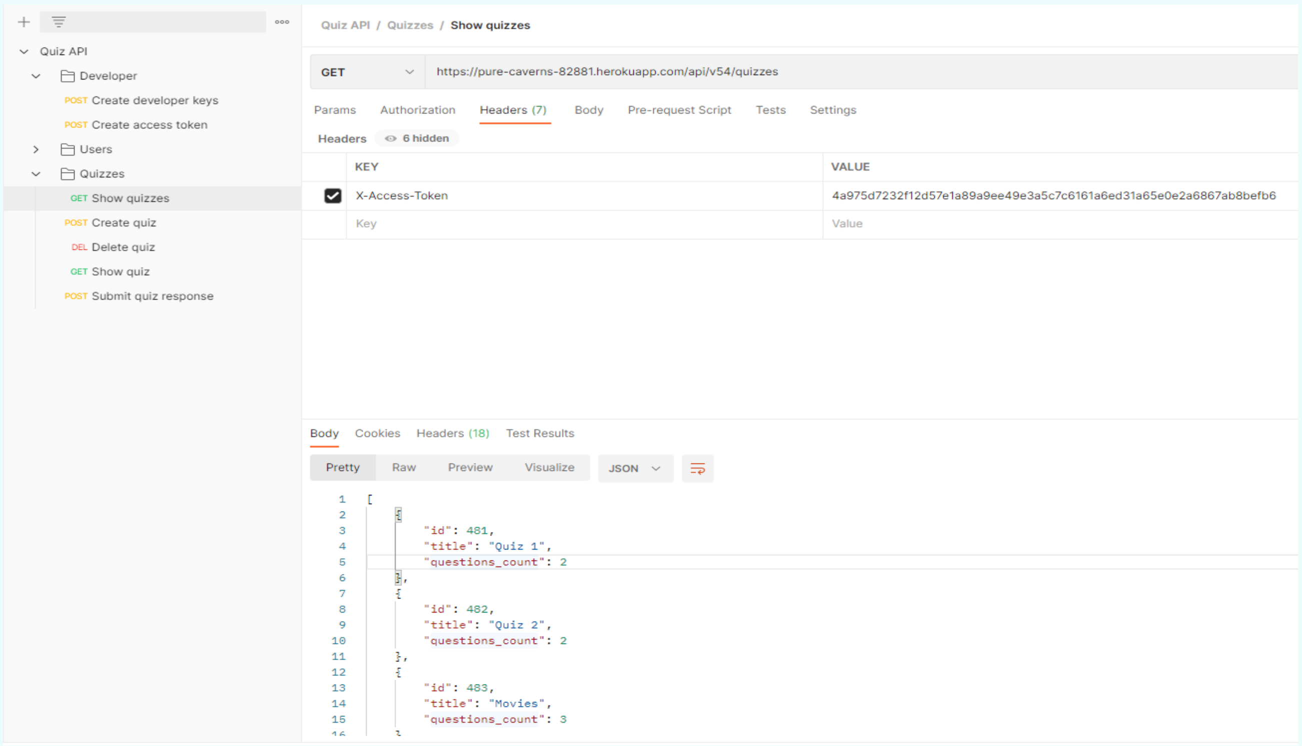


SECTION II

Quiz fetching:

Ouiz API

Open Quiz API offers us an API that we can call to receive a given number of questions on a given topic. Quizes are creating same as User by accesing /api/v54/quizzes endpoint. And the process of submitting reposnes for a quiz for a particular user we use api/v54/quizzes/{{quiz-id}}/submit endpoint.

