ABSTRACT

The digital space in this generation is filled with many clouds like Google photos, Instagram, Facebook, flickr, Google drive, PC hardware drive etc giving options of storing photos. Do you find it difficult to remember where you've stored a particular photo? Do you wish to see it all in one place, sorted according to your wishes?

Zumba is a website which accesses all your photos from your multiple storage spaces and displays it. It does not save any of your photos except your website username and password. It allows you to sort and display photos using different categories using machine learning algorithms.

CONTENTS

FRONT PAGE

AKNOWLEDGEMENT

CERTIFICATE

ABSTRACT

CONTENTS:

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

The report discusses the result, the concepts and the work done in the development of the mini-project “Zumba-a collective photo displaying web application”. It aims to display all the photos present on different clouds on one platform.

1.2 WHAT IS WEB DEVELOPMENT?

Web development is a term that takes on different meanings depending on the audience and context. In practice, web development requires people with complementary but distinct expertise working together toward a single goal. Whereas a graphic designer might regard web development as the application of good graphic design strategies, a database administrator might regard it as a simple interface to an underlying database. Software engineers and programmers might regard web development as a classic software development task with phases and deliverables, where a systems administrator sees a system that has to be secured from attackers. With so many different classes of user and meanings for the term, it’s no wonder that web development is often poorly understood. Too often, in an effort to fully cover one aspect of web development, the other principles are ignored altogether, leaving students without a sense of where their skills fit into the big picture. A true grasp of web development requires an understanding of multiple perspectives. As you will see, the design and layout of a website are closely related to the code and database.

1.3 HTML

HTML is defined as a **markup language**. A markup language is simply a way of annotating a document in such a way as to make the annotations distinct from the text being annotated. Markup languages such as HTML, Tex, XML, and XHTML allow users to control how text and visual elements will be laid out and displayed. The term comes from the days of print, when editors would write instructions on manuscript pages that might be revision instructions to the author or copy editor.

At its simplest, **markup** is a way to indicate *information about the content* that is distinct from the content. This "information about content" in HTML is implemented via **tags.**

1.3.1 HTML SYNTAX

HTML documents are composed of textual content and HTML elements. The term **HTML element** is often used interchangeably with the term **tag**. However, an HTML element is a more expansive term that encompasses the element name within angle brackets (i.e., the tag) and the content within the tag (though some elements contain no extra content). An HTML element is identified in the HTML document by tags. A tag consists of the element name within angle brackets.

The element name appears in both the beginning tag and the closing tag, which contains a forward slash followed by the element's name, again all enclosed within angle brackets. The closing tag acts like an off-switch for the on-switch that is the start tag. HTML elements can also contain attributes. An **HTML attribute** is a name=value pair that provides more information about the HTML element.

1.4 CASCADING STYLE SHEETS(CSS)

CSS is a W3C standard for describing the appearance of HTML elements. Another common way to describe CSS's function is to say that CSS is used to define the **presentation** of HTML documents. With CSS, we can assign font properties, colours, sizes, borders, background images, and even position elements on the page. CSS can be added directly to any HTML element (via the style attribute), within the <head> element, or, most commonly, in a separate text file that contains only CSS.

1.4.1 CSS SYNTAX

A CSS document consists of one or more **style rules**. A rule consists of a **selector** that identifies the HTML element or elements that will be affected, followed by a series of **property: value pairs** (each pair is also called a **declaration**), as shown in Figure 3.2. The series of declarations is also called the **declaration block**. As one can see in the illustration, a declaration block can be together on a single line, or spread across multiple lines. The browser ignores white space (i.e., spaces, tabs, and returns) between your CSS rules so you can format the CSS however you want. Notice that each declaration is terminated with a semicolon. The semicolon for the last declaration in a block is in fact optional. However, it is sensible practice to also terminate the last declaration with a semicolon as well; that way, if you add rules to the end later, you will reduce the chance of introducing a rather subtle and hard-to-discover bug.

1.5 JAVASCRIPT

JavaScript is an object-oriented, dynamically typed, scripting language. Also, it is primarily a client-side scripting language.

DIFFERENCE BETWEEN JAVASCRIPT AND JAVA

Although it contains the word *Java*, JavaScript and Java are vastly different programming languages with different uses. Java is a full-fledged compiled, object-oriented language, popular for its ability to run on any platform with a Java Virtual Machine installed. Conversely, JavaScript is one of the world's most popular languages, with fewer of the object-oriented features of Java, and runs directly inside the browser, without the need for the JVM. Although there are some syntactic similarities, the two languages are not interchangeable and should not be confused with one another.

1.5.1 JAVASCRIPT SYNTAX

JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script>. You can place the <script> tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the <head> tags. The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows. The script tag takes two important attributes −

- Language − This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- Type − This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

So your JavaScript segment will look like −

```
<script language="javascript" type="text/javascript">
   JavaScript code
</script>
```

1.6 NODE JS

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

```
Node.js = Runtime Environment + JavaScript Library
```

1.6.1 FEATURES OF NODE JS

Following are some of the important features that make Node.js the first choice of software architects.

- **Asynchronous and Event Driven** − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- **Very Fast** − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** − Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** − Node.js applications never buffer any data. These applications simply output the data in chunks.
- **License** − Node.js is released under the MIT license.

