

PROJECT REPORT (BTCS 703-18)

On
Algorithms Api

Submitted in partial fulfillment of the
Requirements for the award of

Degree of Bachelor of Technology in Computer Science & Engineering



Submitted To:
Mr. Rajeev Sharma
Assistant professor

Submitted By:
Sachin Kumar
1902219

Department of Computer Science & Technology
CHANDIGARH ENGINEERING COLLEGE, LANDRAN

CERTIFICATE

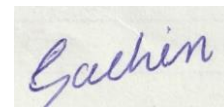
This is to certify that **Mr. Sachin** has completed the Project-II during the period from **July 2022** to **November 2022** in our Organization / Industry as a Partial Fulfillment of Degree of Bachelor of Technology in Computer Science & Engineering.

(Signature of Project supervisor)

Date: 30-11-2022

DECLARATION

I hereby declare that the Project Report entitled ("**Algorithms Api**") is an authentic record of my own work as requirements of 7th semester academic during the period from August to December for the award of degree of B. Tech. Computer Science & Engineering, **Chandigarh Engineering College** under the guidance of **Ms. Monika Takhi.**



(Sachin)
(1902219)

Date: 30-11-2022

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

Signatures

Examined by:

1.

2.

3.

4.

**Head of Department
(Signature and Seal)**

ACKNOWLEDGMENT

I take this opportunity to express my sincere gratitude to the Director- Principal Dr. Rajdeep Singh Chandigarh Engineering College, Landran for providing this opportunity to carry out the present work.

I am highly grateful to the **Dr. Sukhpreet Kaur** HOD CSE, Chandigarh Engineering College, Landran (Mohali). I would like to express my gratitude to other faculty members of Computer Science & Engineering department of CEC, Landran for providing academic inputs, guidance & Encouragement throughout the training period. The help rendered by **Mr. Rajeev Sharma**; Supervisor for Experimentation is greatly acknowledged. Finally, I express my indebtedness to all who have directly or indirectly contributed to the successful completion of my software training.

TABLE OFCONTENT

AKNOWLEDGMENT.....	I
TABLE OF CONTENT.....	II
LIST OF FIGURES.....	IV
LIST OF TABLES.....	V

CHAPTER 1: INTRODUCTION

1.1 ABOUT PROJECT	1
1.2 PROBLEM STATEMENT	3
1.3 OBJECTIVES	3
1.4 SCOPE OF THE PROJECT.....	3

CHAPTER 2: LITERATURE SURVEY

2.1 TECHNOLOGY.....	4
2.2 WEB APPLICATION.....	4
2.3 EXPRESS.....	5

CHAPTER 3: REQUIREMENTANALYSIS

3.1 FUNTIONAL REQUIREMENTS.....	10
3.2 NON-FUNCTIONALREQUIREMENTS.....	10
3.3 DOMAINAND UI REQUIREMENTS.....	10
3.4 HARDWAREAND SOFTWARE REQUIREMENTS.....	11

CHAPTER 4: DESIGN

4.1 DESIGN GOALS.....14

4.2 OVERALL SYSTEMARCHITECTURE.....14

4.3 FLOWCHART.....15

4.4 DATAFLOW CHART.....16

4.5 USE-CASE DIAGRAM.....17

4.6 SEQUENCE DIAGRAM.....18

4.7 STATE DIAGRAM.....18

CHAPTER 5: IMPLEMENTATION

5.1 JSON.....19

5.2 DATAREPRESENTATION.....24

5.3SEGMENTATION.....25

5.4 WEB ARCHITECTURE.....31

5.5 BUILDING MODEL.....26

CONCLUSION.....60

REFERENCES61

TABLE OF FIGURES

Figure 1.1.....	6
Figure 1.4.....	14
Figure 2.3.....	12
Figure 2.4.....	14
Figure 2.5.....	15
Figure 3.1.....	18
Figure 3.2.....	19
Figure 4.3.....	24
Figure 4.4.....	25
Figure 4.5.....	26
Figure 4.6.....	27
Figure 4.7.....	27

LIST OF TABLES

Figure 1.2	11
Figure 4.2... ..	13
Figure 5.3... ..	20
Figure 6.1.....	28
Figure 6.2.....	29
Figure 6.3.....	30
Figure 6.4.....	37

CHAPTER 1: INTRODUCTION

1.1 ABOUT PROJECT

An application programming interface (API) is a way for two or more computer programs to communicate with each other. It is a type of software interface, offering a service to other pieces of software.^[1] A document or standard that describes how to build or use such a connection or interface is called an *API specification*. A computer system that meets this standard is said to *implement* or *expose* an API. The term API may refer either to the specification or to the implementation.

In contrast to a user interface, which connects a computer to a person, an application programming interface connects computers or pieces of software to each other. It is not intended to be used directly by a person (the end user) other than a computer programmer who is incorporating it into the software. An API is often made up of different parts which act as tools or services that are available to the programmer. A program or a programmer that uses one of these parts is said to *call* that portion of the API. The calls that make up the API are also known as subroutines, methods, requests, or endpoints. An API specification *defines* these calls, meaning that it explains how to use or implement them.



Fig. 1.1

The term API is often used to refer to web APIs,^[2] which allow communication between computers that are joined by the internet. There are also APIs for programming languages, software libraries, computer operating systems, and computer hardware. APIs originated in the 1940s, though the term did not emerge until the 1960s and 1970s. Recent developments in APIs have led to the rise in popularity of microservices, which are loosely coupled services accessed through public APIs.

1.2 PROBLEM STATEMENT

Web development is changing at lightning speed with evolving technology. Developers have to be updated with all latest web technologies that can be implemented. To maintain and improve web development, it is necessary to select the right tool and it can be a difficult decision since there are many options available.

We have detailed the latest web development technologies being used by developers in 2022.

1. Progressive Web App

Progressive web app has characteristics of both website and a locally installed program. It has new capabilities of modern apps. It offers advantages of a traditional local app in a setup used for different device types. It is painless to manage and maintain.

PWAs include modern features of patterns, APIs, and others. These app work offline, are installable, is easy to synchronize, and can send push notifications, etc. Companies that migrate to PWA see a 20 percent to 250 percent boost in traffic, according to Smashing Ideas.

Examples –

Starbucks built a PWA app, which delivers experience of native app. It runs in offline mode, allows their customers to browse the menu, customize their orders, and add items to their cart. On online app, it can show location-specific pricing and food and drinks order.

