

PROJECT REPORT ON **CODELINE**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
**INSITUTE OF ENGINNERING &
TECHNOLOGY**
GLA UNIVERSITY, MATHURA

SUBMITTED TO:

Ms. Madhu
(Technical Trainer)

SUBMITTED BY:

Aastha Singh (201500005)
Pranjal Gupta (201500498)
Sarvagya Bansal (201500622)

DECLARATION

I would like to express my special thanks of gratitude to my project guide **Ms. Madhu (Technical Trainer)** mam who gave me the golden opportunity to do this wonderful project on the topic **Codeline**, which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

SUBMITTED BY:

Aastha Singh (201500005)

Pranjal Gupta (201500498)

Sarvagya Bansal (201500622)

CERTIFICATE

Certified that this project report Codeline is the bonafide work of Aastha Singh, Pranjal Gupta and Sarvagya Bansal who carried out the project work under my supervision.

SUPERVISOR

Ms. Madhu

(Technical Trainer)

Table of Content

1. Introduction

1.1 Overview

1.2 Objective

2. Technology Used

2.1 REACT JS

3. System Requirements

3.1 Software Required

3.2 Hardware Required

4. Implementation

4.1 Explanation of Source Code

4.2 Final Code

4.3 Output

5. Conclusion

INTRODUCTION

In the mini project our main objective is to create a system in which that we have the compilers of HTML, CSS and JavaScript alongside on a single setup and the output is obtained at once without the need of creating separate repositories for accessing them, thus saving time and effort at the same time. Front End languages (Html, CSS, JavaScript) communicate requests to Back End languages. Every website has a server, database, and other applications that interact with the Front End through code created by a Back End development. HTML, CSS, and JavaScript are the topmost languages used for Front End development. The structure, design, behavior, and content of everything seen on browser screens when websites, web applications, or mobile apps are opened up, is implemented by Front End developers. Hence, implementation of these front end languages in our project will provide the best, quick and accurate results which will be beneficial to the user at its utmost.

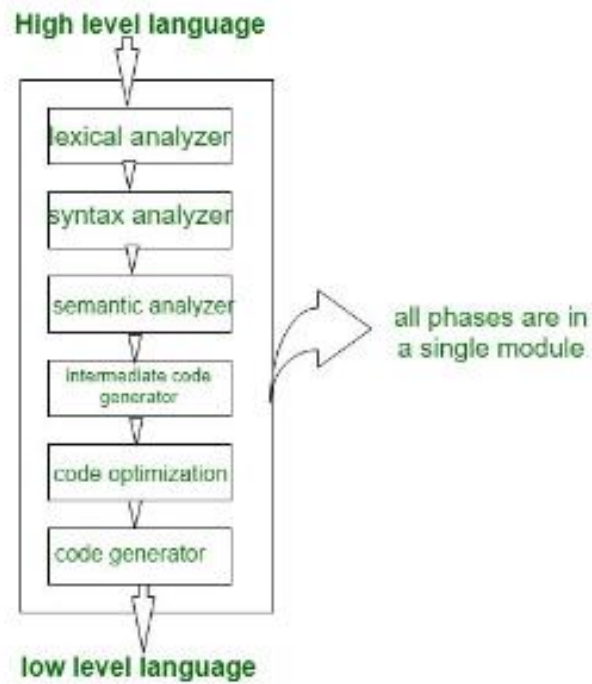
PROJECT- DESCRIPTION

A compiler is a computer program that helps in translating the computer code from one programming language into another language. Basically, it translates the program written in the source language to the machine language. The compiling process contains an essential translation operation and error detection.