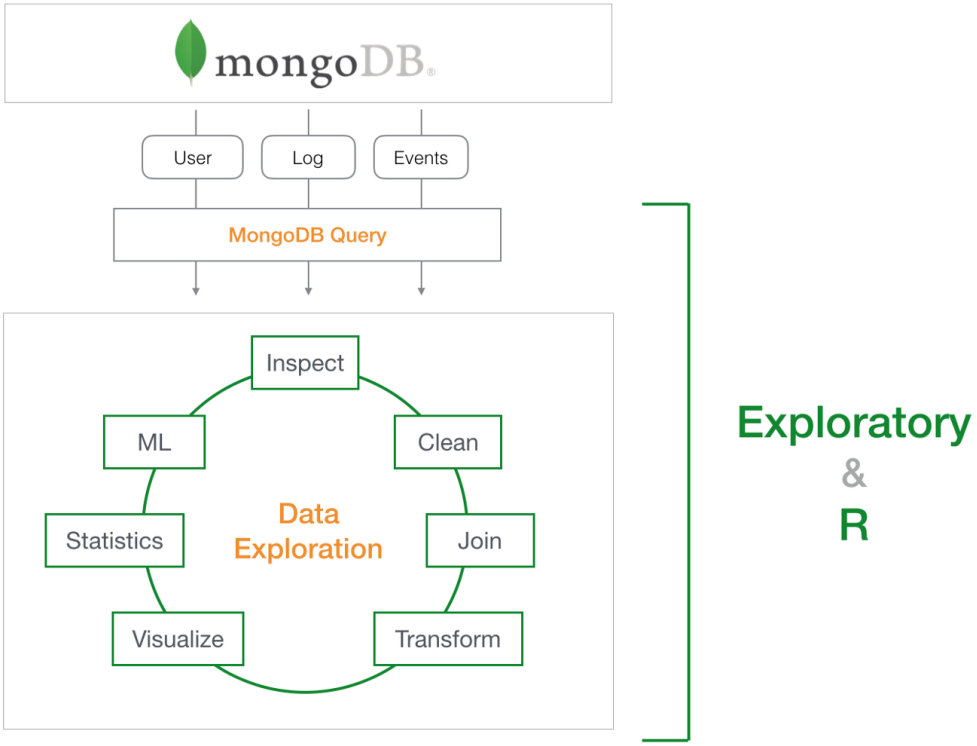
1.7 MONGO DATABASE

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document. A single MongoDB server typically has multiple databases.

Following example shows the document structure of a blog site, which is simply a comma separated key value pair.

```
{
  _id: ObjectId(7df78ad8902c)
  title: 'MongoDB Overview',
  description: 'MongoDB is no sql database',
  by: 'tutorials point',
  url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'],
  likes: 100,
  comments: [
    {
      user:'user1',
      message: 'My first comment',
      dateCreated: new Date(2018,1,20,2,15),
      like: 0
    },
    {
      user:'user2',
      message: 'My second comments',
      dateCreated: new Date(2018,1,25,7,45),
      like: 5
    }
  ]
}
```

_id is a 12 bytes hexadecimal number which assures the uniqueness of every document. You can provide _id while inserting the document. If you don't provide then MongoDB provides a unique id for every document. These 12 bytes first 4 bytes for the current timestamp, next 3 bytes for machine id, next 2 bytes for process id of MongoDB server and remaining 3 bytes are simple incremental VALUE.



CHAPTER 2: LITERATURE SURVEY

2.1 MOTIVATION

There are many websites like Trivago, Justdial etc which collect information desired for their objective. For example, Trivago is a website which collects all the information regarding hotels, sorts it and displays it on the basis of prices. This way, all the information regarding a particular topic is found under one roof. Zumba is a similar Web application trying to bring all the photos under one roof so that you can manage them all well.

There are no other websites found which can integrate so many API calls at once which implies that Zumba is a unique and new idea.

2.2 SCOPE and LIMITATION

The objective of the project is not completely achievable as yet. The most important aspect of this project are its API calls. If the APIs are not available, the project is partially complete. Facebook API called Graph requires Security survey and privacy policies to be written and handed over for them to allow access to desired components. Since Instagram is a part of Facebook, the Instagram API is time consuming to be accessible but not impossible.

Different libraries from ***Express js*** are used to make the programming easier. ***FancyBox*** has been used for image showing layout. Currently only google API is accessible. Hence, the users who have photos synced to their google account can sign in and view their photos. Mongo Database is an unstructured efficient database that is written in json.

2.3 SECURITY

The login system is accompanied by hashed cryptographic techniques. The google API has its own secure login authentication system, so does Facebook and Instagram. The Website user has his/her own separate username and password which are securely hashed and saved.

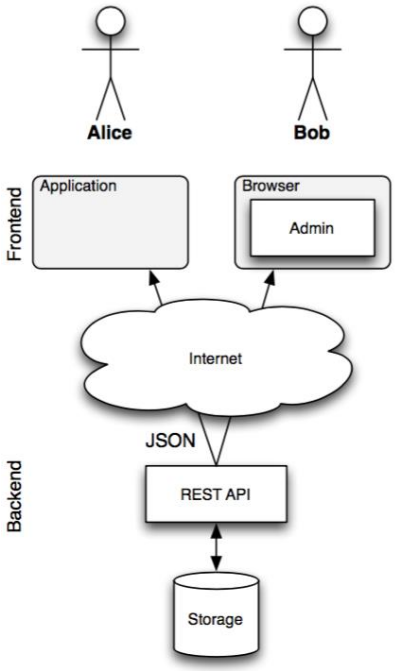
2.4 IMPLEMENTATION

2.4.1 DESIGN

- Body colour used is material dark which has the hex value #191414.
- The default font family is Poiret-One in cursive which is a google font.
- The default text-colour on the body colour has the value #e49023.
- Animate.css is a library which is used to animate objects.
- Ejs is an embedded JavaScript template used instead of html. It is same as html but has a greater advantage.

