

# Automated Production of Personalised Doodles

Akash Raja A<sup>1</sup>, Dr. Sudha Sadhasivam G<sup>2</sup>  
{akashraja1995@gmail.com<sup>1</sup>, gss.cse@psgtech.ac.in<sup>2</sup>}

UG scholar, Dept. of Computer Science and Engineering, PSG College of technology, Coimbatore, India<sup>1</sup>, Professor and Head, Dept. of Computer Science and Engineering, PSG College of technology, Coimbatore, India<sup>2</sup>

**Abstract.** Doodles celebrate interesting events and anniversaries around the world by capturing one's attention. Doodles reflect Google's personality and love of innovation. Although doodles are simple scribbled piece of text, they mean more than that. Art and Animations have added real value to them. For instance, google portrays its logo in a creative fashion, with a new one every other day, by making use of these doodles. They are not only meant for creating an impact, but they also provide at the backdrop, a valuable piece of information to the customer who visits them. This information may comprise the current affairs, festive occasions or day-to-day important happenings around the world. Each doodle developed by Google goes through a series of procedures that involve deep thinking and designing activities. The timespan for producing each doodle is estimated to be a month. This paper suggests a procedure for automating the production of doodles using a novel approach. This approach uses scrapped web data relating to the particular dates and specialties about them (from sources like Wikipedia and Hindu API which have huge potential for information). Images pertaining to the gathered event related data are to be used as textures, which acts as a basic technique to overlay on the textual information or logo. Random selection of styles, designs and animations could add to the performance, by producing several variants of doodles. The linked information behind these doodles could add to the knowledge of the customers thereby increasing the productivity, clicks and visits to the site. This process aids to transform doodles from a generic version to a personalized version. A sync to the customer's calendar dates allows for personalized promotions. For example, Amazon may optionally deliver gifts from one customer to another by texturing their logos with the sender's photos. This may be overwhelming to both receiver and sender. Automation, therefore eases the procedures involved behind the pleasant doodles.

**Keywords:** Doodles, automation, sentiment, event date, personalization.

## 1 Introduction

Doodle as the definition states is 'some scribbled piece of text'. The famous example of doodles that strikes the mind directly are the Google Doodles. Even though the concept of doodles may look simple, extensive research is essential to automate their creation. Doodles embed Google's Logo within the content they present. They as perceived by Google as a means for conveying emotion and therefore add a new look and feel to their page, which the world uses heavily. Google also hosts competitions for creating impressive and personalized doodles.

A system to personalize the doodles to a large customer base is not yet available. This can provide the customers with fun and add emotional attachment to the organization presenting them with the doodles. Hence there is a need for a system that generates random doodles personalized to the consumer, and also at the backdrop provide some valuable information related to the content of the day.

At present, customer personalization occurs to a level at which the possible areas towards which a user is most likely to be driven are taken into account, based either on demographical identifiers or some calculative metrics that define the user interests. These personalization concepts try to make things customer-oriented rather than product-oriented. These concepts have brought remarkable changes to the way a customer sees and makes use of the application. But, people do tend to leave some gaps, at levels of personalization, where they tend to focus more toward the interior leaving the exterior opportunities detached.

Facebook tries to make things personalized for a customer. His birthdays, friendship details, festive occasions etc. appear to be made especially for him, weaved under a single common logic. Google Doodles are something people perceive every other day and get surprised each time they see something new. Amazon is the earth's most customer centric company and personalization is the key aspect of their business that they believe should be followed at all levels. How can a connection be established between the three and benefit the customer with something that excites him and arouse his curiosity; and at the same time provide him with some valuable information that could add to his knowledge?

Everyone wants their name to be sounding and reach places. A name in technical terms, is a primary key, with violations of the uniqueness property. This violation is handled by a large number of biological features associated with a human being, which doesn't concern us much. In a similar manner, the concept of personalized doodles is closely associated with a customer. These may be either provided to him attached to the cardboard box that is used for packaging products or via an online mechanism that gets delivered to the user each time he makes a purchase. An online mechanism can be attributed with more effects like animations and other easily accessible links to the intended information which is to be conveyed to the customer. The customer gets something that is personalized to his name, an additional information related to the delivery date and along with it lots of smiles and emotions. A customer can be turned to a child by exposing the curiosity inside.

