

**MINI PROJECT – REPORT**

**Bachelor’s Of Technology**

**In**

**Computer Science & Engineering**

**SUBMITTED TO :**

**SUBMITTED BY:**

ALOK PANDEY(1900520100005)

DURGESH MAURYA(1900520100005)

# Table Of Contents :

* Acknowledgement
* Introduction
* Software Requirement
* Hardware Requirement
* Overview
* Program
* Code-
* Backend
* Frontend
* Snapshots
* Future Scopes & Improvements
* Conclusions

# Acknowledgement

I wish to express our specials thanks of gratitude to Head of Department of computer science ***Dr .Divakar Singh Yadav*** as well as to ***Dr .Aditi Sharma*** who helped us throughout this mini project of creating a C/C++ code runner(codeSlate) ,which helped us in doing a lot of research and gaining precious knowledge .

Date : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **codeSlate**

# Intoduction –

Browser based code runner is a web application that can be used to run program written in various languages.

Our project’s scope is limited up to running only C/C++ program.

It is intended primarily for use in computer programming courses.

It is cross platform software made using Node.js for backend , HTML,CSS, vanilla javascript for frontend.

It takes source code and optional program input as input from user , execute at backend and returns the result back to the output window.

# Software Requirements

* Web Browser
* Node.js
* Npm
* GNU mingw compiler

# Hardware Requirements –

* **Minimum Hardware Requirements** 
  + Processor Pentium III
  + Hard disk drive 40 GB
  + RAM 128 MB
  + Cache 512 kb

* Preferred Hardware Requirements
  + Processor Pentium IV
  + Hard disk drive 80 GB
  + RAM 256 MB ➢ Cache 512 kb

**OVERVIEW:**

This web app can used for running C/C++ program . It can be used as tool for developer to type and edit text , mostly programming language file. Our editor are not directly related to programming , infact they are designed to work with framework or language of your choosing. Some well known other options are codechef ide, coding block ide and several other. It’s offline alternatives are sublime text , VS code , vim editor etc.

When we look at these editors we begin to realize that they all come down to same basics set of text editing. Default editor for Microsoft windows is notepad and for Linux OS default editor is vim. These provide copy , cut , paste , undo and redo.

**PROGRAM :**

While the decision between a text editor or an IDE is surprisingly a personal thing, both are very different programs with different features and functionalities. A text editor is exactly what it sounds, a plain simple text editor without the extra bells and whistles of a full-featured IDE. A text editor is simply a computer program and a tool used for editing plain text. An IDE, on the other hand, is a full-fledge software environment that consolidates basic developer tools required to build and test software.

**CODE:**

**Backend:**

let express = require("express")

let path = require("path")

let bodyParser = require("body-parser")

let fs = require("fs")

let app = express()

let proc = require("child\_process")

app.use(bodyParser.urlencoded({ extended: true }));

app.use(express.static(path.join(\_\_dirname, "static")))

app.get("/", (req, res) => {

res.sendFile(path.join(\_\_dirname, "index.html"))

})

app.post("/run", (req, res) => {

let body = req.body

let text = body.input

fs.writeFile("in.txt", body.input, (err) => {

if (err)

res.sendStatus(404)

else {

fs.writeFile("ghf.cpp", body.code, (err) => {

if (err)

res.sendStatus(404)

else {

proc.exec("g++ ghf.cpp && a.exe < in.txt ", (err, resOfCpp) => {

if (err) { res.send(err.toString()); }

else {

//console.log(`<pre>${resOfCpp.toString()}<pre>`)

res.send(resOfCpp.toString())

//res.json({ result: `<pre>${resOfCpp.toString()}<pre>` })

}

})

}

})

}

})

})

app.listen(3000,"0.0.0.0", () => console.log('started'));

**FRONT END: HTML**

<!DOCTYPE html>

<html>

<head>

<title>Run code</title>

<link type="text/css" rel="stylesheet" href="utils.css">

</head>

<body>

<form id="\_program" action="/run" method="post" onsubmit="getRes(this);return false">

<textarea id="\_code" name="code" autofocus></textarea>

<div class="formWrapper">

<fieldset>

<legend>Input</legend>

<textarea class="\_inout" name="input"></textarea>

</fieldset>

<fieldset>

<legend>Output</legend>

<samp class="\_inout \_output" id="output"></samp>

</fieldset>

</div>

<button id="\_submit" type="submit">Run</button>

</form>

</body>

<script src="fn.js"></script>

</html>

**CSS:**

#\_code{

--bgcolor: #3f3f3f;

background-color: var(--bgcolor);

color: white;

caret-color: rgb(0, 255, 0);

font-size: 1.7em;

width: 85vw;

margin: 5px auto;

min-height: 30vh;

height: 75vh;

padding: 1.2em 2.4em;

display: block;

margin-top: 2.4em;

border-radius: 4px;

resize: none;

border-top:40px solid rgb(13, 83, 163);

box-shadow: inset 0 3px 5px 5px black;