2.4.2 API CALLS

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is the messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.



2.4.3 GOOGLE API

ACCESSING API WITH GOOGLE-API-CLIENT

To access a service that requires user authorization, first sign the user in, and request permission to access the scopes required by the service. Then, get an instance of the service's client object, passing it the user's GoogleSignInAccountobject in addition to a Context or Activity.

```
// This account must have the necessary scopes to make the API call
// See https://developers.google.com/identity/sign-in/android/
GoogleSignInAccount account = GoogleSignIn.getLastSignedInAccount(this);

// Get the app's Drive folder
DriveResourceClient client = Drive.getDriveResourceClient(this, account);
client.getAppFolder().
      .addOnCompleteListener(this, new OnCompleteListener<DriveFolder>() {
         @Override
         public void onComplete(@NonNull Task<DriveFolder>() {
            // ...
         }
      });
```

AUTHENTICATING YOUR CLIENT

Certain Google Play services (such as Google Sign-in and App Invites) require you to provide the SHA-1 of your signing certificate so we can create an OAuth2 client and API key for your app. To get your SHA-1, follow these instructions:

Open a terminal and run the keytool utility provided with Java to get the SHA-1 fingerprint of the certificate. You should get both the release and debug certificate fingerprints.

To get the release certificate fingerprint:

```
keytool -exportcert -list -v \
-alias <your-key-name> -keystore <path-to-production-keystore>
```

CHAPTER 3: HARDWARE AND SOFTWARE REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

1. Pentium or higher processor
2. 64-bit system
3. Internet connection

3.2 SOFTWARE REQUIREMENTS

1. Browser support – Google Chrome is recommended.
2. Sublime text 3 used as text editor.
3. Ubuntu terminal/ bash shell for server commands used.
4. Node js- Express js for Server-side programming tool
5. Mongo DB used for login system database