These personalized doodles may not be restricted to convey information, but to make as meaningful designs as possible with code. Designing at present is an art, which is levied to the creative thinkers who make use of software like Photoshop, and Illustrator, to bring out something attractive and meaningful. But, it isn't practically feasible to create designs for each user manually, when the customer base matches the number of people buying products at Amazon, search Google every day for some information or number of users who visit Facebook every day for monitoring their feeds. This calls for an automation process that is creative enough to cater to the personalized needs of each customer. The ultimate objective is to make the customer believe that the doodle has been produced exclusively for him, knowing his needs. This is not a series collection of trump cards or kinder joy toys which is universal, but rather a doodle toy that is personalised for the customer.

These doodles, apart from the personalized customer name can be attributed to show some event notifications or propagate some message to the community. These notifications can themselves be embedded to produce some doodle, which acts as both an advertising strategy and convey the information in a more powerful manner. As the quote goes "A picture is worth a thousand words". The objective of the work is to build a system that automatically generates content based doodles that is self-sufficient to achieve the desired design goals for

productive use and at the background befriend the customer as an information source providing potential knowledge.

Related work is presented in section 2. System requirements and architecture are described in sections 3 and 4 respectively. Implementation details are provided in section 5. Section 6 presents the results followed by conclusion

## **2 Related work**

No such profound efforts have been recorded as to automate the process of creating doodles.

As far as creation of doodles are considered, Google gains the prime position with their changing logos to appreciate innovations in the elements of the past. These innovations include the scientists, artists, inventors, musicians and political figures. It is done as a means to connect emotionally with the users.

Most doodles at Google are created out of simple graphics. At times, simple animated GIFs are the ones that are presented as doodles. The ones that come up to the competence levels starts with proper use of HTML and JavaScript to animate and move objects around. The challenging makes up to their standards by means of technologies like HTML5 and Flash. At times, doodle creation competition are rolled out to people to showcase their creativity and passion towards both doodles and art. [1]

The average time spent by Google Engineers in creating a doodle is typically four weeks [2] and sometimes iterations go back and forth when the doodle is planned for international search engine releases to ensure the spirit of the event or person they are trying to celebrate is captured to the fullest. [3]

Google makes use of a filmstrip technique for creating animations. These are advantageous over GIF's in the way that the color palette is not restricted to 256 colors as in GIFs, the memory required is also comparatively lesser and also the animations are less jerky [4] [5].

To play music of an expert artificial intelligenced powered doodles are designed by Google. Machine learning model is trained using the expert's music nuances. This trained model should be embedded in the web browser [6]. This helps to personalize doodles to some extent.

## **3 System Requirements**

The system must appropriately map the events to the dates and provide users with a variety of doodle options which varies in the content (Image), layout, fonts and animations. The randomness plays a key role here in bringing out the best possible doodles. The sentiment of the sentence is to be analyzed to appropriately identify the pre-tagged fonts and animations that matches the mood of the content. A customization element at the backdrop will be really helpful which allows a user to create his own doodle by modifying some aspects of the doodle. Say, a user may feel that a particular font suits the situation better than the one provided by the system. So, he should be able to change these customizable items and perceive the changes in an interactive manner.

The system must be stable enough to provide the user with custom templates in case no image turned out as a result of failure to fetch data at the backend due to network failure or no proper data related to the content to be fetched has been found. The system must be extendable to allow more layouts to be defined and provide the users of the system with compatible options for creating more templates to benefit from.

Fonts play a key role in how the doodle is experienced by the user. A font that can capture heights of negativity for an event like blood-spills would be highly appreciated. The font-color written in red with horrifying text will add more value rather than some fonts like Jokerman that is something meant for a happy or pleasant event.

Animations must reflect the sentiment born out of the content. A joyous event like a festival may be accompanied some bright lights or glittering effects and an uneventful happening should capture emotions in sepia or black and white shades. Systems that can identify the negativities or positive impact of the images and capture the elements that are responsible for bringing out that impact must be developed. These responsible elements can then be added to the doodle to create the effect that is to be born out of the doodle.

The templates in the future must be adaptable to changes where custom objects can be designated for building the characters in a step-by-step manner and the backend may deal with the logic to identify the content and appropriately deliver in a random manner the identifier for the custom objects. We can also have an entire custom object replacing the characters in the text provided unlike the previous approach where the character was built out of the custom objects. The custom objects must be something that is related to the scene in place in the image.