Stat

According to Gartner Research, Progressive Web Apps replaced almost 50% of all consumer-facing native apps by 2020.

Advantages

- Good connectivity and works offline
- Easy to update
- Cost efficient
- Low average bounce rate

2. Yii

Yii is an open-source web application development framework. It uses PHP5 and has good tools for debugging and app testing. It is very simple and easy for developers to use.

Example - Skill Share is an American online learning platform that sponsors the Yii framework developers and actively uses it in their projects.

Advantages

- Easy to install
- Easy data migration
- Short time for development
- Simplified security

3. Meteor JS

Meteor JS uses Node.js and it helps developers to create web applications used on various platforms. It is an open-source JavaScript web framework it has low webpage loading time.

Example – Pathables website runs virtual, hybrid, and in-person conferences and events. People use their end-to-end solutions to learn and connect to business events.

Advantages

- Community support
- Based on single language – JavaScript
- Easy to learn
- Real-time web application

4. Django

Django is popular frameworks based on Python and follows MVC architecture. It makes the app development process easier and provides multiple tools like ORM, Models, Django admin, templates, etc.

Example - Instagram is an example of a website-built Django. It contains huge data and user interactions (likes, shares, etc.). With Django, simplifies data management and ready-to-implement instruments with simpler UI and UX.

Advantages

- Rapid development
- High security
- Community support
- Scalable
- Object oriented
- Django CMS pricing starts at \$0.01 per feature
- There is a free version
- Django CMS offers a free trial

5. Motion UI

Customer want a mobile-friendly experience with interactive use. Motion are used for pre-made effects as CSS classes. These motions are used for transition effects for better animation to slide, hinge, fade, scale, and spin.

Motion UI is used for fun elements like Zoom Out, Zoom In, Slide, and parallax animation.

It adds liveliness to the site and the app.

Advantages

- Increase user engagement
- Intuitive and interactive UI
- High retention rate

6. Blockchain

Blockchain technology has revolutionized the development industry. Many companies in various industries are searching for new and advanced ways to use blockchain technology.

A qualified blockchain developer understands the blockchain architecture, underlying technologies and what it takes to implement the individual blockchain components into a unified system. The developer knows how to fit blockchain structure with cryptography and data structures. They know how to build smart contracts and decentralized applications.

Advantages

- Transparency
- Security
- Efficiency
- Cost reduction

7. Laravel

Laravel is a popular backend web framework. Laravel is a free, open-source PHP web framework, intended for building state-of-the-art web applications following the model–view–controller (MVC) architectural pattern.

Example –

Barchart is a forex, stock, cryptocurrency, futures research platform. It provides traders and investors with a quick overview of the markets.

They have used larval to build a website that has great tools for technical research for domestic as well as global markets.

Know more about why Larvel is the best PHP development framework in 2022.

Advantages

- Seamless user authentication
- Cross-platform application
- Fast caching
- Secure truncations
- Automated testing

Table 1.2

Larvel Developers by Country	Average Hourly Rate
Ukraine	\$25.30
United States	\$47.37
India	\$15.74
Germany	\$34.62
United Kingdom	\$36.04

8. Serverless Architecture

Developers always need a technology that can decrease the amount of data loss, system overloading, and development costs. This unending search led to invention of serverless technology. Serverless architecture uses cloud technology and allows users to run code for nearly any form of application or backend service with almost negligible management.

Google, Amazon, and Microsoft are examples of corporations using serverless architecture.

Advantages

- Low cost
- Better UX
- Scalability
- flexibility

9. API-first Development

APIs are high priority in API-first development. As APIs facilitate connections, which developers need to establish with software and devices. API-first development is user-focused and it reduces development time and costs.

Example - Netflix uses API architecture for programming interfaces and to support public and private internal requests.

Advantages

- Fast development
- Low failure risks
- Low development cost

10. Data Security

Data security means protecting your digital data and preventing cyberattacks. Cyberattacks can cause loss to business and reputation. Security is one of the main concerns for companies who want to improve their web development technologies in 2022. Over the years AI will become more helpful in data security.

Advantages

- Data Protection
- Prevent cyber threats
- Prevent hackers

1.3 PROBLEM STATEMENT

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface. Web services are Web apps by definition and many, although not all, websites contain Web apps. According to Web.App Storm editor Jarel Remick, any website component that performs some function for the user qualifies as a Web app.

Web applications can be designed for a wide variety of uses and can be used by anyone; from an organization to an individual for numerous reasons. Commonly used Web applications can include webmail, online calculators, or e-commerce shops. Some Web apps can be only accessed by a specific browser; however, most are available no matter the browser. How Web applications work.

Web applications do not need to be downloaded since they are accessed through a network. Users can access a Web application through a web browser such as Google Chrome, Mozilla Firefox or Safari.

For a web app to operate, it needs a Web server, application server, and a database. Web servers manage the requests that come from a client, while the application server completes the requested task. A database can be used to store any needed information.

Web applications typically have short development cycles and can be made with small development teams. Most Web apps are written in JavaScript, HTML5, or Cascading Style Sheets (CSS). Client-side programming typically utilizes these languages, which help build an applications front-end. Server-side programming is done to create the scripts a Web app will use. Languages such as Python, Java, and Ruby are commonly used in server-side programming.

Benefits

Web applications have many different uses, and with those uses, comes many potential benefits. Some common benefits of Web apps include:

Allowing multiple users access to the same version of an application.

- Web apps don't need to be installed.
- Can be accessed through multiple browsers.

Web Application vs. other application types

Within the mobile computing sector, Web apps are sometimes contrasted with native apps, which are applications that are developed specifically for a particular platform or device and installed on that device. However, the two are not mutually exclusive.

Native applications are applications typically downloaded and made specifically for the type of device it is downloaded on. Native apps can

commonly make use of the device-specific hardware, such as a GPS or camera on a mobile native app.

Programs that combine the two approaches are sometimes referred to as hybrid applications. Hybrid apps work similar to a Web app but are installed to the device as a native app would be. Hybrid apps can also take advantage of device-specific resources by using internal APIs. Downloaded native apps can sometimes operate offline; however, hybrid apps don't have this functionality. A hybrid app will typically share similar navigation elements are a Web app since they are based on Web apps.