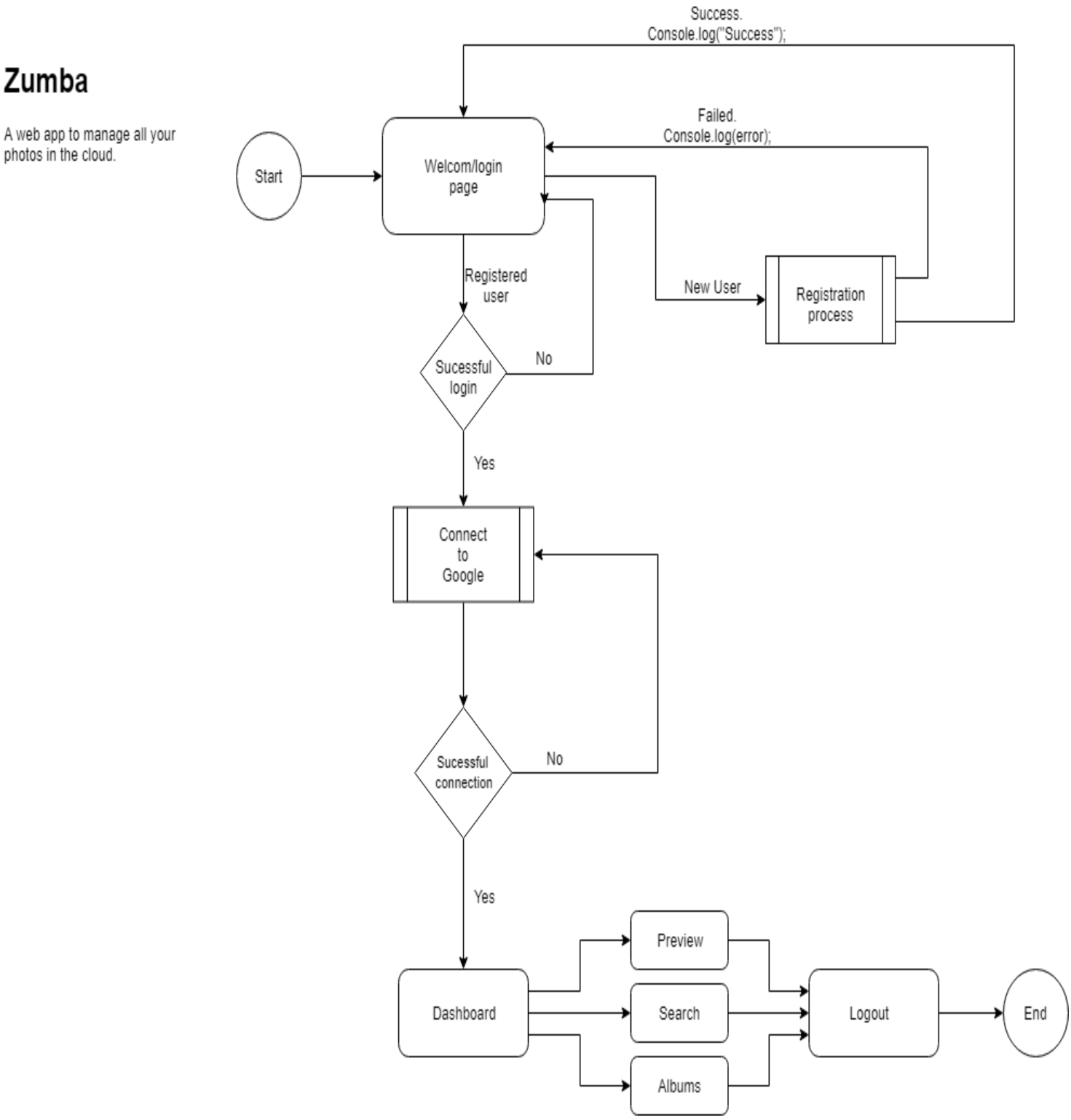
3.3 PROGRAMMING LANGUAGES USED

1. HTML
2. CSS
3. JAVASCRIPT
4. EJS

3.4 DESIGN PLATFORM

MATERIAL LITE DESIGN

CHAPTER 4: FLOWCHART



CHAPTER 5: SCREENSHOTS

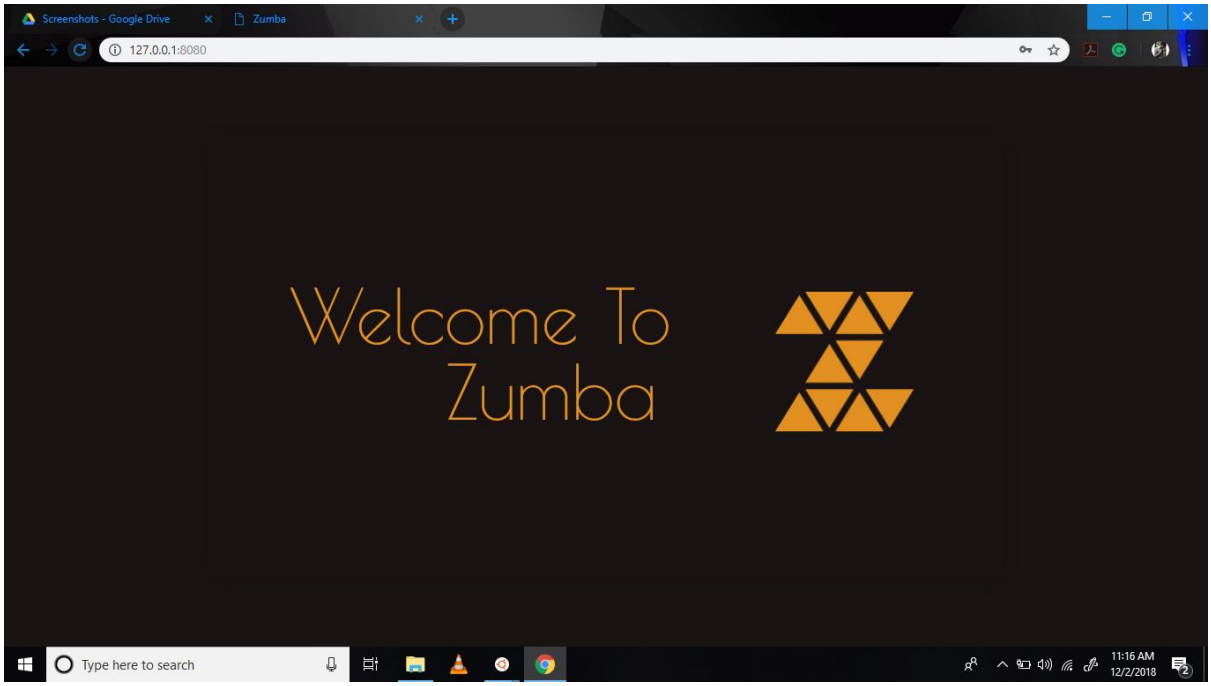


Figure 1: Landing Page of the website

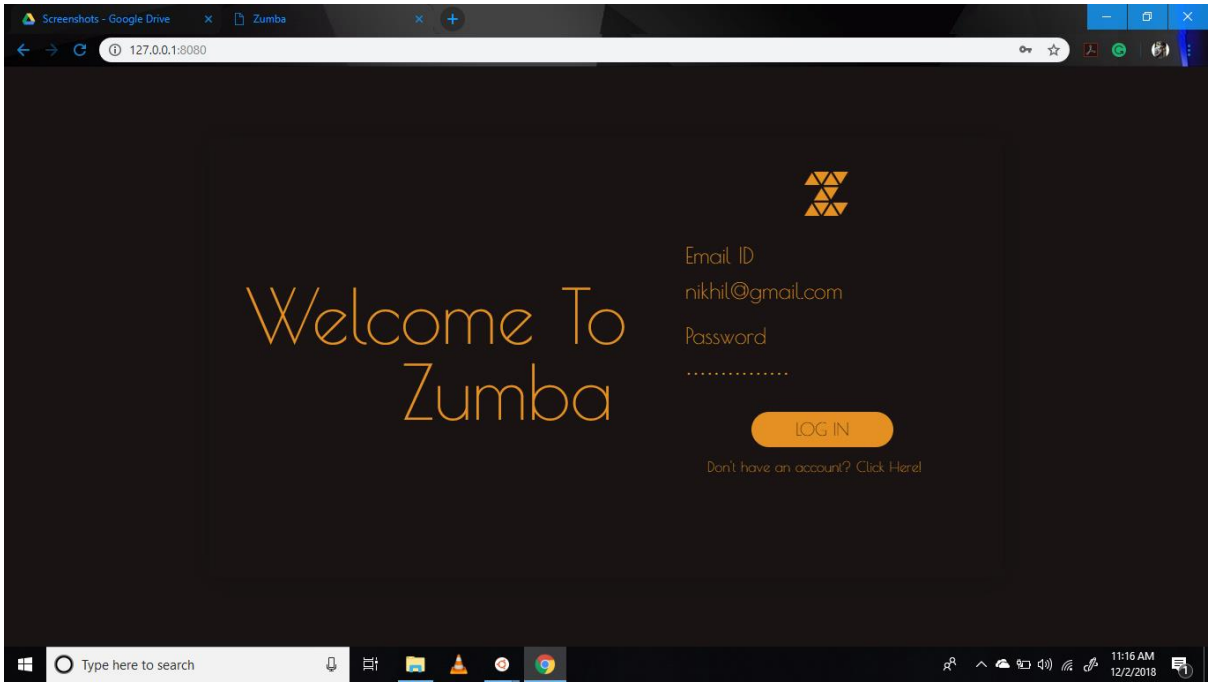


Figure 2: Login Page

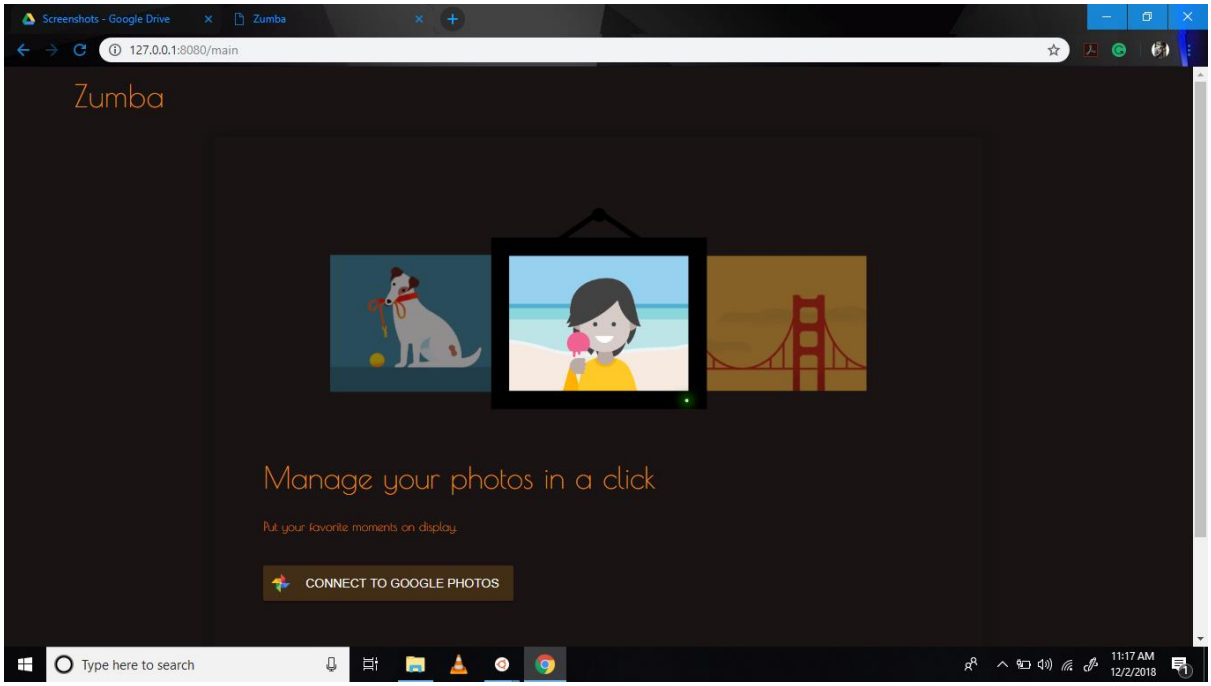


Figure 3: Home Page

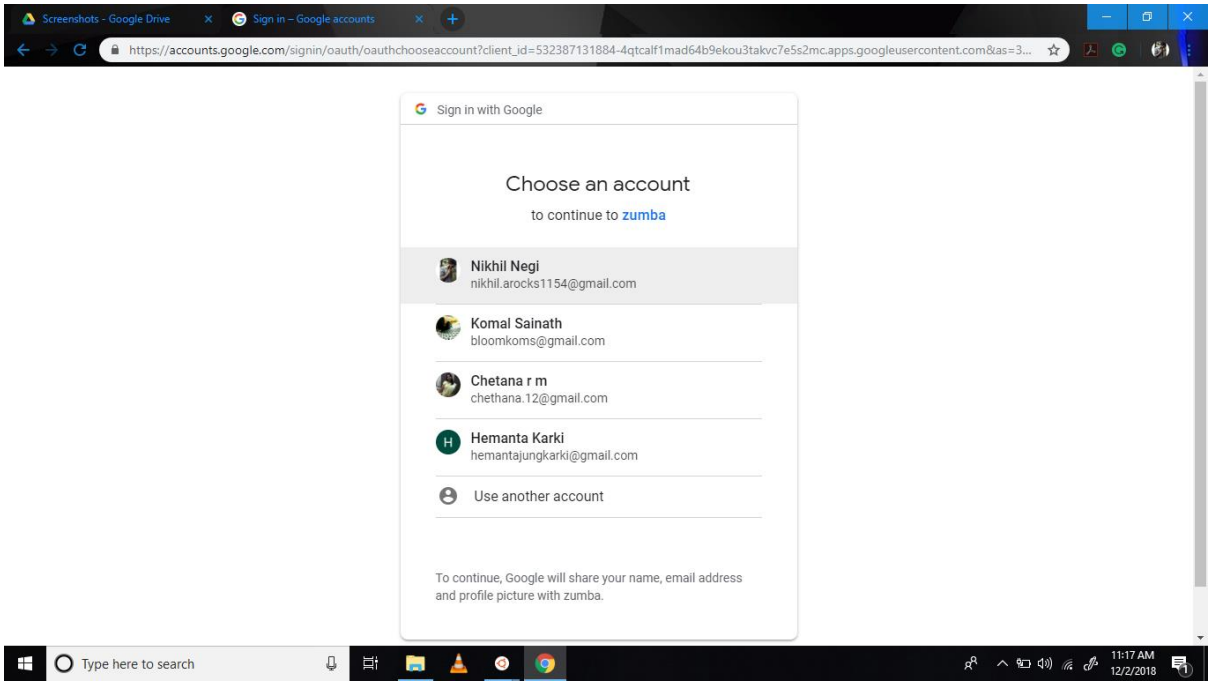


Figure 4: Google Sign-In Page

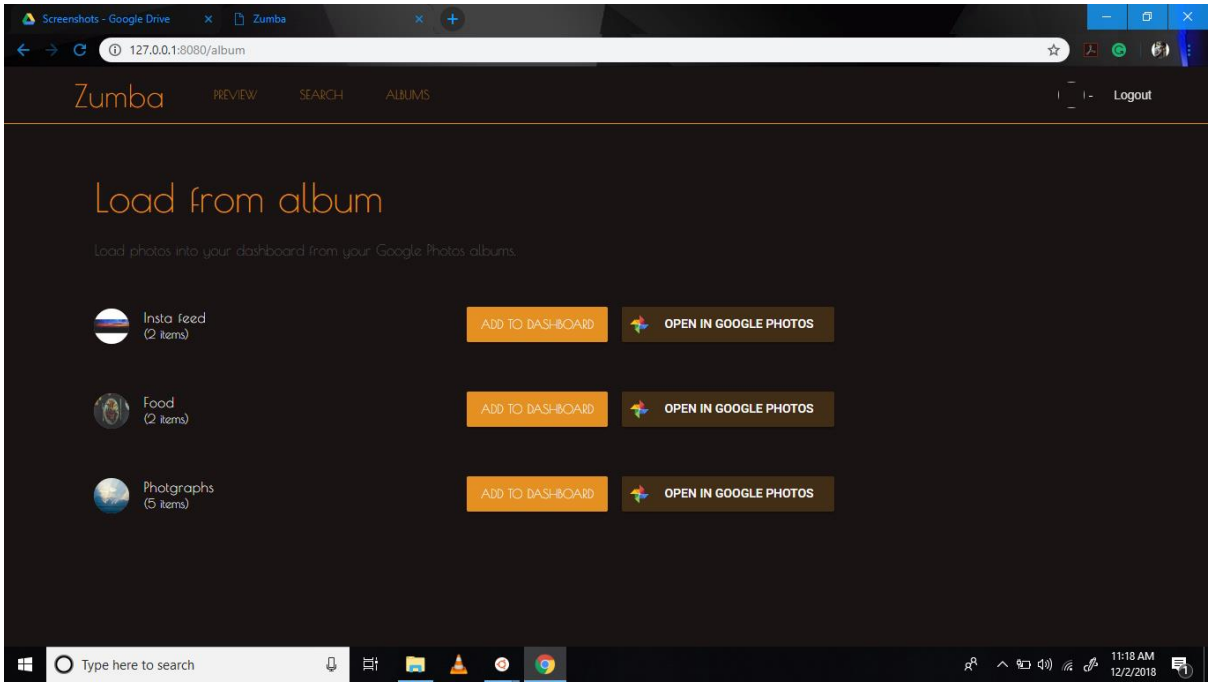


Figure 5: Albums to be added to the dashboard

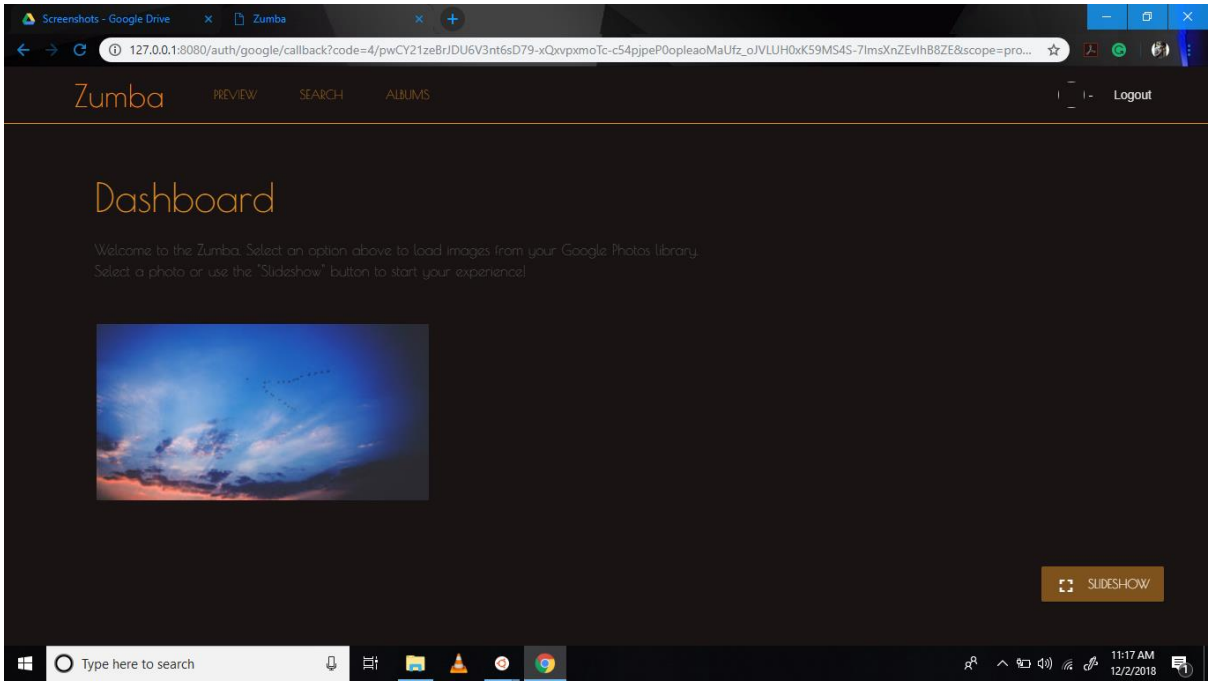