The system must be stable and meet the following functionalities:

- Must produce random results as far as possible to bring out increased variations to satisfy customer needs.
- Guide the customer with messages in case the validation checks get failed.
- Allow customization of the doodle by selecting a specific doodle among the lot and changing attributes like the background image, font and animations applied.
- Provide the customer with a specified number of doodles provided as input.
- Provide the customer with a link to the relevant content from which the image has been picked for displaying to the user.

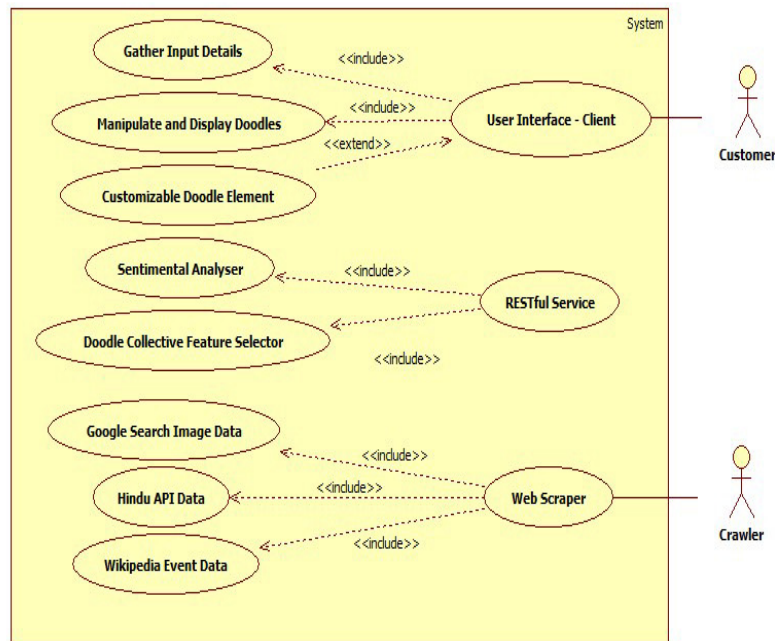
The system must be extendable for future changes in the template. The system must be reliable and must provide with a bare minimal set of default images in case no image related to the content is available. The system must be able to handle multiple requests in parallel provided by multiple users of the system relating to different themes and content. Once the system requirements are obtained, the architecture of the proposed system is developed as shown in the next section.

## **4 System Architecture**

There are two major users of the system – customers and Web crawler. The Web crawler extracts information from the Web and stores it. Using this information, doodles are designed. Users of the system include e-commerce vendors, advertising agencies and logo designers. E-commerce platforms can create personalized tags of interest to their customers in form of doodles to boost their sales.

The same advertisement or message may be delivered in different templates and can also be made personalized to a user. For example, an AC advertisement stating ‘Hi ABC, get a new AC this summer’ to one user who doesn’t own an AC with general themes related to the AC and another advertisement ‘Hi XYZ, get a new AC with deductions for your old AC’ . Designing a logo for an organization can be simplified using a doodle generator. This section describes the system by specifying the behavioral model, interaction model followed by the system architecture

*Behavioral Model:* Behavior of the system is described using usecase model of the system comprising of three major use cases – User Interface, RESTful Service and Web Scraper as discussed below (figure 1)



**Figure 1: Behavioral Model of the System**

a) Client’s User Interface: User interface of this system is the client side web browser that provides facilitation to interact with the user by getting inputs and providing outputs. So by means of UI, users will provide essential information for the system in order to achieve desired work. User Interface use case has two INCLUDE components. One component for getting above mentioned user input and another component is a view provided to display the output content.

b) Web Scraper :Web scraping (web harvesting or web data extraction) is data scraping used for extracting data from websites. It is a form of copying, in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, which in our case is files storage for later retrieval or analysis. In our system, web scrapping is done on Wikipedia events data and google image data.

c) Service Provider: This is basically anApacheTomcatServer which is a Java Servlet Container that provides a pure java HTTP server of restful web service environment. Java program is written for implementing the server service such as Randomized feature generator

and Image manipulator. Sentimental analysis is done to extract user sentiments from the sentence to capture the user's mood. It also has a data structure, which is designed in a way to store data residing in files, for easy access by Key: Value pairs and also ensures the data availability within the server all time, thus reducing the data retrieval time from database. Once the server starts running, it collects the data from files, which is collected by the web scraping component from web pages and server stores them into designated data structure written on server.