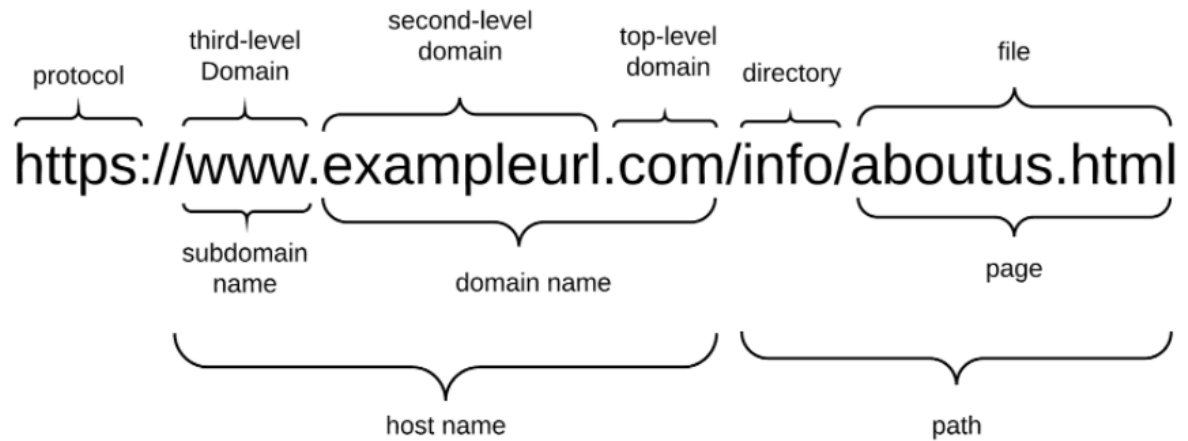
A problem statement is an actionable summary of your user, their goals, and what you need to solve to meet those goals. It sets the course for your project's user experience; however, it shouldn't include anything about the design elements or user functionality you might employ to solve the problem. Instead, it should provide insight into why you're designing the user experience to begin with.

In other words, a user doesn't need to be able to use two-factor authentication to get into a banking app, they need to be able to access their banking information securely. The problem statement should lay out the user's need (secure access) but leave out any possible solutions (two-factor authentication), which shouldn't be considered until further along in the design process.

1.4 OBJECTIVES

To Help People to get algorithms on demand to get algorithm's in c++ language to use that in their web application

Figure 1.4



Like:

Request: <https://AlgoApi.com?algo=mergesort>

Response: {

algo: “merge sort”,

sol: “`#include <iostream>`

`#include <vector>`

`using namespace std;`

`// Function to merge two sorted lists`

`vector<int> merge(vector<int> left, vector<int> right)`

`{`

`// Create a new list to store the merged elements`

`vector<int> merged;`

`// Set indices for the left and right lists`

`int i = 0;`

`int j = 0;`

`// Merge the lists`

`while (i < left.size() && j < right.size())`

`{`

`if (left[i] < right[j])`

`{`

`merged.push_back(left[i]);`

`i++;`

`}`

`else`

`{`

`merged.push_back(right[j]);`

`j++;`

```

    }
}

// Add any remaining elements from the left list
while (i < left.size())
{
    merged.push_back(left[i]);
    i++;
}

// Add any remaining elements from the right list
while (j < right.size())
{
    merged.push_back(right[j]);
    j++;
}”
}

```

1.5 SCOPE OF THE PROJECT

Can be used to learn algorithm’s by fresher’s and need not to search algorithms. A website development scope of work is a document that web developers send to clients to provide details on a specific project and its requirements.

Once signed, it’s a legally binding document that helps both parties avoid conflict, minimize misunderstandings, and stay on the same page.

A clear website development scope of work template lays the foundation of a new project by outlining important deadlines, deliverables, and key expectations, all you need to do is tweak it per client.

It helps the client clarify expectations, visions, and responsibilities at the beginning of a project.

Communication throughout any web design project is *key*—keep touching base with clients to gather requirements for the SOW and validate their web design and development expectations. This will help you avoid miscommunication, hiccups, or misunderstandings.

CHAPTER 2: LITERATURE SURVEY

2.1 TECHNOLOGY

Node.js (Node) is a cross platform runtime environment originally developed in 2009 by Ryan Dahl for developing server-side applications. It can be regarded as server-side JavaScript. It was created to address the issues platforms can have with the performance in network communication time dedicating excessive time processing web requests and responses. “Node.js is a platform built on Chrome’s JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices [1].” Node has become popular as it makes creating high performance, real-time web applications easy. Node allows JavaScript to be used end to end, both on the server and on the client. JavaScript has originally run only in the web browser, but the considerable demand has brought it to the server-side. JavaScript has developed very much and has exceled to dominate server-side scripting. We need to analyze the security issues in Node applications because of its use with JavaScript, which has security liabilities. Node is event-based rather than thread-based. Node uses an event loop within a single thread instead of multiple threads, and is able to scale to millions of concurrent connections. In Node, a single thread can accomplish a high concurrency. Every I/O operation in Node is carried out in an asynchronous fashion, meaning that the server can continue to process incoming requests while the I/O operation is taking place [2]. Because 2 Node is also asynchronous, AJAX (See Chapter 3) could be mistakenly considered equivalent to Node, though they are significantly different. Objectives of our Study JavaScript was created shortly after the World Wide Web (WWW) came into existence. JavaScript has played an integral role in adding interaction to the user interface of web applications and websites until the recent release of HTML5 (Hyper Text Markup Language) and modern JavaScript frameworks.

2.2 WEB APPLICATION

JavaScript is also an integral part of AJAX which was introduced in late 1990’s with the advent of Web 2.0 to add real-time like interactivity in the webpages. Despite all of this progress, JavaScript has been considered as the scripting language for client-side programming (that runs only from the

browser). However, this approach has changed with the development of server-side JavaScript (among which Node is considered prominent). Node has not just strengthened server-side JavaScript, but also has been competitive with other popular server-side scripting languages with respect to performance and scalability. In this paper, we will describe the advantageous features of Node. The underlying features of Node: single-threaded, event-driven I/O, and asynchronous programming are discussed with sufficient examples to give better insight into the working architecture of Node that led to Node's success. In this paper, we also distinguish Node from JavaScript, which is the backbone on top of which Node has been developed. The same is true of AJAX, which is often time confused with Node. We perform an analysis of Node's performance with real-time data by implementing two well-known applications (Fibonacci number calculation and reading large text file).