}

#\_submit{

outline: 2px solid black;

color: tomato;

padding: 3px;

font-size: 1.3em;

margin: 19px auto;

display: block;

height: 3em;

width: 5em;

}

.formWrapper{

text-align: center;

margin-top: 5vh;

}

form fieldset{

border:2px double black;

width:fit-content;

display: inline-block;

}

form fieldset:first-child{

margin-right:30px;

}

.\_inout{

background-color: lightgray;

color: black;

caret-color: blue;

font-size: 1.2em;

width: 30vw;

margin: auto;

height: 30vh;

padding: 2.4em;

display: block;

margin-top: 2.4em;

border: 4px groove black;

border-radius: 4px;

resize: none;

text-align:left;

}

.\_inout.\_output{

background-color: blanchedalmond;

overflow: auto;

}

**FRONTEND : JS**

let lstCol=["red","blue","black","green","yellow","maroon","orange"];

let docCod=document.getElementById("\_cod");

docCod.addEventListener("keydown",(e)=>{

spaceIt(e)

})

function getRand(){

return Math.floor(Math.random()\*1000000)

}

function colorIt(it){

let flr=Math.floor(Math.random()\*lstCol.length)

it.style.color= lstCol[flr]

console.log(it.style.color)

}

function spaceIt(e){

if(e.key=="Enter")

{

let le=docCod.children

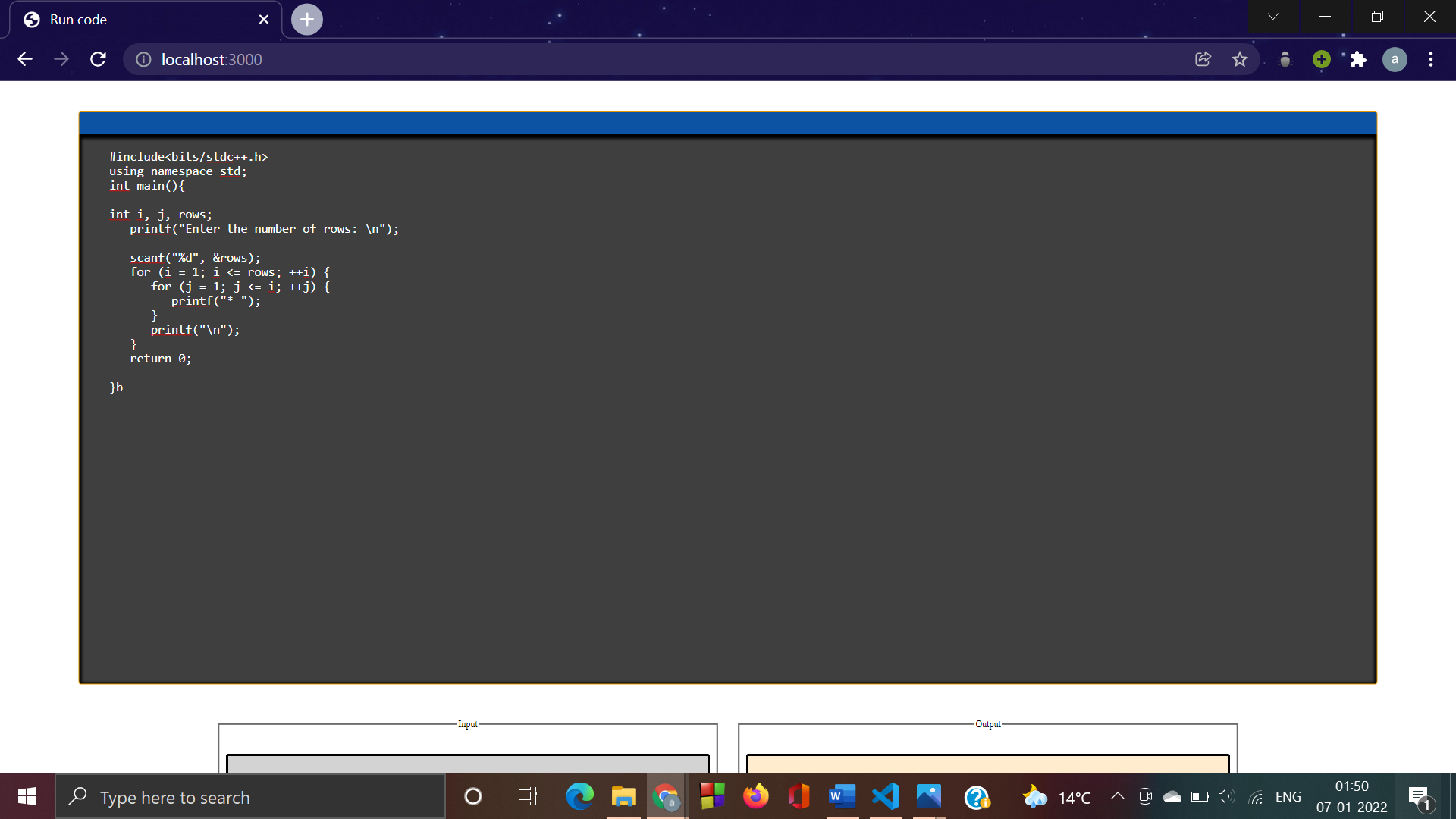
if(le.length>0)