Figure 6: dashboard of the selected album

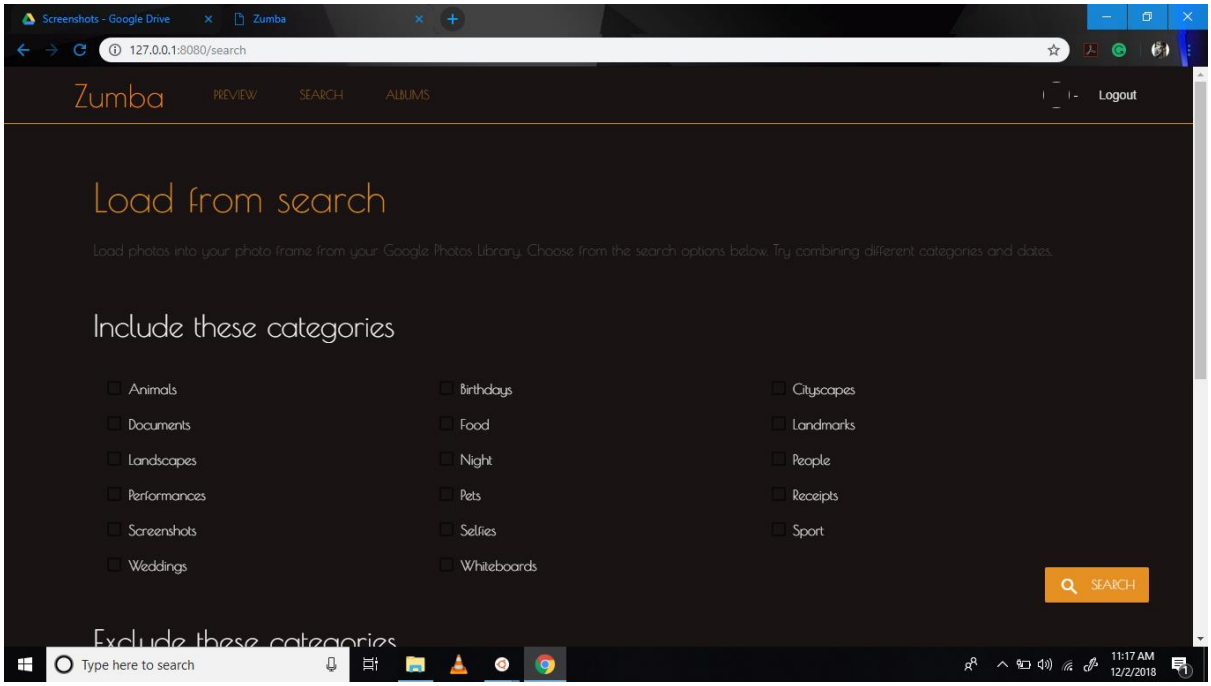


Figure 7: Choose categories to display photos from search

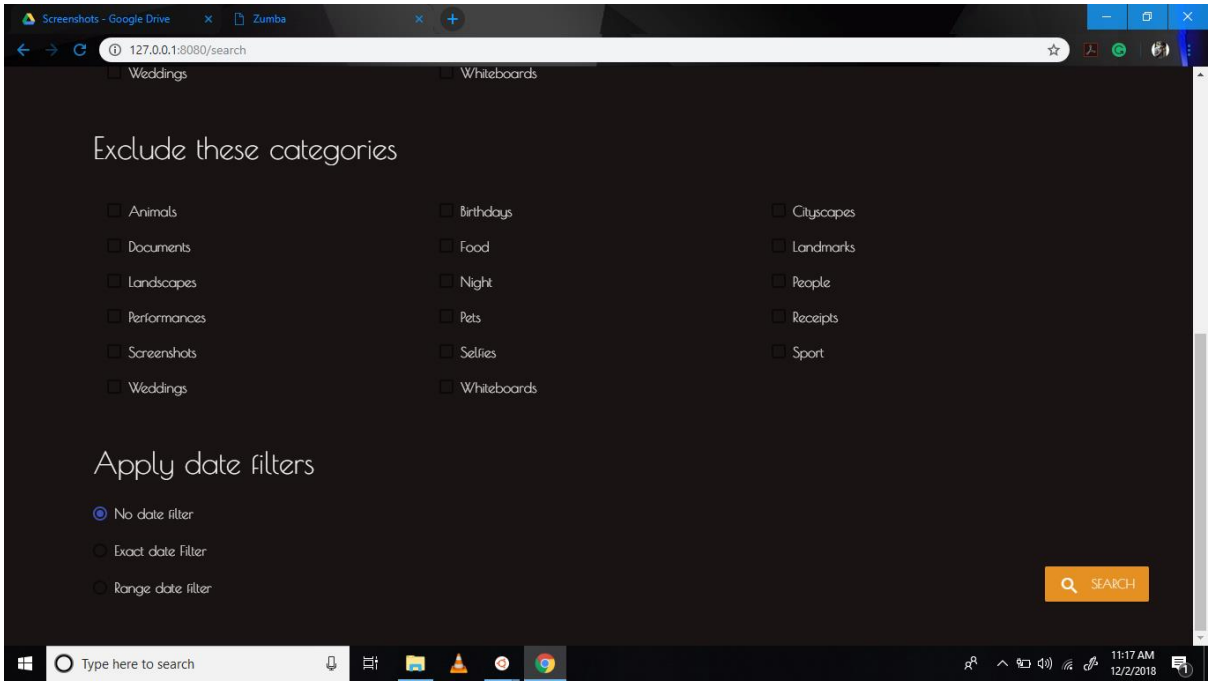


Figure 8: Choosing categories to display photos from

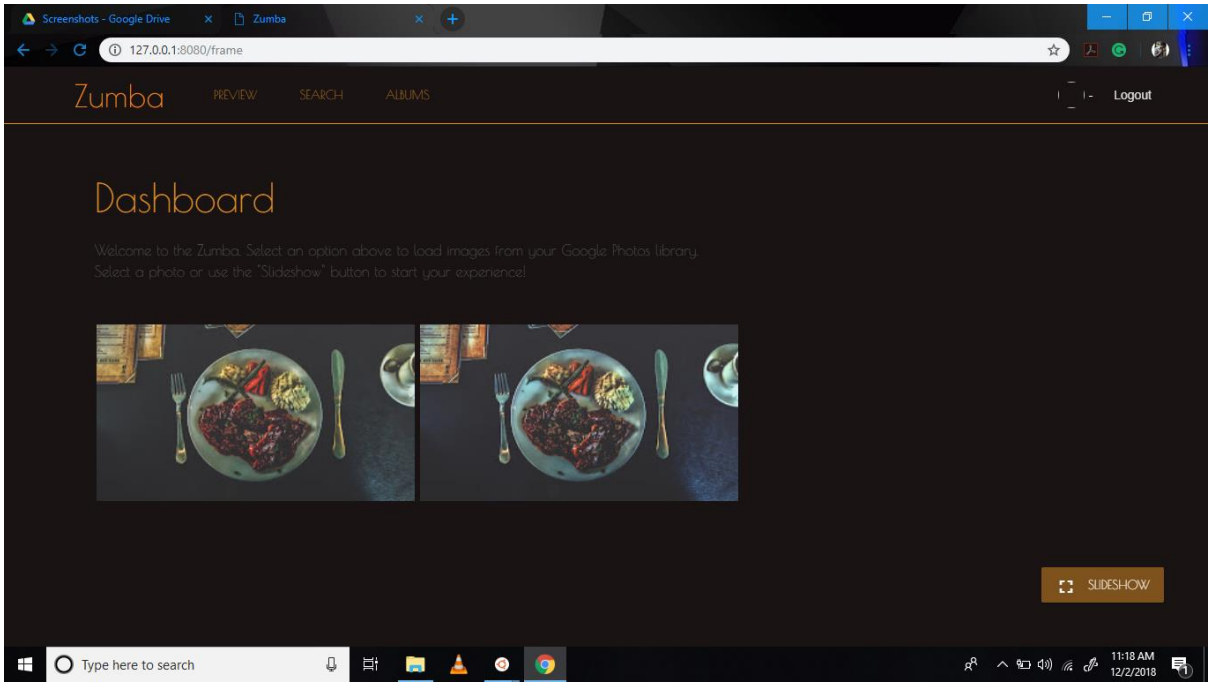


Figure 9: Displaying photos from chosen category

CHAPTER 6: ADVANTAGES AND DISADVANTAGES

6.1 ADVANTAGES

- The photos stored in all the clouds are just being displayed on the website. Hence, there are no issues regarding memory management of the photos.

- The cache and cookies are destroyed automatically after every 10 minutes starting from the time the users log-In.

