|  |
| --- |
| PIANO-APP |
|  |

|  |  |
| --- | --- |
| 26/11/2021 | Team-2 Virtual Piano using React |

The project is a web-based application of a functional **Virtual Piano** built with React hooks and React Sound-font library.

Developed By: Shivangi Patel & Felipe Prudencio

PIANO-APP

Team-2 Virtual Piano using React

# REQuireMEnt Gathering

As per Instruction we had to choose one out of contribution to open-source project and developing a new project. After discussing we decided on developing a new project on 2nd October 2021.  
The projects requirements were to build a web-based application on React integrating diverse technologies to have an app that would meet the three principles learned throughout the course: Verifiable, does it work as it should? Repeatable, the result is repeated and Automated, the project is built automatically and has continuous development.

We researched on following websites for our project Idea

* <https://www.freecodecamp.org/news/8-reactjs-project-ideas-to-start-learning-by-doing/>
* <https://www.freecodecamp.org/news/5-react-projects-you-need-in-your-portfolio/>
* <https://www.edopedia.com/blog/open-source-html5-and-javascript-games/>

And finally, we decided on building a Piano based on ReactJS. The reason for choosing ReactJS was very straightforward as both of us were confident with the language. We wanted to do something that we had never developed, and which would also help us to show on our portfolio.

After deciding on the topic, we started discussing on how to build the piano app, from where we would get the sounds/notes, how we would map the key-board keys to the notes and how to render the application.

We referred to the following links to study how the application would be. Our research includes the example of vanilla JavaScript applications where we studied how will get the exact note as of real piano.

* <https://codepen.io/noogn/pen/LAiDz>
* <https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Simple_synth>

We also came across the in-built react library “react-piano” rather it was a npm package (refer the link below), but we decided not to use this because library had built in react functional components to display keys on screen and handling key press or click events directly, that would make our job too much easy and we wanted to learn something out of the project.

* <https://www.npmjs.com/package/react-piano>

Finally, we decided on using react library Soundfont-player and react hooks to developer application.

# Diagrams

# Sequence diagram

# Graphical user interface Description automatically generated

# Class-Diagram

# Graphical user interface, application, Teams Description automatically generated

# Meetings

The team met every Saturday during lab time and a bit afterwards to discuss the projects direction and make decisions. The scrum and sprints were made using the Trello app, where the team defined and assigned tasks to be completed. The teammates talked during the week about the development and completion of the sprints via slack chat. The division of research and coding was divided in a way that each member had a contribution to the project development.  
Most decisions were made in person, and we would often document it with a message on the Slack team chat. The GitHub commit messages also serve as documentation of changes made to the app code.

# Testing

We have used in-built library to get the piano notes and play them. Mapped the notes with keyboard key and UI buttons.

Writing the test case and building automation or coverage was not possible.

Thus, conducted acceptance testing.

Tested tune comparing with application build on same library and recorded the same.

Also tested the notes with virtual piano and have recorded.

Video included in submission.

# Debugging

For debugging the team used the built-in features of VS code in which we would create breakpoints to check the individual functionality of the piano functions and to check how the components interacted with each other.  
We also used the warning and error messages to identify problems with the builds and correct them as needed.

# DOCKING

# We used open-source platform Docker for containerization.

# Followed standard process for creating docker image.

# Step-1: Added Docker file in project referred Docker documentation for creating the file for react application.

# Step-2: Defined workflow for docker image. Referred lab exercise.

# Step-3: Added docker login password and username as secret in GitHub

# Step-4: Executed the workflow.

# Docker Link:

# https://hub.docker.com/repository/docker/shivangipatel9170/piano-docker

# continuous Deployment

The team implemented continuous deployment by having the app in GitHub together with a Docker image. This works in the way that every time the team pushes a change to the production code the repo and image are updated automatically, and the new code is used.  
When the base app was uploaded to GitHub it just had the basic functionalities so that the app would work. As the work on the app progressed and new features were added the deployment to GitHub and later Docker was updated as well to reflect the changes.

# github URL

https://github.com/ShivangiPatel1/Piano-app