*B. Interaction Model:* On the server side, before running the server, web scraping is done to retrieve contents related to all 365 days on the basis of several events and it is stored in the form of files. Server reads all the files and stores it in data structure that is appropriate for fetching information instantaneously.

User views the front page of the application in the browser and it is an input form component that prompts the user to enter the details (figure 2).

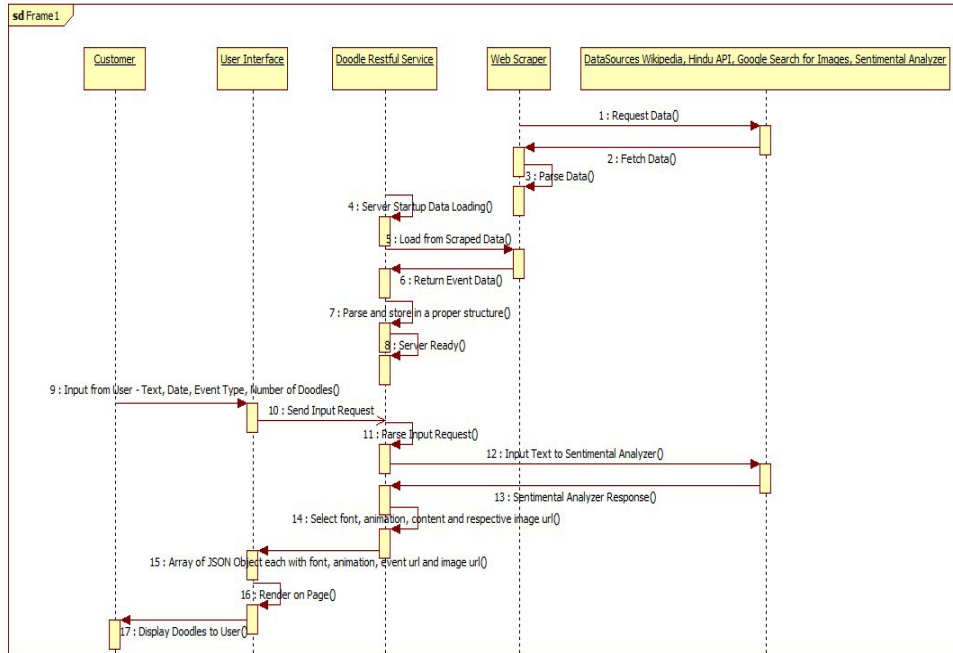
The user of the system has to provide the following inputs:

- User entered text
- Date and month (Generic dates or current updates)
- Specify category of events
- Number of doodles, user wants to receive as output

User is required to fill all the inputs. Once completion (hit button) is done, an ajax call is made to the server, which in turn returns an axios promise object. The returned JSON object has randomly generated properties like fonts and animation styles along with content related image link. The JSON object consists of 'n' number of objects which is a user specified number from the browser. So 'n' number of objects with 'n' number of randomly generated properties are iterated through and apply on the doodle component and displayed in the front end. Figure 2 models the sequence of interactions in the proposed system.

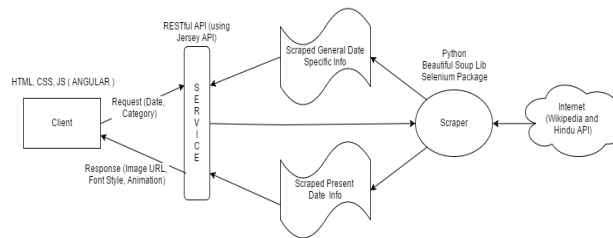
*C. Overall System Architecture:* The information about each of the contents that have potential to produce the doodle is scraped out from Wikipedia and Hindu API. Each event has a link to the event that provides some information about it and an image link depicting the event.

The Server is loaded with the scraped information during the start-up of the Server. It provides a REST API post for the client to call to grab information for constructing the doodle.



**Figure 2: Sequence Model of the System**

The Client obtains input parameters from the user. The client makes a call to the REST API provided by the Server, with the parameters obtained from the user. The server responds with the information based on the input parameters which is then rendered by the client.