3 Simple HTTP Server with Node.js

One of the common uses of Node is to build servers. Node can be used to create different types of servers [7]. A simple HTTP (Hyper Text Transfer Protocol) web server that responds "Hello Node!" to every request it receives can be created with very few lines of code. Type the following code in the text editor, save it as `hello_server.js` and execute it by typing `node hello_server.js` from the command prompt. Then, visit this URL: `http://localhost:8000` which should print the "Hello Node!" message in the browser.

Figure 1 Simple HTTP Server in Node.js

Description of Figure 1 is as follows [6]:

0. The 'use strict' string is a directive that enables strict mode, which is a restricted subset of the language that fixes a few important language deficiencies and provides stronger error checking and increased security. For example, strict mode makes it impossible to accidentally create global variables.
1. Imports the `http` module and assign it to the `http` object.
2. Defines the port number from which the server will be accepting connections.
3. Create the web server by calling the factory method from the `HTTP` module and send it a callback function. This anonymous function takes `req` and `res` (HTTP request and response) objects as parameters. Every time a client makes a request, this function will get called.
4. Displays to standard output the request information (method, requested resource, and request headers) followed by a new line.

2.3 EXPRESS

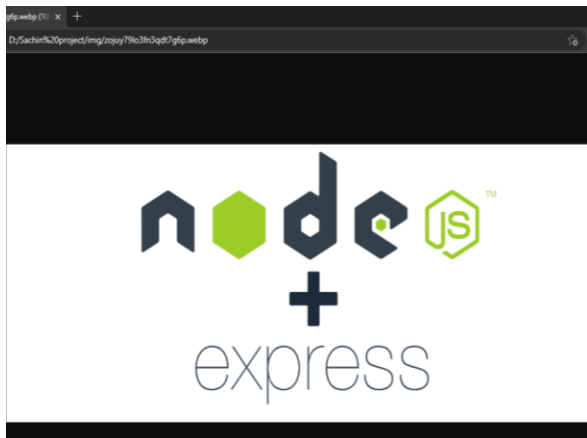
The following program will respond to HTTP GET requests with the text 'Hi, your request has been received', and listen to the port the program is running on; Port 2000, in this case.

```
// Import the Express library.
const express = require('express');

// Initializing the app.
const app = express();

// Getting the path request and sending the response with text
app.get('/', (req,res) => {
  res.send('Hi, your request has been received');
});

// Listen on port 2000
app.listen(2000, () => {
  console.log('listening at http://localhost:2000');
});
```



CHAPTER 3: REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

All applications start with an idea that's a potential solution to an identified problem. It's no different for a web application. Once the idea for a web application has been identified, it's important to formally define the stakeholder requirements of the application. The process of generating the requirements (i.e., the needs, tasks, and goals of the application) is known as requirements gathering.

Say an online collaboration application called Docetism allows users to create documents, logs, videos, and discussions and collaborate on them. Docetism also has an approval workflow for content publishing, as well as an admin function that can generate participation and usage reports.

The process of requirements gathering is the first and crucial phase of the software development life cycle (SDLC) process as it sets the stage for the subsequent phases. It is at this phase that the assessment of the business goals or the objective that dotcom needs to meet is fleshed out.

In software engineering, requirements gathering identifies the functional and non-functional requirements of the web application.

Figure 3.1



Mobile Application vs. Web Page

One question you will need to ask right away is: Are we building a mobile application, a web page, or both?

For a majority of applications, users are going to want a mobile-friendly option. As such, we should set our frame of mind for a **mobile first** strategy. This means that our design starts with the smallest screen and adjusting so that the application can run on any screen size. When it comes to the functional requirements listed below, you will want to ensure those components are front and centre (or at least easily available and visible in a mobile application or web application).

When it comes to other requirements, you will want to ensure they are organized in such a way that they do not interfere with the core site. Add a small menu called 'extras' for example

Having understood that we should look to mobile first, we can then begin the requirements gathering for our web application.

Functional Requirements

Functional requirements are needs related to the technical functionality of the system. Functional requirements state how the users will interact with the application, so the application must be able to comply and be testable. For the online collaboration application, dotcom, the functional requirements may include descriptions of:

- Content to be created and published in the system (i.e., documents, blogs, videos, etc.)
- Operations performed by each screen
- Content approval work-flow process
- Participation and usage reports

3.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements specify criteria that can be used to judge the operation of a system in particular. These are conditions rather than specific behaviors. While functional requirements define what a system is supposed to do, non-functional requirements define how a system is supposed to be. Non-functional requirements are often called "quality attributes" of a system.

Examples for dotcom may include:

- Performance - how much time each page should take to load
- Scalability - will the system be able to handle large volume of users that keeps increasing?
- Capacity - how much storage will be needed?
- Availability - availability and downtime of the application
- Security - this includes security of the content and encryption

Methods of Requirements Gathering

Requirements gathering can begin as a brief description of what the online collaboration application must have as a minimum and what users will be able to do on it. As those requirements are fleshed out, they need to be defined, described, and documented.

While there is no perfect way to identify and gather requirements, some commonly used methods include:

3.3 DOMAIN AND UI REQUIREMENTS

Hypertext Markup Language (HTML)

Hypertext Markup Language -- known as HTML -- is the lingua franca for publishing on the Web. Following the success of W3C's HTML 3.2 and HTML 4.0, the Consortium is designing the next generation of the markup language. The new HTML, called XHTML, is re-cast in XML and is being designed so that it can be used in combination with other XML applications.

The HTML Activity also includes work on the next generation of forms. This work started as a subgroup of the HTML working group but has now been spun off into an independent working group (XForms). The key idea is to separate the user interface and presentation from the data model and logic, allowinging the same form to be used on a wide variety of devices such as voice browsers, handhelds, desktops and even paper. XForms brings the benefits of XML to Web forms, transferring

form data as XML. XForms aims to reduce the need for scripting, and to make it easier to achieve the desired layout of form fields without having to resort to using nested tables etc.

Style Sheets

Style sheets offer precise control over the presentation of Web pages. Not only can Web designers specify the visual effects they want, but also aural style sheets give control over voice, pitch and other aspects of how the text will sound when rendered into speech. After the publication of two recommendations, CSS1 and CSS2, W3C continues to evolve the Cascading Style Sheets (CSS) language to provide even richer stylistic control, while emphasizing the importance of the thorough implementation of CSS on browsers.