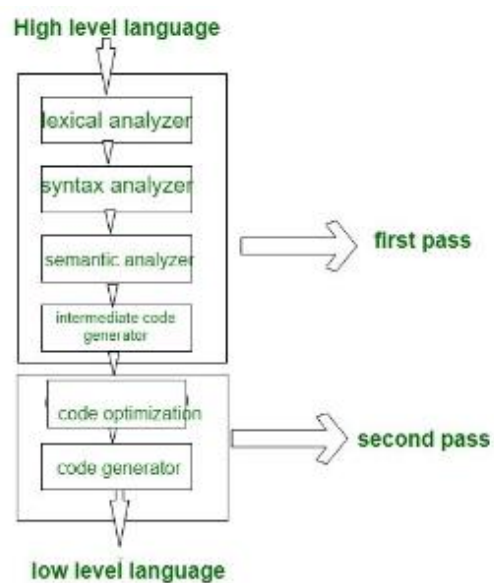
Majorly, there are three types of compilers:

- Single Pass Compilers
- Two Pass Compilers
- Multi-pass Compilers Single Pass Compiler:

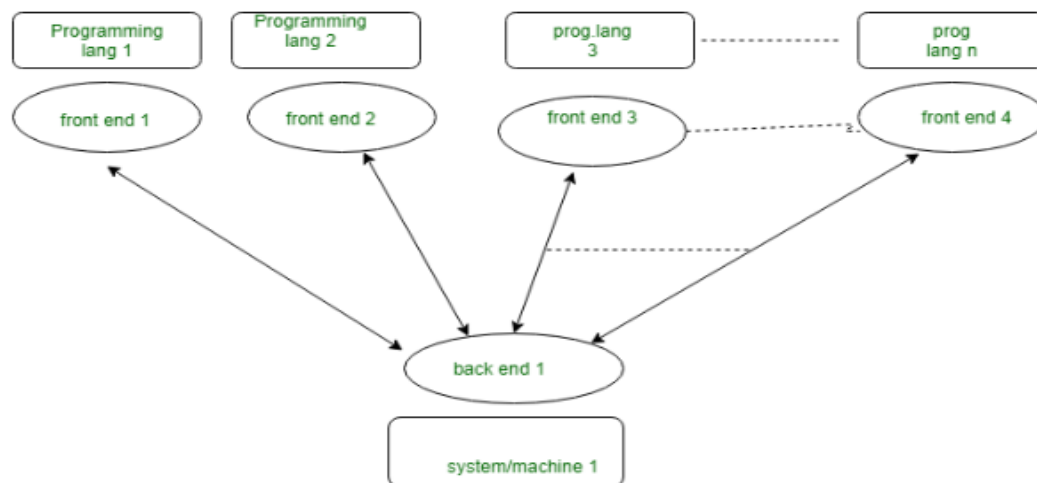
When we merge all the phases of compiler design in a single module, then it is called a single pass compiler. In the case of a single pass compiler, the source code converts into machine code.



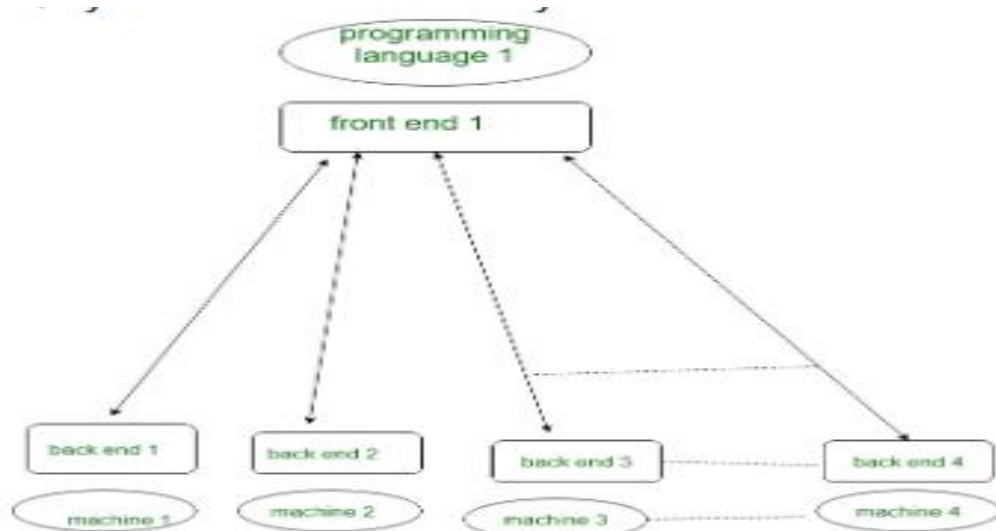
Two Pass Compiler: A processor that runs through the program to be translated twice is considered a two-pass compiler.



