|  |  |
| --- | --- |
| **Project Title** | **Calculator App** |
| **Technologies** | ReactJS |
| **Domain** | Education |
| **Project Level** | Easy |
| **Organization** | iNeuron Intelligence Private Limited |

Table of Content

Contents

1. [Problem Statement 2](#_TOC_250010)

................................................................................................................................................................... 2

1. [Project Evaluation metrics: 2](#_TOC_250009)

[2.2. Database 2](#_TOC_250008)

1. [Submission requirements: 2](#_TOC_250007)

................................................................................................................................................................... 2

* 1. [High-level Document 2](#_TOC_250006)
  2. [Low-level document 2](#_TOC_250005)
  3. [Architecture 3](#_TOC_250004)
  4. [Wireframe 3](#_TOC_250003)
  5. [Project code 3](#_TOC_250002)
  6. [Detail project report 3](#_TOC_250001)
  7. [Project demo video: 3](#_TOC_250000)
  8. [The project LinkedIn a post 3](#_bookmark0)

# Problem Statement:

**Calculator App**

Build a simple calculator app which performs simple arithmetic operations such as addition, subtraction, multiplication and division.

# Project Evaluation metrics:

* + - 1. **Code:**
* You are supposed to write code in a modular fashion
* Safe: It can be used without causing harm.
* Testable: It can be tested at the code level.
* Maintainable: It can be maintained, even as your codebase grows.
* Portable: It works the same in every environment (operating system).
* You have to maintain your code on GitHub.
* You have to keep your GitHub repo public so that anyone can check your code.
* Proper readme file you have to maintain for any project development.
* You should include basic workflow and execution of the entire project in the readme file on GitHub.
* Follow the coding standards.

# Database:

**No database is used**

# Submission requirements:

# High-level Document:

**[ReactJS | Calculator App ( Structure )](https://www.geeksforgeeks.org/reactjs-calculator-app-structure/?ref=lbp)**

* [ReactJS | Calculator App ( Building UI )](https://www.geeksforgeeks.org/reactjs-calculator-app-building-ui/?ref=lbp)
* [ReactJS | Calculator App ( Adding Functionality )](https://www.geeksforgeeks.org/reactjs-calculator-app-adding-functionality/?ref=lbp)
* [ReactJS | Calculator App ( Styling )](https://www.geeksforgeeks.org/reactjs-calculator-app-styling/?ref=lbp)
* [HTML Calculator](https://www.geeksforgeeks.org/html-calculator/?ref=lbp)
* [HTML Scientific Calculator](https://www.geeksforgeeks.org/html-scientific-calculator/?ref=lbp)
* [ReactJS | Calculator App ( Introduction )](https://www.geeksforgeeks.org/reactjs-calculator-app-introduction/?ref=lbp)
* [Sum of an array using MPI](https://www.geeksforgeeks.org/sum-of-an-array-using-mpi/?ref=lbp)
* [MPI – Distributed Computing made easy](https://www.geeksforgeeks.org/mpi-distributed-computing-made-easy/?ref=lbp)
* [Cloud Computing](https://www.geeksforgeeks.org/cloud-computing/?ref=lbp)
* [Cloud Based Services](https://www.geeksforgeeks.org/cloud-based-services/?ref=lbp)
* [Types of Cloud](https://www.geeksforgeeks.org/types-of-cloud/?ref=lbp)
* [Virtualization in Cloud Computing and Types](https://www.geeksforgeeks.org/virtualization-cloud-computing-types/?ref=lbp)
* [Hypervisor](https://www.geeksforgeeks.org/hypervisor/?ref=lbp)
* [Characteristics of Cloud Computing](https://www.geeksforgeeks.org/characteristics-of-cloud-computing/?ref=lbp)
* [Issues in Cloud Computing](https://www.geeksforgeeks.org/issues-in-cloud-computing/?ref=lbp)
* [Anatomy of Cloud Computing](https://www.geeksforgeeks.org/anatomy-of-cloud-computing/?ref=lbp)
* [Cloud Computing Services in Financial Market](https://www.geeksforgeeks.org/cloud-computing-services-in-financial-market/?ref=lbp)
* [Economics of Cloud Computing](https://www.geeksforgeeks.org/economics-of-cloud-computing/?ref=lbp)
* [How to Become a Full Stack Web Developer in 2021?](https://www.geeksforgeeks.org/how-to-become-a-full-stack-web-developer-in-2021/?ref=lbp)
* [How to Become a Full Stack Web Developer in 2019 : A Complete Guide](https://www.geeksforgeeks.org/how-to-become-a-full-stack-web-developer-in-2019-a-complete-guide/?ref=lbp)
* [What is full stack development ?](https://www.geeksforgeeks.org/what-is-full-stack-development/?ref=lbp)
* [How to choose a Technology Stack for Web Application Development ?](https://www.geeksforgeeks.org/how-to-choose-a-technology-stack-for-web-application-development/?ref=lbp)
* [MERN Stack](https://www.geeksforgeeks.org/mern-stack/?ref=lbp)
* [Introduction to Express](https://www.geeksforgeeks.org/introduction-to-express/?ref=lbp)
* [Installation of Node.js on Linux](https://www.geeksforgeeks.org/installation-of-node-js-on-linux/?ref=leftbar-rightbar)
* [Top 10 Projects For Beginners To Practice HTML and CSS Skills](https://www.geeksforgeeks.org/top-10-projects-for-beginners-to-practice-html-and-css-skills/?ref=leftbar-rightbar)
* [How to insert spaces/tabs in text using HTML/CSS?](https://www.geeksforgeeks.org/how-to-insert-spaces-tabs-in-text-using-html-css/?ref=leftbar-rightbar)
* [How to update Node.js and NPM to next version ?](https://www.geeksforgeeks.org/how-to-update-node-js-and-npm-to-next-version/?ref=leftbar-rightbar)
* [How to calculate the number of days between two dates in JavaScript ?](https://www.geeksforgeeks.org/how-to-calculate-the-number-of-days-between-two-dates-in-javascript/?ref=leftbar-rightbar)