For more complex publishing tasks, such as automatically producing a table of contents, and for converting documents written in XML into HTML for publication, W3C is developing the Extensible Style Sheets Language (XSL), which builds upon experience in CSS and DSSSL.

Synchronized Multimedia

W3C's Synchronized Multimedia Activity focusses on the design of a language for scheduling multimedia presentations where audio, video, text and graphics are combined in real-time. The language, the Synchronized Multimedia Integration Language (SMIL) is written as an XML application and is currently a W3C Recommendation. Simply put, it enables authors to specify what should be presented when.

Math

Communicating mathematical and other technical notation is a challenging and important task. The demand is high for effective means of electronic scientific communication. To address the needs of the scientific community, W3C has developed a core specification for embedding mathematical expressions in HTML and XML documents. In April 1998, W3C published a Recommendation entitled Mathematical Markup Language, or MathML, which provides a way of encoding both mathematical content and visual presentation for mathematics at all levels, from elementary school to scientific research.

Graphics

Graphics are the most visible part of the modern Web and arguably one of the primary reasons for its popularity and explosive growth. Successful use of graphics on the Web depends on interoperability across platforms, output resolutions, color spaces, and software products.

In October 1996, W3C issued a Recommendation for Portable Network Graphics (PNG), a format for bitmapped images. Interoperable methods of integrating CGM vector graphics have then been developed and constitute the WebCGM Profile Recommendation. W3C is now developing Scalable Vector Graphics (SVG), an open vector graphics format written in XML, and designed to work across platforms, output resolutions, color spaces, and a range of available bandwidths.

Voice Browser

W3C is working to expand access to the Web to allow people to interact with Web sites via spoken commands, and listening to prerecorded speech, music and synthetic speech. This will allow any telephone to be used to access Web-based services, and will be a boon to people with visual impairments or needing Web access while keeping their hands & eyes free for other things.

Internationalization

The Web was originally developed to enable people throughout the world to communicate with one another. Having a single system that can deal with all languages and cultures has many advantages: when the same protocols are used everywhere, the same software can likewise be used.

W3C has successfully stressed the role of Unicode as the basis for identifying characters in documents. Work is continuing on providing markup and style components for international needs.

Device Independence

W3C's Device Independence Activity is working to ensure a seamless Web for all access devices by reorganizing the Mobile Access and Television and the Web Activities into one.

Web services are becoming accessible from a wide range of devices from desktop PCs to in-car computers, TV, digital cameras, and cellular phones. W3C is well-positioned to lead development to avoid incompatibility and to achieve single Web authoring.

Open Source Code

Amaya

Amaya is a highly advanced and powerful Web client which acts as both a browser and an authoring tool. It has been designed with the primary purpose of being a testbed for experimenting with, testing and demonstrating new specifications and extensions of Web protocols and formats.

Before Amaya, Arena has been developed by W3C as a testbed for HTML and CSS.

About the User Interface Team

Currently ten members strong, W3C's User Interface team brings together some of today's most respected innovators in Web design tools. Its representatives have considerable knowledge in markup languages, style sheets, graphics, fonts, and internationalization, among other areas. The team is led by Vincent Quint, whose areas of expertise include electronic documents, document models, hypertext, and document production systems. The team also solicits external advice from leading experts in the field.

CHAPTER 4: DESIGN

4.1 DESIGN

1.1. INPUT

<https://AlgoApi.com?algo=mergesort>

1.2. OUTPUT

```
: {  
  
    algo: "merge sort",  
  
    sol: "#include <iostream>  
  
#include <vector>  
  
  
using namespace std;  
  
  
// Function to merge two sorted lists  
  
vector<int> merge(vector<int> left, vector<int> right)  
  
{  
  
    // Create a new list to store the merged elements  
  
    vector<int> merged;  
  
  
    // Set indices for the left and right lists  
  
    int i = 0;
```

```
int j = 0;
```

```
// Merge the lists
```

```
while (i < left.size() && j < right.size())
```

```
{
```

14

```
    if (left[i] < right[j])
```

```
    {
```

```
        merged.push_back(left[i]);
```

```
        i++;
```

```
    }
```

```
    else
```

```
    {
```

```
        merged.push_back(right[j]);
```

```
        j++;
```

```
    }
```

```
}
```

```
// Add any remaining elements from the left list
```

```
while (i < left.size())
```

```

{

    merged.push_back(left[i]);

    i++;

}

// Add any remaining elements from the right list

while (j < right.size())

{

    merged.push_back(right[j]);

    j++;

}”

}

```

4.2 OVERALL SYSTEMARCHITECTURE

Architecture

A web app architecture presents a layout with all the software components (such as databases, applications and middleware) and how they interact with each other. It defines how the data is delivered through HTTP and ensures that the client-side server and the backend server can understand. Moreover it also secures that valid data is present in all user requests. It creates and manages records while providing permission-based access and authentication. Choosing the right design defines your company growth, reliability and interoperability and future IT needs. As such, it is important to understand the components comprising the architecture of web apps.

Web Application Architecture Components

Typically a web-based application architecture comprises 3 core components:

- 1) Web Browser: The browser or the client-side component or the front-end component is the key component that interacts with the user, receives the input and manages the presentation logic while controlling user interactions with the application. User inputs are validated as well, if required.
- 2) Web Server: The web server also known as the backend component or the server-side component handles the business logic and processes the user requests by routing the requests to the right component and managing the entire application operations. It can run and oversee requests from a wide variety of clients.
- 3) Database Server: The database server provides the required data for the application. It handles data-related tasks. In a multi-tiered architecture, database servers can manage business logic with the help of stored procedures.