**Figure 3: Overall System Architecture**

## 5 System Implementation

React JS, Eclipse Java EE and Python. React JS has been used to build the UI for the System and make requests to the backend for fetching the appropriate data based on the input provided by the user. Java has been used here to implement the server side code, which based on the user inputs provided by the REST POST API call, responds with the appropriate

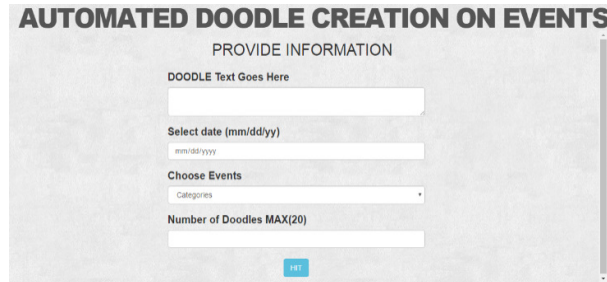
information for building the Doodles. Python has been used to build the Dataset that is required for attaching to each doodle by scraping data from Wikipedia and Hindu API. The implementation procedure is discussed in this section.

#### *Client Side Implementation*

React JS technology deals with creating view of the browser in the form of components. In our project, we use browser as a tool to showcase the doodles generated. So each view in the browser is created in the form of components as follows:

a) Input Component: The default main page is set to be this input component inside which the user finds to enter the required details such as date and the doodle text using which doodle is created. This page is created using HTML and materialized using styling classes made available by Google.

Input is entered using form elements (figure 4). Form validation is done to check whether the user enters the data in all the input element and if not, then appropriate error messages are shown. After all details are entered and HIT button is clicked, the flow is behind the scenes. On the click of hit button, an AJAX request is made to the server which is a post call that carries all the data that user has entered.



**Figure 4. Default User Input Page**

The server sends a JSON object as response to the client. Axios is used for making ajax calls, it is made to return an axios promise object for asynchronous purpose. This axios converts the response from the server to JSON object, even if initially the response is not in JSON format

From the React-Redux perspective, this ajax call is made by dispatching an action that returns an object (for ajax request, it's an axios promise object) to the reducer that is associated with the data returned from server and is stored in the global store. Now the global store has the JSON response that contains the properties that is to be applied on the doodle text like animations, font styles that are randomly generated and the image associated with the user selected event.

All the properties corresponding for each object is iterated through the JSON array and applied on the Doodle Component dynamically.

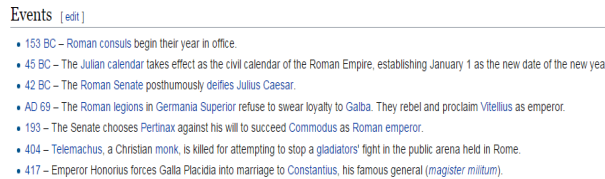
b) Doodle List and Doodle Component: Doodle list generates a list of 'n' number of doodle components by iterating through the state data in the store, which is the returned Json from server. While instantiating doodle component, the doodle list passes the current iterated object that contains the properties required for that current doodle.

Doodle component consists the layout for Doodles written in HTML in order to view it in the browser. User can also click on each of the doodles that is shown. Further, the user can change the properties of that particular doodle.

c) Data Collection by Web Scraping: Wikipedia has been used as the main source of the information to get contents related to the dates based on several categories like Events, Births,



Deaths and Holidays and Occurrences. It provides date-wise information under each of the categories. The contents under each of the categories are available in sorted order of dates. This can be used as supportive information to provide the users with priority-based information, whether old or new contents are important to the user. A sample of the same is given in figure 5.



**Figure 5: Wikipedia – related information**

Another online API related to the current proceedings is Hindu API. It provides trending news that can add better value when an event relating to the present scenario has to be highlighted to the people.

```
{
  "author": "Internet Desk",
  "title": "Indian shot dead in Washington robbery",
  "description": "Indian Consulate in San Francisco is helping the family of VikramJaryal, says SushmaSwaraj",
  "url": "http://www.thehindu.com/news/national/26-year-old-indian-shot-dead-in-washington-sushma-swaraj/article17891982.ece",
  "urlToImage": "http://www.thehindu.com/news/international/article17892001.ece/ALTERNATES/LANDSCAPE_615/MakeAmeriaSafeAgainAFPjpg",
  "publishedAt": "2017-04-08T06:59:55Z"
}
```

d) Scraping Data: Data is scraped from Wikipedia pages and from Hindu. The procedure for the same is discussed in this section.