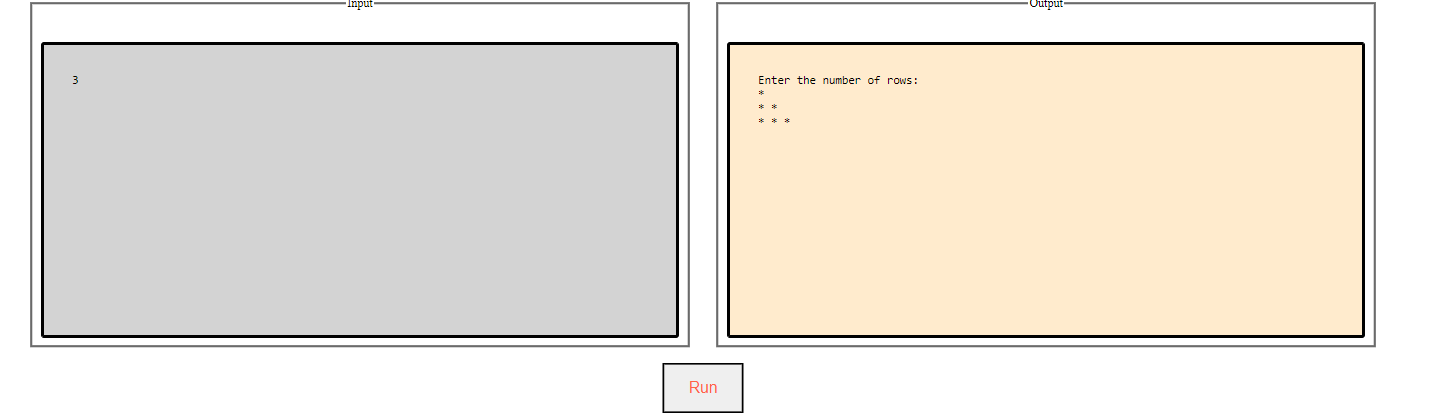
(le[le.length-1]).style.color=lstCol[getRand()%lstCol.length]

}

}

**SNAPSHOTS:**



****

**Future Scopes & Improvements:-**

We will increase the functionality of this web app by adding plugin and extension support . Appropriate Hotkeys will be introduced in the upcoming versions. Color highlighting will be given in the next version.

More customizability will be given to fit the user need. We’ll increase the supported programming language like python , java etc…. in the upcoming versions.

Interactive web terminals will be provided to run interactive programs . Status bar will be given so that user can know what errors he has made. Error echo support will be provided in the near future. Debuggers will be provided to make user feel comfortable while debugging.

We’ll introduce the group coding and live coding playground . We’re thinking of making pastebins so that users can send the code to other users. REPL will be provided to execute python and nodejs program in runtime environment. Error recovery extensions will be given in upcoming versions. Machine learning algorithms will be used to monitor the user experience and according to that code snippet will be recommended.

**Conclusions:**

Thus with the knowledge of node.js ,javascript , HTML ,CSS and shell commands we were able to create this project which can be greatly useful to run C/C++ programs online /offline by competitive programmers, web developers or /and regular software developers. It can also be used as general purpose text editors.

**REFERENCES:**

* <https://www.w3schools.com/tags/default.asp>
* <https://www.w3schools.com/jsref/default.asp>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
* <https://nodejs.org/en/about/resources/>