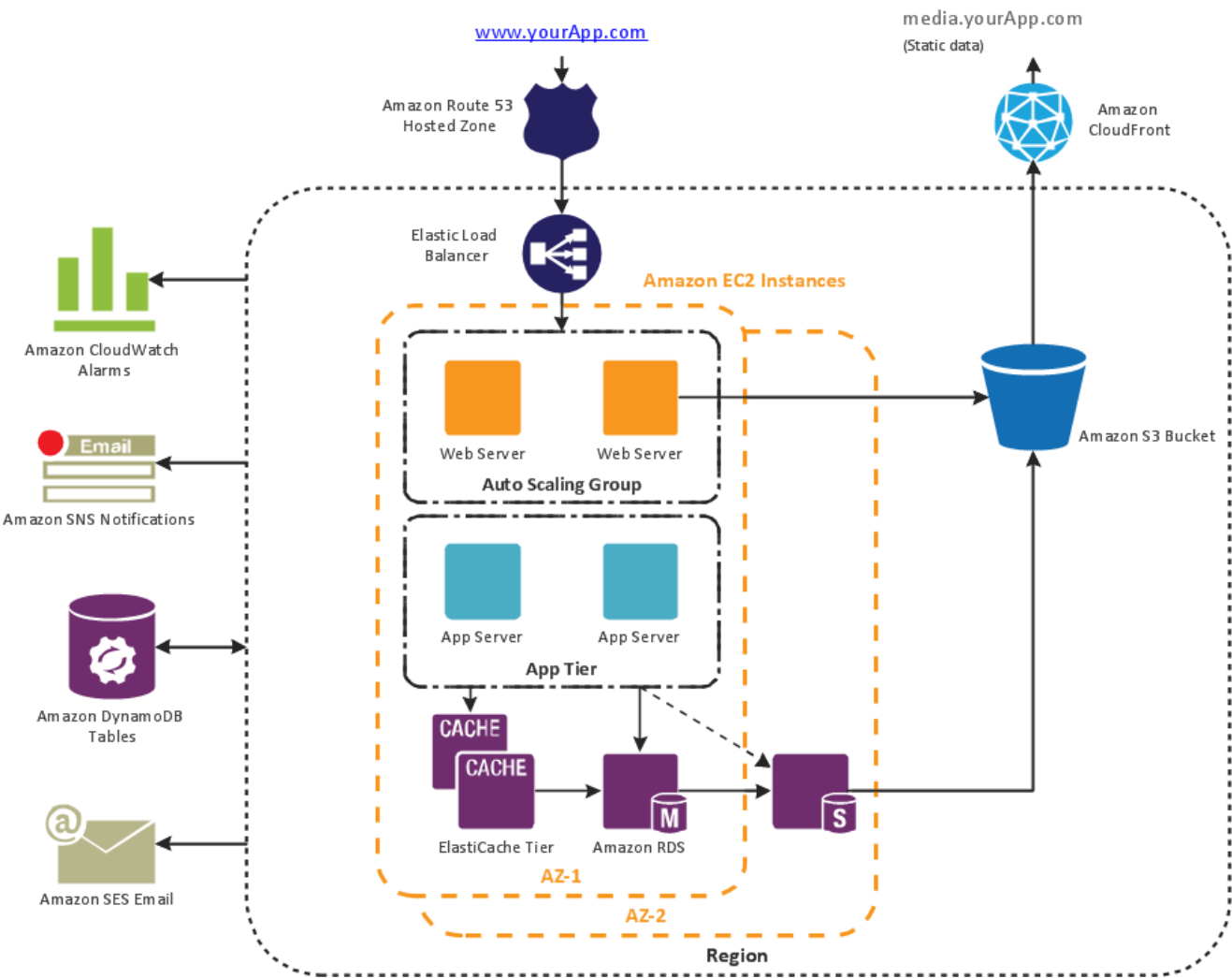
What is a 3-Tier Architecture?

In a traditional 2-Tier architecture, there are two components namely the client side system or the user interface and a backend system which is usually a database server. Here the business logic is incorporated into the user interface or the database server. The downside of 2-tier architecture is that with an increased number of users, the performance decreases. Moreover, the direct interaction of the database and the user device also raises some security concerns. Railway reservation systems, content management systems are a couple of applications that are usually built using this architecture.

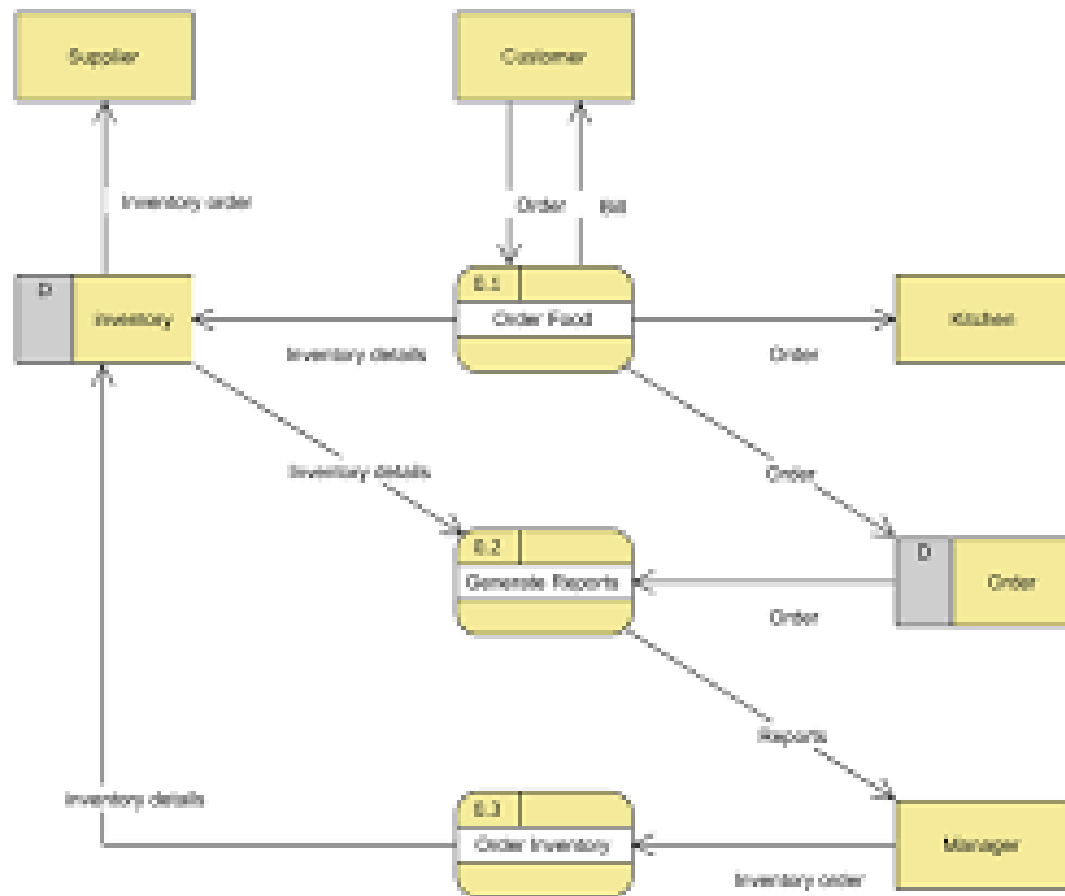
When it comes to 3-Tier architecture, there are three layers:

1. Presentation layer / Client Layer
2. Application Layer / Business Layer
3. Data Layer

4.3FLOWCHART



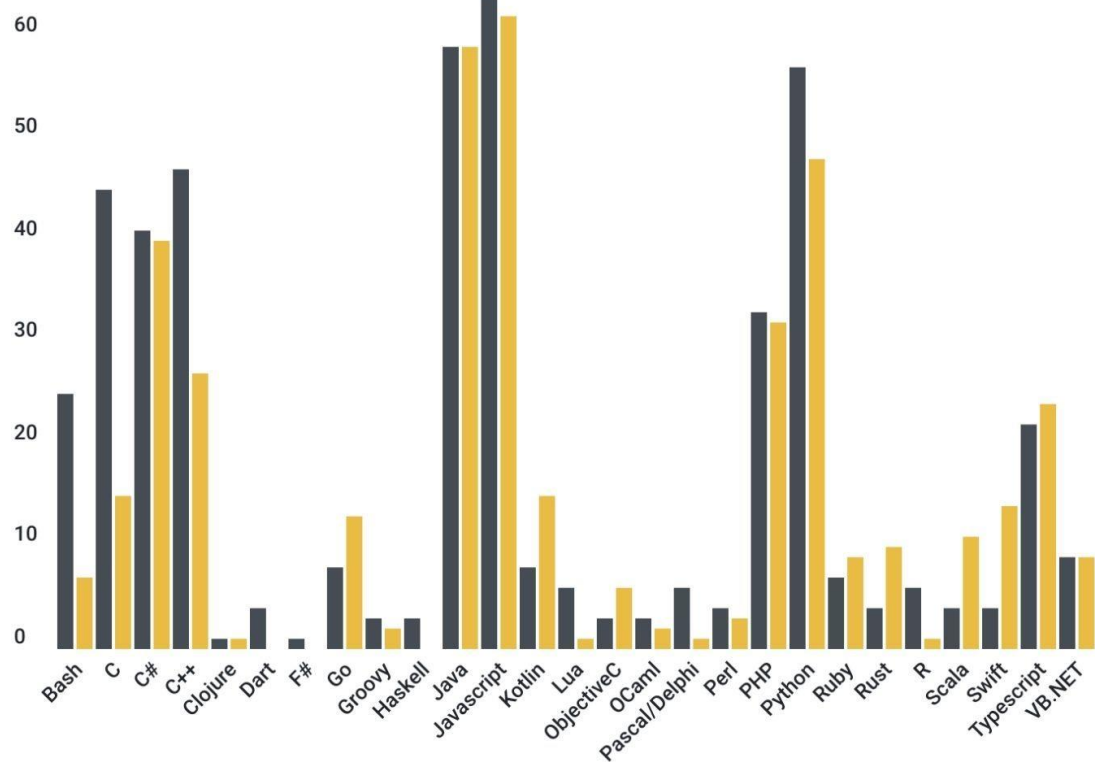
4.4 DATAFLOW



4.5 USE-CASE DIAGRAM



4.6SEQUENCEDIAGRAM



CHAPTER 5: IMPLEMENTATION

5.1 JSON

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.^[3] It has been called the de facto standard server framework for Node.js.^[4]

The original author, TJ Halewyck, described it as a Sinatra-inspired server,^[5] meaning that it is relatively minimal with many features available as plugins. Express is the back-end component of popular development stacks like the MEAN, MERN or MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library.

5.2 DATA REPRESENTATION

Fig 5.2



MongoDB Atlas

Work with your data as code

Documents in MongoDB map directly to objects in your programming language. Modify your schema as your apps grow over time.

Focus on building, not managing

Let MongoDB Atlas take care of the infrastructure operations you need for performance at scale, from always-on security to point-in-time recovery.

Simplify your data dependencies

Leverage application data for full-text search, real-time analytics, rich visualizations and more with a single API and minimal data movement.

5.4 BUILDING MODEL

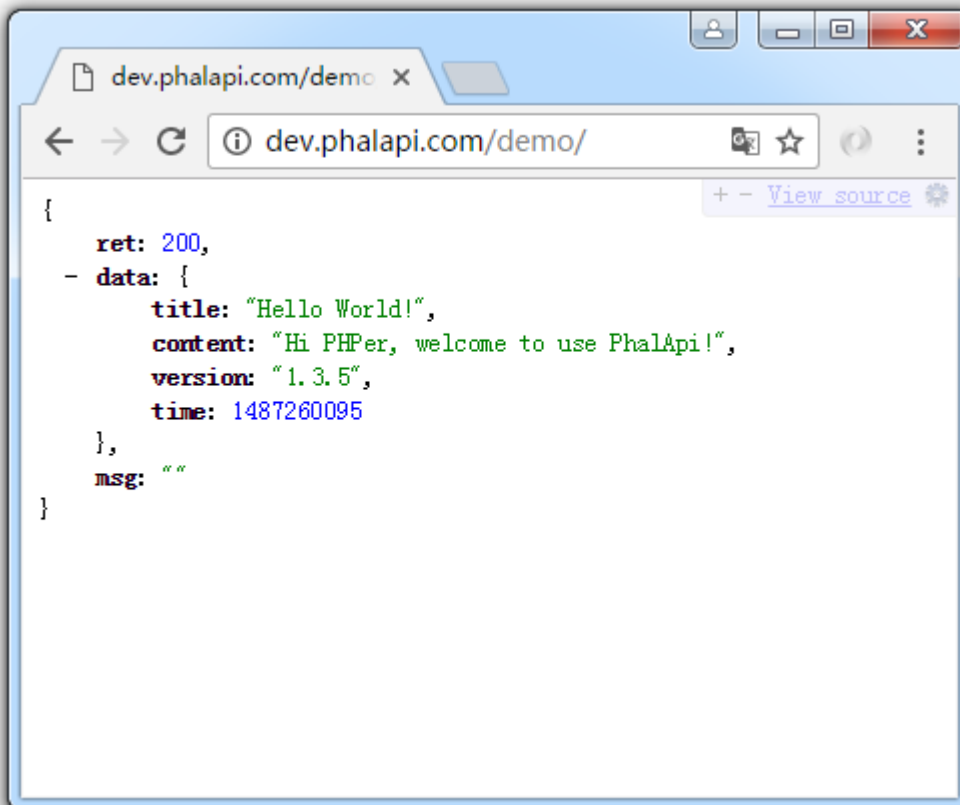
In earlier computing models like client-server, the processing load for the application was shared between code on the server and code installed on each client locally. In other words, an application had its own pre-compiled client program which served as its user interface and had to be separately installed on each user's personal computer. An upgrade to the server-side code of the application would typically also require an upgrade to the client-side code installed on each user workstation, adding to the support cost and decreasing productivity. In addition, both the client and server components of the application were usually tightly bound to a particular computer architecture and operating system and porting them to others was often prohibitively expensive for all but the largest applications (Nowadays, native apps for mobile devices are also hobbled by some or all of the foregoing issues).^[dubious – discuss]