Multipass Compiler: A program's source code or syntax tree is processed many times by the multipass compiler. It breaks down a huge programme into numerous smaller programmes and runs them all at the same time. It creates a number of intermediate codes.



All of these multipasses use the previous phase's output as an input. As a result, it necessitates less memory. 'Wide Compiler' is another name for it. To conclude, compilation is the process of translating the code we write, into code that a computer understands. It does this by scanning and parsing your source input, and mapping this input to the target output which your computer can understand.



Further, a compiler can be built and implemented in any language. Keeping that in mind, we have used REACT JS to implement the compilers in our project. In the mini project we have created, a system in which that we have the compilers of JS, HTML and CSS alongside on a single setup and the output is obtained at once without the need of creating separate repositories for accessing them, thus saving time and effort at the same time. HyperText Markup Language (HTML): The most extensively used Frontend Language is HTML, which is a markup language. HyperText Markup Language is the abbreviation for HyperText Markup Language. It's a programming language that is used to make websites and web apps. Cascading Style Sheets: It is a design language that

simplifies making web pages presentable. Selectors are used to pick elements and apply styles to them. JavaScript's: Its primary purpose is to enhance a user's interaction with a web page. In other words, you can utilize this programming language to improve the vibrancy and engagement of your website. JavaScript is also widely used in the development of games and mobile apps. Excellent speed, cross-browser interoperability, and straightforward semantics are just a few of JavaScript's essential qualities, providing a seamless developer experience. Front End languages communicate requests to Back End languages. Every website has a server, database, and other applications that interact with the Front End through code created by a Back End development is implemented by Front End developers. Hence, implementation of these front end languages in our project will provide the best, quick and accurate results which will be beneficial to the user at it

TECHNOLOGY USED

A Compiler can be built and implemented in any language.

Keeping that in mind, We have used REACT JS to implement the compilers in our project. React (also known as React.js or React-JS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

SYSTEM REQUIREMENTS

Software Requirement-

To build website –

- JSDK1.5
- React-JS
- Oracle 8i

Hardware Requirement –

- x86_64 CPU architecture; 2nd generation Intel Core or newer
- 8 GB RAM or more
- 8 GB of available disk space minimum

IMPLEMENTATION

Final Code –

```
import React, { useState, useEffect } from 'react';
import Editor from './Editor'
import useLocalStorage from '../hooks/useLocalStorage'
```

```
function App() {
  const [html, setHtml] = useLocalStorage('html', '')
  const [css, setCss] = useLocalStorage('css', '')
  const [js, setJs] = useLocalStorage('js', '')
  const [srcDoc, setSrcDoc] = useState('')
```

```
  useEffect(() => {
    const timeout = setTimeout(() => {
      setSrcDoc(`
        <html>
          <body>${html}</body>
          <style>${css}</style>
          <script>${js}</script>
        </html>
      `)
    }, 250)
```

```
    return () => clearTimeout(timeout)
  }, [html, css, js])
```

```
  return (
    <div className="pane top-pane">
      <Editor
        language="xml"
        displayName="HTML"
```

```

        value={html}
        onChange={setHtml}
      />
      <Editor
        language="css"
        displayName="CSS"
        value={css}
        onChange={setCss}
      />
      <Editor
        language="javascript"
        displayName="JS"
        value={js}
        onChange={setJs}
      />
    </div>
    <div className="pane">
      <iframe
        srcDoc={srcDoc}
        title="output"
        sandbox="allow-scripts"
        frameBorder="0"
        width="100%"
        height="100%"
      />
    </div>
  </>
)
}

export default App;
import React, { useState } from 'react'
import 'codemirror/lib/codemirror.css'
import 'codemirror/theme/material.css'
import 'codemirror/mode/xml/xml'