Data scraping from Wikipedia involves obtaining the webpage related to each of the dates and then obtain the elements of interest from those pages. The html document in Wikipedia has been divided into sections like Introduction, History, Events, Deaths, Births and Holidays\_and\_Occurrences. The beautiful soup package in Python that has been already discussed above is used to easily identify portions that are required. Each of the categories is an unordered list identified by <ul> tag. The list has elements each identified by the <li> tag. So, the beautiful soup conceptually identifies the first unordered list under the events section and then propagates to find all the elements in the list. The description of each of the items in the list is obtained and also the respective links, which is required for redirecting the user to the content based on which the doodle has been built.

```
{
  "April_1": {
    "Births": [
      {
        "desc": "1220 \u2013 Emperor Go-Saga of Japan (d. 1272)",
        "link": "/wiki/Emperor_Go-Saga"
      },
      {
        "desc": "1282 \u2013 Louis IV, Holy Roman Emperor (d. 1347)",

```

```

"link": "/wiki/Louis_IV,_Holy_Roman_Emperor"
},...
]
"Events": [
...
],
"Deaths": [
...
],
"Holidays_and_Occurrences": [
...
]
},
"April_2": {
...
}
}

```

The information in JSON format for all the days in a year is stored in a file. The data from this file is provided as input to the image retrieving system.

The HTTP request to the Hindu API provides us with the data in the JSON format. The appropriate information like title, url and urlToImage are stored in a file for fetching images similar to the case from Wikipedia.

```

[
  {
    "desc": "6.5% turnout in violence-marred Srinagar bypoll",
    "image": "http://www.thehindu.com/news/national/article17896994.ece/ALTERNATES/LANDSCAPE_615/VBKBSF",
    "link": "http://www.thehindu.com/news/national/polling-for-srinagar-lok-sabha-seat-begins/article17896913.ece"
  },
  {
    "desc": "36 killed in Egypt Palm Sunday blasts",
    "image": "http://www.thehindu.com/news/international/article17897457.ece/ALTERNATES/LANDSCAPE_615/EGYPT-VIOLENCE-CHURCHjpg",
    "link": "http://www.thehindu.com/news/international/many-killed-in-palm-sunday-blasts-in-egypt/article17897158.ece"
  },
  ...
]

```

e) Fetching Image Links from Google: Image links for each of the data item fetched from Google is collected from google. This process is automated using Beautiful Soup package. The search for image process is automated by a python script, which constructs a URL out of the search term. This produces the image results page. Google's image search engine brings out the closest matching images relating to the provided description about the event. The first five image links are extracted out using the unique html template ids and classes. Beautiful soup has options for finding all the tags with the specific ids and classes.

```
{
```

```

    "April_1": {
      "Births": [
        {
          "desc": "1220 \u2013 Emperor Go-Saga of Japan (d. 1272)",
          "links":
            "[ 'https://upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Emperor_Go-Saga.jpg/220px-Emperor_Go-Saga.jpg',
              'https://d1k5w7mbrh6vq5.cloudfront.net/images/cache/27/9b/05/279b05ea695f81d9b840f56e6488f2eb.jpg',
              'https://d1k5w7mbrh6vq5.cloudfront.net/images/cache/14/c2/95/14c295777ff5190c193eeb53dbf60569.jpg',
              'https://d1k5w7mbrh6vq5.cloudfront.net/images/cache/ba/b1/26/bab1261b0ed168fa40f19d066d0e724d.jpg',
              'https://d1k5w7mbrh6vq5.cloudfront.net/images/cache/3b/4e/3f/3b4e3f2e9d161265fb34474a86f9f4ce.jpg']"
        }, ...
      ]...
    }...
  }

```

f) Server Side Implementation: Server Side has been implemented using Java. Tomcat has been used as the server for hosting the application. The application is exported as a web WAR file. This is deployed on the server and can be viewed from the localhost:portNumber of the system. The server before starting runs a startup code. This consists of the information for parsing the JSON object from the collected data and store it as an in-memory structure for serving the customer with necessary details for doodle generation.

g) Doodle API Endpoint – POST Request: The Doodle API service serves as the endpoint for communication for the REST Clients using the service. The client implemented using React makes an AJAX call to the Restful Doodle service at the backend. The implementation is a POST that prevents the data from being passed as URL parameters and acts as a more secure means of communication. The actionable items like the content text, the date information, the type of event and the number of doodles to be generated are passed to the service via the post body. A POJO (Plain Old Java Object) Object has parameters and getters and setters that maps to the parameters passed in the body text of the post call.

h) Doodle Input Request and Output Response Processing: The input parameters from the POJO object is processed to provide the necessary output for doodle creation.