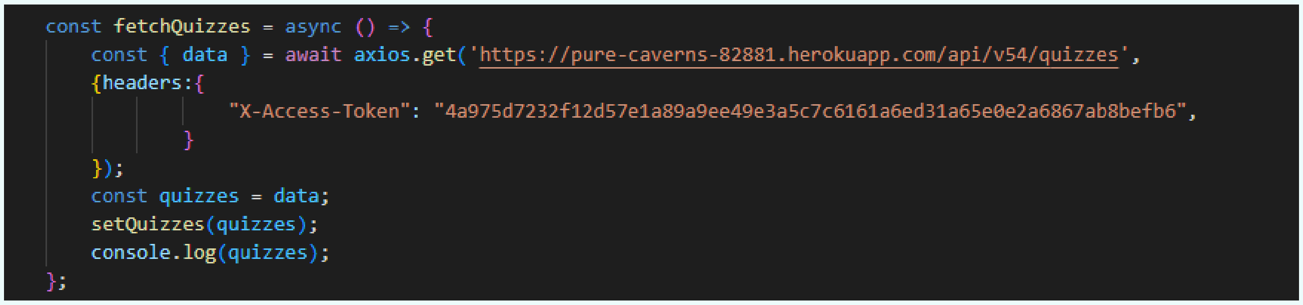


GET https://pure-caverns-82881.herokuapp.com/api/v54/quizzes

	KEY	VALUE
✓	X-Access-Token	4a975d7232f12d57e1a89a9ee49e3a5c7c6161a6ed31a65e0e2a6867ab8befb6
	Key	Value

```
[
    {
        "id": 481,
        "title": "Quiz 1",
        "questions_count": 2
    },
    {
        "id": 482,
        "title": "Quiz 2",
        "questions_count": 2
    },
    {
        "id": 483,
        "title": "Movies",
        "questions_count": 3
    }
```

You can generate a custom token for accesing the quizzes API.

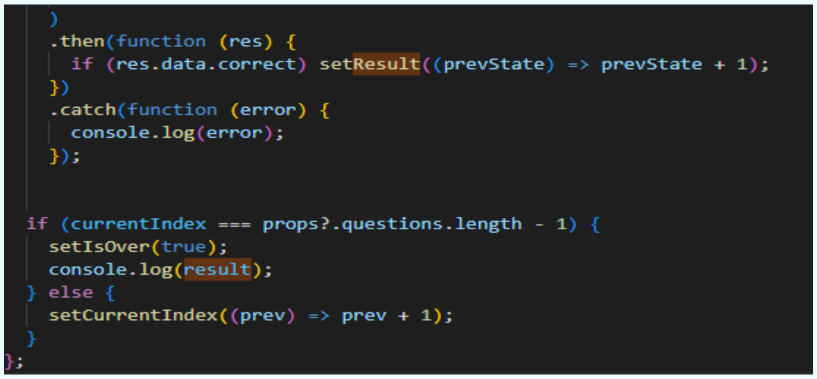


```
const fetchQuizzes = async () => {
    const { data } = await axios.get('https://pure-caverns-82881.herokuapp.com/api/v54/quizzes',
    {headers:{
            "X-Access-Token": "4a975d7232f12d57e1a89a9ee49e3a5c7c6161a6ed31a65e0e2a6867ab8befb6",
        }
    });
    const quizzes = data;
    setQuizzes(quizzes);
    console.log(quizzes);
};
```

The questions variable is going to be an array of question objects we get from the API. The loaded variable is going to let us know if the questions have beenloaded yet, so we don’t accidentally try to access them too early.

Managing Score

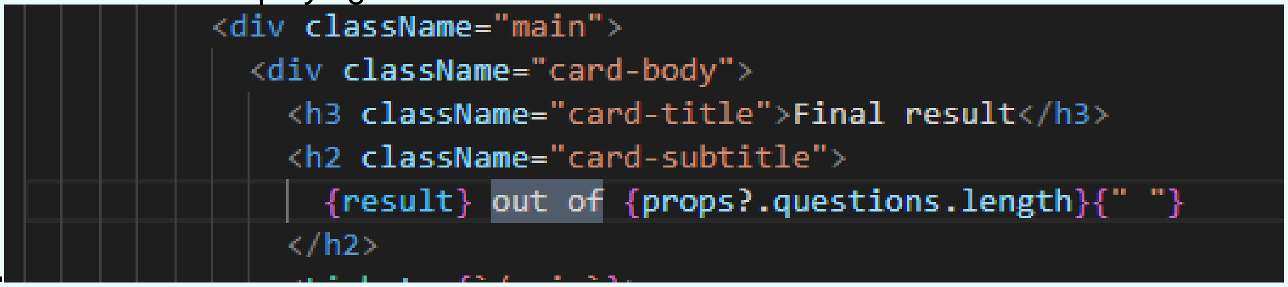
The next step is going to be to manage the user's score. To do so, we are going to create some stateful variables:



```
    )
    .then(function (res) {
      if (res.data.correct) setResult((prevState) => prevState + 1);
    })
    .catch(function (error) {
      console.log(error);
    });

  if (currentIndex === props?.questions.length - 1) {
    setIsOver(true);
    console.log(result);
  } else {
    setCurrentIndex((prev) => prev + 1);
  }
};
```