```

```
import 'codemirror/mode/javascript/javascript'
import 'codemirror/mode/css/css'
import { Controlled as ControlledEditor } from 'react-codemirror2'
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome'
import { faCompressAlt, faExpandAlt } from '@fortawesome/free-solid-svg-icons'
```

```
export default function Editor(props) {
```

```
  const {
    language,
    displayName,
    value,
    onChange
  } = props
```

```
  const [open, setOpen] = useState(true)
```

```
  function handleChange(editor, data, value) {
    onChange(value)
  }
```

```
  return (
```

```
    <div className={`editor-container ${open ? '' : 'collapsed'} `>
```

```
      <div className="editor-title">
```

```
        {displayName}
```

```
        <button
```

```
          type="button"
```

```
          className="expand-collapse-btn"
```

```
          onClick={() => setOpen(prevOpen => !prevOpen)}
```

```
        >
```

```
          <FontAwesomeIcon icon={open ? faCompressAlt : faExpandAlt}
```

```
/>
```

```
        </button>
```

```
      </div>
```

```
      <ControlledEditor
```

```
        onBeforeChange={handleChange}
```

```

    value={value}
    className="code-mirror-wrapper"
    options={{
      lineWrapping: true,
      lint: true,
      mode: language,
      theme: 'material',
      lineNumbers: true
    }}
  />
</div>
)
}
import { useEffect, useState } from 'react'

const PREFIX = 'codeline'

export default function useLocalStorage(key, initialValue) {
  const prefixedKey = PREFIX + key

  const [value, setValue] = useState(() => {
    const jsonValue = localStorage.getItem(prefixedKey)
    if (jsonValue !== null) return JSON.parse(jsonValue)

    if (typeof initialValue === 'function') {
      return initialValue()
    } else {
      return initialValue
    }
  })

```



```
})
```

```
useEffect(() => {  
  localStorage.setItem(prefixedKey, JSON.stringify(value))  
}, [prefixedKey, value])
```

```
return [value, setValue]
```

```
}
```

```
{
```

```
  "name": "codeline",
```

```
  "version": "0.1.0",
```

```
  "private": true,
```

```
  "dependencies": {
```

```
    "@fortawesome/fontawesome-svg-core": "^1.2.30",
```

```
    "@fortawesome/free-solid-svg-icons": "^5.14.0",
```

```
    "@fortawesome/react-fontawesome": "^0.1.11",
```

```
    "@testing-library/jest-dom": "^4.2.4",
```

```
    "@testing-library/react": "^9.5.0",
```

```
    "@testing-library/user-event": "^7.2.1",
```

```
    "codemirror": "^5.57.0",
```

```
    "react": "^16.13.1",
```

```
    "react-codemirror2": "^7.2.1",
```