The input text is passed to the sentimental analysis API which provides the sentiments of the input text as ‘Positive’, ‘Negative’ or ‘Neutral’. This information is then used to determine the font, animation type and the type of event. These fonts, animations and event types are tagged as whether positive, negative or neutral in prior by manual examination. This is similar to adding details for the items in a catalog.

The date provided by the date picker is to be parsed to map to the stored data format in the background collected data. This is then used to select the events belonging to the particular date.

The event parameter from the dropdown list is used to select from one of the items in the list of items available for the particular date and event.

The number of doodles determines the number of objects in the result array to be passed as result to the client for the post call made. These items like fonts, animations, image link and content link are passed to the client encapsulated inside a JSON Object. A JSON Array whose

size equals the number of doodles parameter is the response to the system. These are then rendered based on the matching field names in the doodle class and react client.

## 5 Results

The input screen specifies doodle text, date, connected events and the number of doodles to be generated. If no specific event is related for the text “PSG TECH” the following doodles are generated (figure 6)



**Figure 6: Generic doodles for the text “PSG TECH”**

If the event is selected as Valentine’s Day, then the following doodle (figure 7) is generated.



**Figure 7: Event based doodle for the text “PSG TECH”**

Doodles can be customised using various settings as shown in figure 8



**Figure 8: Customisation of Doodles after generation**

## 6 Conclusion

A novel approach for personalizing doodles has been proposed in this project. The customer is provided with emotions to share, knowledge to learn and curiosity for more of it. The Doodles here represented modifications in terms of the content, layout, fonts and animations. Deeper research into this subject can lead to more creative doodles. Future work may include use of custom objects to represent the characters in the doodle text, or custom objects to represent portions of those characters. The custom objects must belong to the context of the images on top of which they will be presented. This, as an initial implementation can be done by grouping all the possible images in the universe into few groups based on the content of the images. The pre-tagged custom objects relating to the context from image can then be used to build the doodles. The event to be picked can be mapped to the doodle text given by the user. This may require grouping the doodle text and text for each content into categories and then finding an intersection between the two. More layouts which are dynamic in nature can be defined. Future technologies to abstract the features from one image and apply it on another, will be of potential help to this project. Also, sentimental analysis on the images can allow for better mapping of the doodle text to the events.

## References

- [1] What tools does Google use to create Doodles, <https://www.quora.com/What-technology-is-used-to-design-Google-Doodles>, accessed on 19th Oct 2021.
- [2] How Google Doodles are made, <https://www.theatlantic.com/technology/archive/2011/1/how-google-doodles-are-made/334909/>, accessed on 10th Oct 2021.
- [3] The logic behind Google doodles, <https://www.pnclogos.com/the-logic-behind-the-google-doodles/>, accessed on 12th Oct 2021.
- [4] How do Google Doodles work?, <https://zoompf.com/blog/2012/04/how-do-google-animated-doodles-work>, accessed on 09th Oct 2021
- [5] The best art and design that Gogle Doodles ever Created, <http://www.complex.com/style/2013/07/google-doodle-art/will-eisner>, accessed on 09th Oct 2021
- [6] First artificial intelligence Google Doodle features Bach, <https://www.google.com/doodles/celebrating-johann-sebastian-bach>, accessed on 10th Oct 2021.

- [7] D. S. Vijayan, A. Leema Rose, S. Arvindan, J. Revathy, C. Amuthadevi, "Automation systems in smart buildings: a review", *Journal of Ambient Intelligence and Humanized Computing* <https://doi.org/10.1007/s12652-020-02666-9>
- [8] Amutha devi, Gayathri Monicka.J, "Emerging bio-medical applications and open research challenges in cognitive internet of things (CIOT)" *International Journal of Pharmacy and Technology* Dec-2016 Vol. 8 Issue No.4 5049- 5054
- [9] Gayathri Monicka, J & Jamuna, V 2015, 'Hybrid Cascaded MLI topology using Ternary Voltage Progression Technique with Multicarrier Strategy', *Journal of Electrical Engineering &Technology (JEET)* vol.10,pp.1610-1620,2015