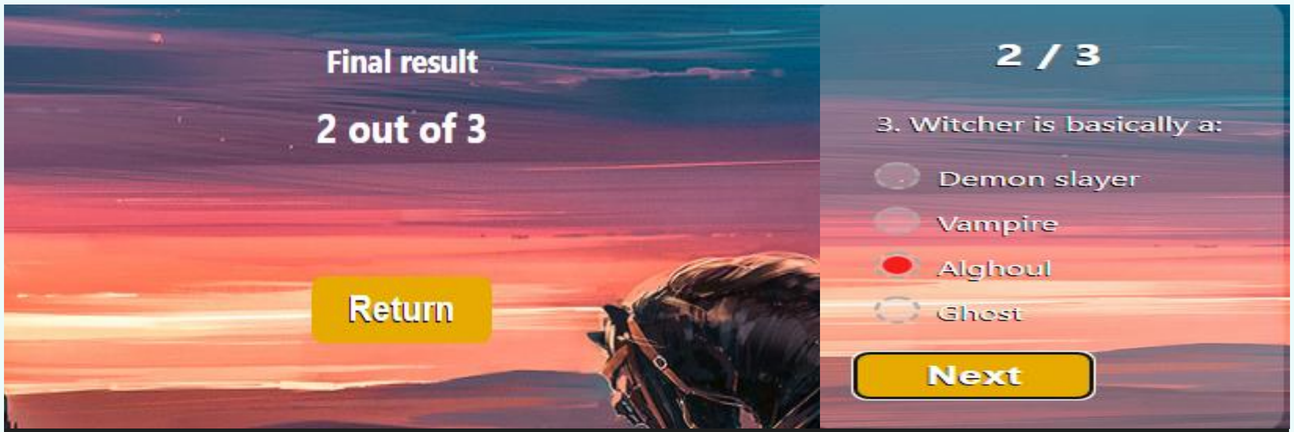
The **result** variable will count and store the correct answers while the **currentIndex** will count the total of imported questions.

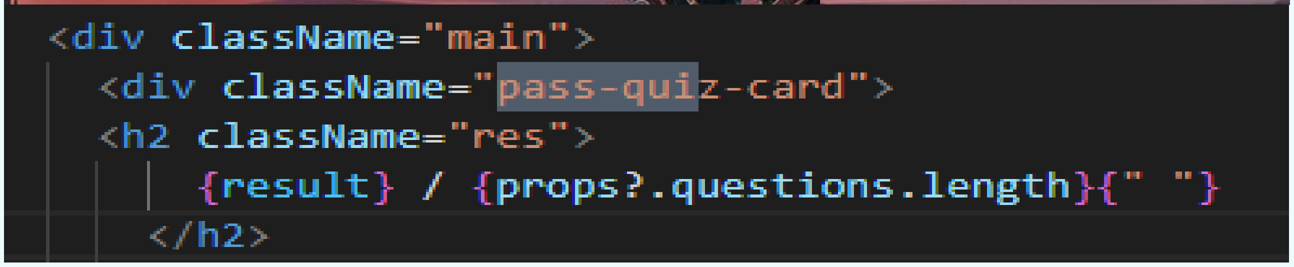
And of course displaying them in the UI:



```
<div className="main">
  <div className="card-body">
    <h3 className="card-title">Final result</h3>
    <h2 className="card-subtitle">
      {result} out of {props?.questions.length}{" "}
    </h2>
```

.





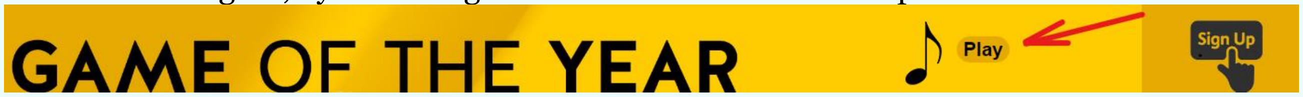
```
<div className="main">
  <div className="pass-quiz-card">
  <h2 className="res">
    {result} / {props?.questions.length}{" "}
  </h2>
```

Style:

Create React App configures Webpack for us so that we can define separate CSS files for each module. It will then bundle all of our CSS into one file upon saving. We won't be diving too much into styling CSS the easy way we can just take from the Github repository here, and replace the current contents of index.css with it.

Music:

While you are answering the quzzies, you can be accompanied by beautiful music from the taverns of Novigrad, by launching the track in advance in the top headers menu.



References

- **Frankly speaking this Indian guide did all the learning job :**

https://www.youtube.com/watch?v=dg7XmuLvsbs&t=1648s

~~To be continued...~~