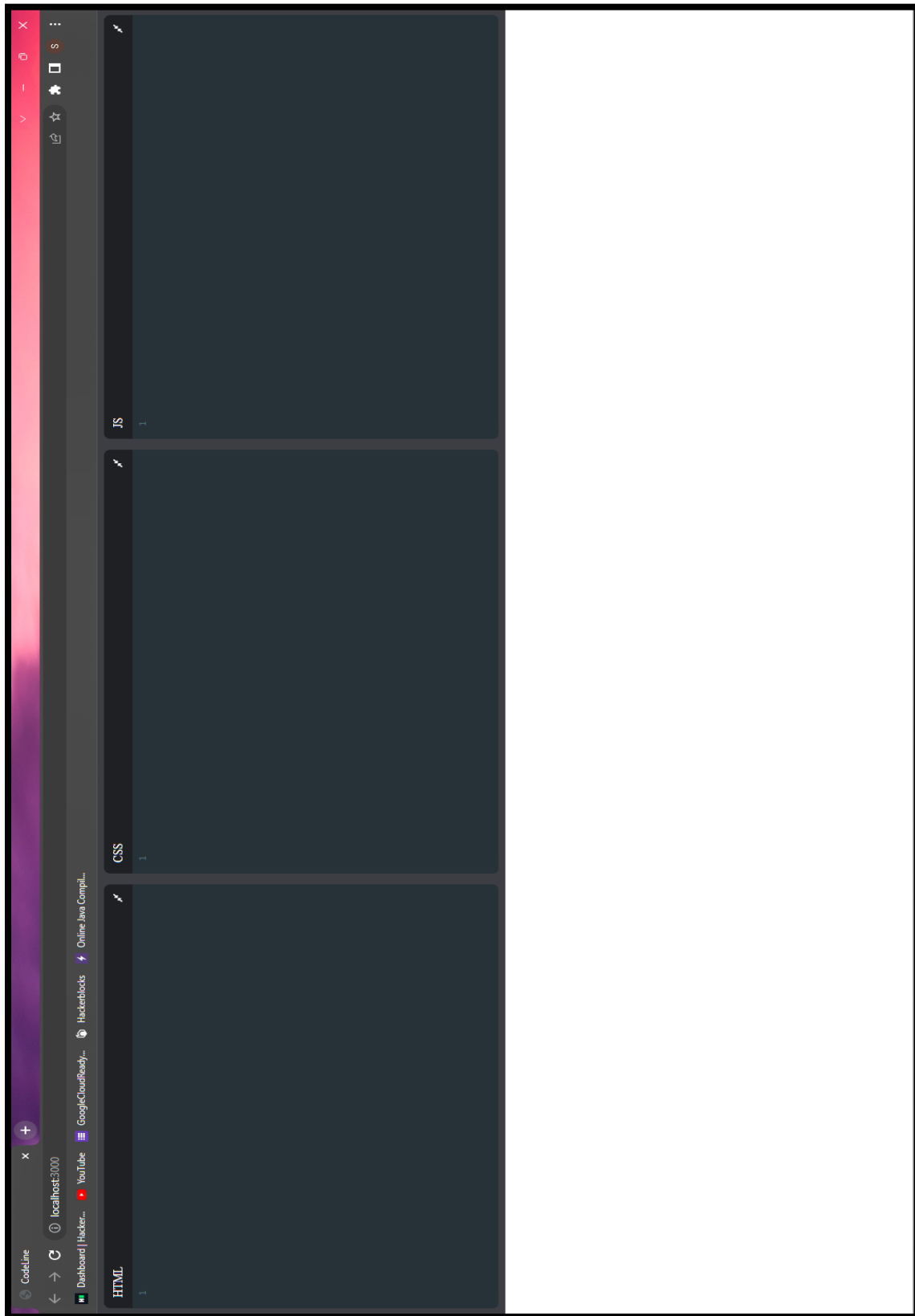
```
    "react-dom": "^16.13.1",
```

```
    "react-scripts": "^5.0.1"
```