In 1995, Netscape introduced a client-side scripting language called JavaScript allowing programmers to add some dynamic elements to the user interface that ran on the client side. So instead of sending data to the server in order to generate an entire web page, the embedded scripts of the downloaded page can perform various tasks such as input validation or showing/hiding parts of the page.^[2]

In 1999, the "web application" concept was introduced in the Java language in the Servlet Specification version 2.2. [2.1] that time both JavaScript and XML had already been developed, but Ajax had still not yet been coined and the XMLHttpRequest object had only been recently introduced on Internet Explorer 5 as an ActiveX object.

In 2005, the term Ajax was coined, and applications like Gmail started to make their client sides more and more interactive. A web page script is able to contact the server for storing/retrieving data without downloading an entire web page

5.5Java Script MODEL



JavaScript (/ˈdʒɑːvəskɹɪpt/), often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.^[10] It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

CONCLUSIONS

Ultimately, web application security is a major part of modern organizational risk management.

Web applications are more at-risk now than ever before, so it's imperative that executives and managers take the right steps necessary to secure their web applications against new threats.

Leveraging technology and smart methodologies can go a long way toward minimizing the effects of even the most dangerous cyberattacks.

The system has the potential to challenge traditional writing methods. It eradicates the need to

carry a mobile phone in hand to jot down notes, providing a simple on-the-go way to do the same. It will also serve a great purpose in helping especially abled people communicate easily. Even senior citizens or people who find it difficult to use keyboards will be able to use the system effortlessly. Extending the functionality, the system can also be used to control IoT devices shortly. Drawing in the air can also be made possible. The system will be an excellent software for smart wearables using which people could better interact with the digital world. Augmented Reality can make text come alive. There are some limitations of the system which can be improved in the future. Firstly, using a handwriting recognizer in place of a character recognizer will allow the user to write word by word, making writing faster. Secondly, hand-gestures with a pause can be used to control the real-time system as done by [1] instead of using the number of fingertips. Thirdly, our system sometimes recognizes fingertips in the background and changes their state. Air-writing systems should only obey their master's control gestures and should not be misled by people around. Also, we used the

EMNIST dataset, which is not a proper air-character dataset. Upcoming object detection algorithms such as YOLO v3 can improve fingertip recognition accuracy and speed. In the future, advances in Artificial Intelligence will enhance the efficiency of air-writing.

This project makes the user to have an interactive environment where the user can draw whatever he wants by choosing his required colors from the displayed ones. So, we conclude that Virtual Sketch is developed using the library NumPy and in Open CV where we have many libraries and algorithm built which makes the interfaces more active while using

We used python as, it has many inbuilt libraries and many modules which represent the imagination virtually when used along with OpenCV as well as its morphological processes.

REFERENCES

1. <https://en.wikipedia.org/wiki/MongoDB>
2. <https://www.mongodb.com/>
3. <https://en.wikipedia.org/wiki/API>
4. <https://nodejs.org/en/>
5. <https://en.wikipedia.org/wiki/Node.js>
6. <https://www.w3schools.com/nodejs/>
7. <https://expressjs.com/>
8. <https://en.wikipedia.org/wiki/Express.js>
9. <https://www.clever-cloud.com/deploy-cloud-managed-mysql/>
10. <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
11. <https://developer.mozilla.org/en-US/docs/Web/CSS>
12. <https://code.visualstudio.com/>
13. <https://www.geeksforgeeks.org/>
14. <https://en.wikipedia.org/wiki/C%2B%2B>
15. <https://medium.com/techie-delight/top-25-algorithms-every-programmer-should-know-373246b4881b>
16. https://en.wikipedia.org/wiki/Graphical_user_interface
17. https://en.wikipedia.org/wiki/User_experience
18. <https://chat.openai.com/chat>

19. <https://www.w3schools.com/REACT/DEFAULT.ASP>

REFERENCES:

[1]. Volume 5, Issue 1, January 2015 ISSN: 2277 128X, International Journal of Advanced Research in Computer Science and Software Engineering: Research Paper --Gesture Controlled Computer.

[2]. <https://towardsdatascience.com/Painting-with-opencv>

[3]. <https://www.geeksforgeeks.org/live-webcam-drawing-using-opencv>

[4]. Volume 43, Issue 1, June 2012 ISSN: 2277128X, International Journal of Advanced Research in Computer Science and Software Engineering: Research Paper --Hand Data Glove: A Wearable Real-Time Device for Human-Computer Interaction [5].

International Journal of Artificial Intelligence& Applications (IJAIA), Vol.3, No.4, July 2012, DOI: 10.5121/ijaia.2012.3412 161- HAND GESTURE RECOGNITION: A LITERATURE REVIEW

[6]. OpenCV for Computer Vision Applications, Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), March 20, 2015

[7]. <https://www.geeksforgeeks.org/live-webcam-drawing-using-opencv/> [8].

https://docs.opencv.org/master/d5/d54/group_objdetect.html

[9] <https://circuitdigest.com/tutorial/real-life-object-detection-using-opencv-python>-detecting-objects-in-live-video

[10]. Real Time Object Detection and Tracking Using Deep Learning and OpenCV Proceedings of the International Conference on Inventive Research in Computing)