

---

# **PROJECT ZOOP**

2ID60 Web Technology Report

---

## **Group aa..uu..h...**

Bartosz Bąk (0995912)

João Ferreira (1040756)

Rocco Fortuna (1013054)

Ameya Rao (0993084)

January 15, 2018

# Contents

General . . . . .	2
About . . . . .	2
Introduction . . . . .	2
What is Zoop? . . . . .	2
Hosting . . . . .	3
GitHub Repository . . . . .	3
Browser Testing . . . . .	3
Work Distribution . . . . .	3
Frontend Design . . . . .	3
Backend Design . . . . .	4
Databases . . . . .	4
REST APIs . . . . .	5
Video . . . . .	6
Reflection . . . . .	6

## GENERAL

### About

#### Introduction

In the age of instant communication that the internet has brought upon us, social media plays a key role in the daily life of millions of people. However, a big share of what is communicated consists of condensed descriptions of our experiences. We as humans, as social creatures, have the urge to share the underlying feelings in order to make friends and our loved ones, as well as the society around us, partly live through these meaningful experiences with us. Although words represent an extremely powerful tool to express and explain concepts and ideas, they do have a limit. There are some feelings that can only be conveyed through body language, facial expressions combined with images or objects. All this can be condensed to one and only one word: Emoji. Emojis are undisputedly the language of the youth. To express that one feeling that a video, image or reaction evoked, through emojis, is an art that youngsters have the privilege to own. Emojis are genuine, clear and over all quick; they can bring the message across without diluting its effectiveness. In this new fast-developing world there is no time to waste on writing meaningless propositions. We, **aa..uu..uh...** believe that Zoop could be the next big thing.

#### What is Zoop?

The web application we have created is a smart and functional social media platform featuring the most up to date list of the trendiest Emojis on the web. Emojis are mandatory: a post cannot be published if it does not contain enough Emojis. The users will be able to post so called 'Zoops', to repost other users' Zoops and react on those by means of an Emoji chosen from a short given list. Both the description and the comments must contain Emojis. Users can search and decide to follow each other to have their zoops feature on in their feed. A functional Emoji keyboard will help the users in their tasks and shortcuts can be used starting with a colon to find the desired Emoji faster by searching for its name. Users can even personalize their profile picture uploading whatever they please. All these features give life to a whole new unique virtual universe that will host a dynamic and energetic community ready to experience the future of communication.

## Hosting

The web application 'Zoop' is hosted on the following link: "<https://project-zoop.appspot.com>"

## GitHub Repository

<https://github.com/AmeyaRao98/ProjectZoop>

## Browser Testing

Zoop was tested and works on the latest versions of Chrome, Firefox, Opera and Internet Explorer 11. Safari is not updated enough to support our ambitious web application. It took a lot of effort to minimise the issue of different browsers not supporting certain emojis and different functionalities, so we just recommend the users to use a decent up-to-date browser.

## Work Distribution

While some were better at handling the backend, others looked at the front end features such as the emoji input, while the report was worked on by everyone so that we offered a balanced perspective on the entire assignment.

# FRONT-END DESIGN

## Design

The front-end design of the web application was started up by means of a Bootstrap template that was adapted to the purpose to provide structure and organization. We opted for functionality over aesthetic appeal with our design, as we wanted to make it as user-friendly as possible. Interesting and appealing combinations of colours were chosen in order to make the page more coherent and pleasing to the eye. In particular, we made sure to consistently include shades of yellow to reflect the theme of emojis.

Bootstrap was used for various aspects such as buttons and sizing, so that the website easily adapts to different screen sizes, resulting in a completely responsive web application. JQuery was used in the development of the interactions in the front-end, such as fade-in functions and other stylish features. At some points the front-end design may seem slightly lacklustre, but that is due to the intense work the back-end design required in order to make all the

interactions work smoothly.

In order to type out a Zoop, an Emoji input field is used, which makes use of the following: <https://github.com/mervick/emojionearea>. This allows users to quickly and efficiently type out Emojis. The entire list can be opened by either pressing TAB when the cursor is in the field or by clicking the Emoji at the upper right corner of the field. Alternatively, users can type ':' and an autocomplete suggestion list appears based on what is typed, searching through the names of the Emojis.

## BACK-END DESIGN

The backend used in this app is Django. The main reason for this choice is that members of the group were already familiar with it and because it offers great functionality. We extended it with the python Emoji package to enhance Emoji handling in the code, the Pillow package used in the implementation of the avatar system and djangoRESTframework to provide our REST API. The back-end is pretty robust. It was our main focus, because we wanted our app to be as interactive as possible. We managed to create a well constructed post system in which the user can react to a post or rezoop it (add it to their own timeline). Users are also able to follow specific users to view a personalized timeline consisting of posts created only by those personally followed users. Because of the time constraint, administration system is not implemented in django, but as proven throughout the development, all administrative tasks can be accomplished by a direct connection to the database by means of pgAdmin utility shipped with postgres installation.

## DATABASES

The entire development stage was done in Django using the provided SQLite database, but then we migrated to postgresSQL database because it is supported by Google App engine which we are using to host the app. we connect to it using configuration settings provided by Google in `projectzooop/settings.py` and `app.yaml`

The following are stored:

**Users:** Users can create an account and login. After logging in, they can share zoops, rezoop other people's zoops, follow other people and react to other zoops. A sophisticated recovery system is also in place. Users can reset their password if they forgot it or change it to something else through their account page. **Post:** A foreign key to the user. It has an ID, content, timestamp and the original poster, since it is possible to rezoop a zoop.

**PostReaction:** A foreign key to a user and a post. A user can choose one of six reactions to a post.

**UserDetails:** A OneToOneField with the user. In addition to the regular details Django stores, it also stores an avatar. The system worked while in development environment but we couldn't make it work with the Google Cloud engine.

**Following:** A ForeignKey to users that stores all the users followed by a particular user.

## REST APIs

We don't pull data from any API, as the entire content is generated by users due to the fact that is a social media platform and all data that is displayed and used is user generated.

We provide our own API call in order to let users retrieve posts by user's id:

Usage:

GET: /posts/

parameters:

user (default=None)

count (default=1) (Max = 30)

Example usage:

GET <http://project-zoop.appspot.com/posts/?user=1&count=2>

Response:

HTTP 200 OK

Allow: GET, POST, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
[
  {
    "user": 1,
    "post_id": 3,
    "content": "",
    "timestamp": "2018-01-15T15:04:56.393920+01:00",
    "original_poster": 1
  }
]
```

Note that request only returned 1 post because the user has only 1 post. The content in response is not empty, it is latex that can't render emojis.

## VIDEO

A video explaining how the website functions can be found on the following link:

<https://youtu.be/hEUDN01o16s>

## REFLECTION

Besides learning a lot about Django and technologies used in the development of a web app, we learned that it is important to plan well for projects of this magnitude. Things can quickly become hard to manage without proper organisation. While we are very pleased with the result, we did have to cut out a few features and put less effort into our front-end due to the fact that the majority of the work was done close to the deadline.

If we could start over, we would definitely have done more work before the deadline so that we would be less stressed when the deadline approached. We had a good team that worked very well with each other. Looking back, it may have been more fair to work on the backend together as opposed to completely separating the frontend and backend tasks. Overall, however, this has been a great learning experience and we are glad that we were able to create something that we are proud of.