- You can connect to multiple clouds and organise them according to the multitude of categories to choose from.

6.1 DISADVANTAGES

- This website depends on the API provided by the cloud companies so that the information they are holding can be accessed. Some clouds have no API call facilities while other clouds like Facebook API makes it a hectic process to access certain information.

- The website will only be accessible when the servers are on and are responding back.

- Internet connection is a compulsory requirement.

- The machine learning algorithm used can go wrong sometimes.

CHAPTER 7: FUTURE ENHANCEMENTS and CONCLUSION

1. Addition of many more clouds like Instagram cloud, google drive, DropBox etc.
2. Editing the photos and enabling canvas drawing on them and saving it separately.
3. Archiving photos
4. Shuffling of photos from different clouds and slideshow
5. Use of alternative image layouts
6. Integration of videos
7. Easier user Interface.

We have tried our level best to build the project efficiently and correctly and have succeeded in building a better project but may not be a best project

CHAPTER 8: BIBLIOGRAPHY AND REFERENCES

- https://developers.google.com/android/guides/api-client
- https://linuxize.com/post/how-to-install-node-js-on-ubuntu-18.04/
- https://docs.mongodb.com/
- https://www.w3schools.com/tags/default.asp
- https://css-tricks.com/almanac/
- http://paletton.com/#uid=31L0u0kllllaFw0g0qFqFg0w0aF
- https://codepen.io/
- https://developers.facebook.com/
- https://fonts.google.com/
- https://www.codeofaninja.com/2011/06/display-facebook-photos-to-your-website.html
- Fundamentals of Web Development by Randy Connolly and Ricardo Hoar, Pearson Publications
- JavaScript and jQuery, the missing manual by David Sawyer McFarland, O'Reilly Publications