Sample link:

<https://www.geeksforgeeks.org/reactjs-calculator-app-structure/?ref=lbp>

Low-level document:

* [ReactJS | Calculator App ( Building UI )](https://www.geeksforgeeks.org/reactjs-calculator-app-building-ui/?ref=lbp)
* [ReactJS | Calculator App ( Adding Functionality )](https://www.geeksforgeeks.org/reactjs-calculator-app-adding-functionality/?ref=lbp)
* [ReactJS | Calculator App ( Styling )](https://www.geeksforgeeks.org/reactjs-calculator-app-styling/?ref=lbp)
* [HTML Calculator](https://www.geeksforgeeks.org/html-calculator/?ref=lbp)
* [HTML Scientific Calculator](https://www.geeksforgeeks.org/html-scientific-calculator/?ref=lbp)

[ReactJS | Calculator App ( Introduction )](https://www.geeksforgeeks.org/reactjs-calculator-app-introduction/?ref=lbp)

Sample link:

* <https://www.geeksforgeeks.org/reactjs-calculator-app-structure/?ref=lbp>

# Architecture:

Architecture document design for your project;

In our [previous](https://www.geeksforgeeks.org/reactjs-calculator-app-introduction/) article, we have talked about a Calculator app we are going to develop and also have seen a glimpse of our final project. In this article, we will get our hands ready to start the development of our first application. We have told this earlier also that every application we will develop in React will be made up of pieces called components. We can see a UI broken down into multiple individual pieces called components and work on them independently and merge them all in a parent component which will be your final UI. So let’s now try to break down the UI of calculator App into smaller pieces. We will get the following elements after breaking the UI into small pieces:

* **Calculator Title**: This is the title at the top of our application, “GeeksforGeeks Calculator”.
* **Output Screen**: This will be our output screen, where all text will be shown. Like the input that the user will type and the answer calculated from the user input. So, we can again break down this into two smaller pieces as shown below:
  + **Question Output**: This will be the input given by the user.
  + **Answer Output**: This will be the result calculated from user input.
* **Buttons**: As you have seen in the image of our calculator app, we had used a lot of buttons for the input numbers from ‘0-9’, operators, to clear the screen, for backspace etc.

Sample link:

* <https://www.geeksforgeeks.org/reactjs-calculator-app-structure/?ref=lbp>

# Wireframe:

* **Calculator Title**: This is the title at the top of our application, “GeeksforGeeks Calculator”.
* **Output Screen**: This will be our output screen, where all text will be shown. Like the input that the user will type and the answer calculated from the user input. So, we can again break down this into two smaller pieces as shown below:
  + **Question Output**: This will be the input given by the user.
  + **Answer Output**: This will be the result calculated from user input.
* **Buttons**: As you have seen in the image of our calculator app, we had used a lot of buttons for the input numbers from ‘0-9’, operators, to clear the screen, for backspace etc.

Demo link:

https://www.google.com/search?q=calculator+app+project+by+reactjs+wireframe+architecture&hl=en&sxsrf=APwXEdeRpv1bAiWJ6CDwFJnkgrkBUdvaRQ:1681360540099&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjukO2NhKb-AhVkSmwGHZY9AHwQ\_AUoAXoECAEQAw&biw=1366&bih=657&dpr=1#imgrc=GSXefdmXjKWTaM

# Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link:

# <https://github.com/rajkumar-aerpula>

# Detail project report:

You have to create a detailed project report and submit that document as per the given sample. Demo link:

<https://github.com/rajkumar-aerpula>

Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

Demo link:

[Project sample link](https://www.youtube.com/watch?v=hBLdlx_U4L8)

# The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.

Demo link:

https://www.linkedin.com/posts/raj-kumar-aerpula-9082b9101\_github-rajkumar-aerpulaprojectcalculator-activity-7051820970877075457-h1L8?utm\_source=share&utm\_medium=member\_android