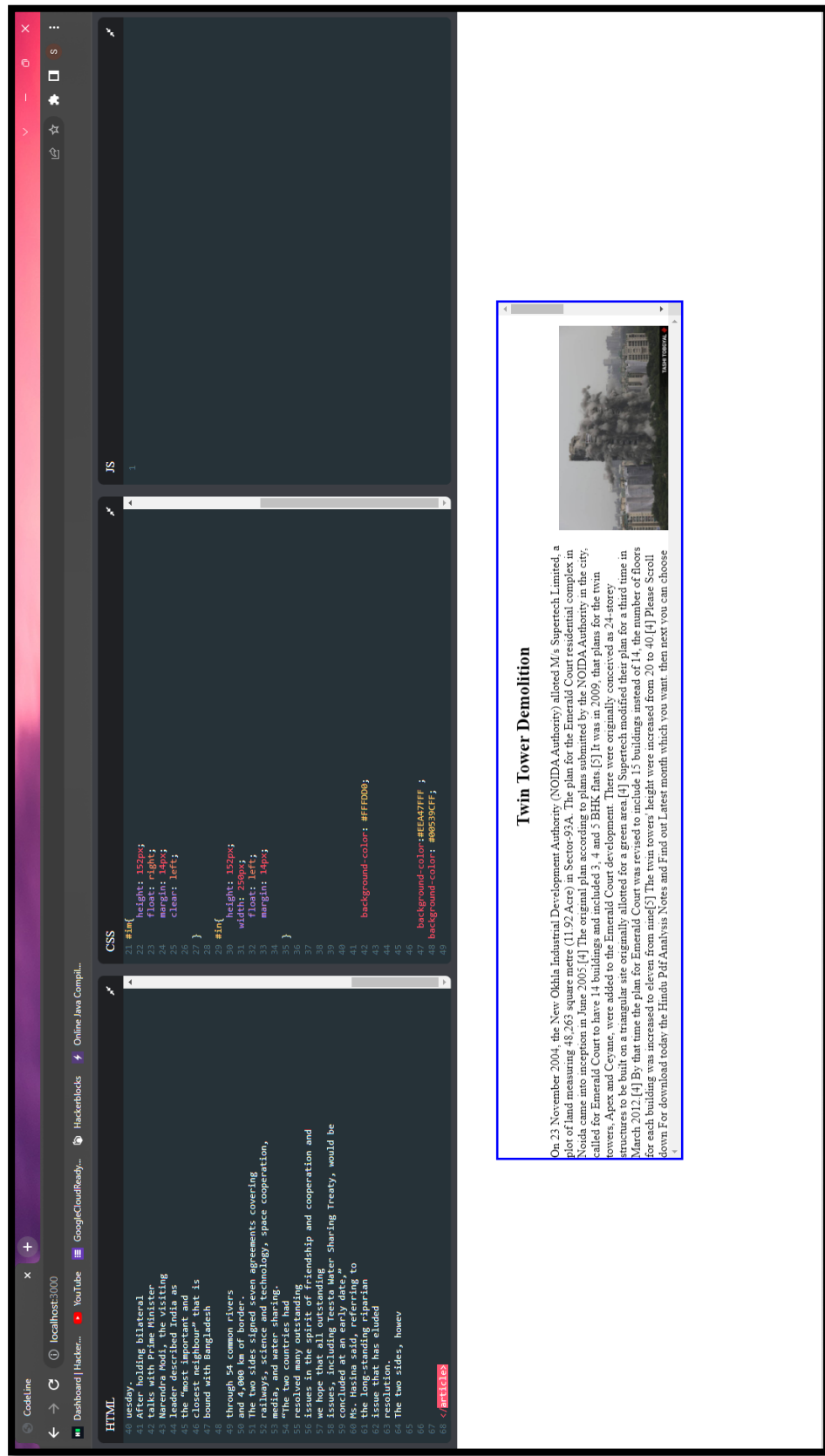
```
  },
```

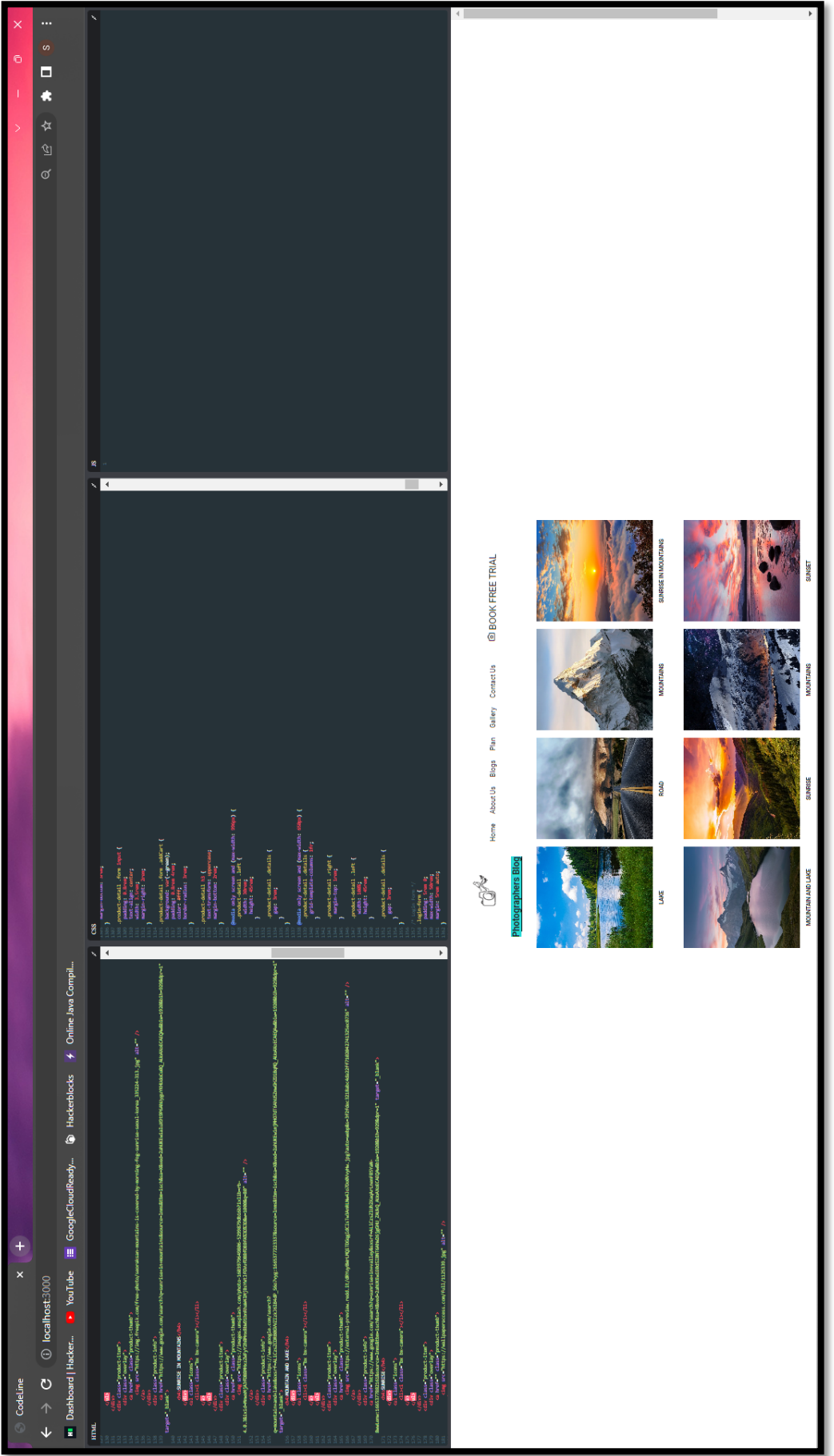
```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject"  
},  
"eslintConfig": {  
  "extends": "react-app"  
},  
"browserslist": {  
  "production": [  
    ">0.2% ",  
    "not dead",  
    "not op_mini all"  
  ],  
  "development": [  
    "last 1 chrome version",  
    "last 1 firefox version",  
    "last 1 safari version"  
  ]  
}  
}
```

Compiler Design



Outputs





CONCLUSION

With web-based compiler system, hassle free web development can be done. There is efficiency in paper procurement for charging the product. The data of all the products is stored in a centralized manner and the costs can be controlled and monitored by the operational manager and owner thus avoiding the over-budgeting. Data storage which is already computerized will ease the process for companies and the users for performing pre-processing, recognizing the buying patterns and maintaining the integrity of the data and use this information to a personal benefit. Our application is user-friendly, open source and is Free to use. It positively impacts the environmental situation by using fewer products a greater number of times. Hiring products provides a simple way of collecting useful information to measure this service. Concentrating on customer satisfaction and the four dimensions, “Reliability”, “Responsiveness”, “Tangibles” and “Quality” helps us to serve the users in a better manner and thus give us a competitive edge over the others.

Github Link

Codeline

<https://sarvagyan.github.io/Codeline/build/>

References

❖ Books:

- HTML
 - HTML - by Jon Duckett
 - Head First HTML and CSS by Elizabeth Robson and Eric Freeman
- CSS
 - CSS - by Eric A. Meyer
 - CSSbyJonDucket
- JAVASCRIPT
 - You Don't Know JS by Kyle Simpson
 - A Smarter Way to Learn JavaScript by Mark Myers

❖ Websites:

- www.geeksforgeeks.org
- www.reactjs.org
- www.projectdeveloper.com
- www.javatpoint.com
- www.youtube.com

THANK YOU