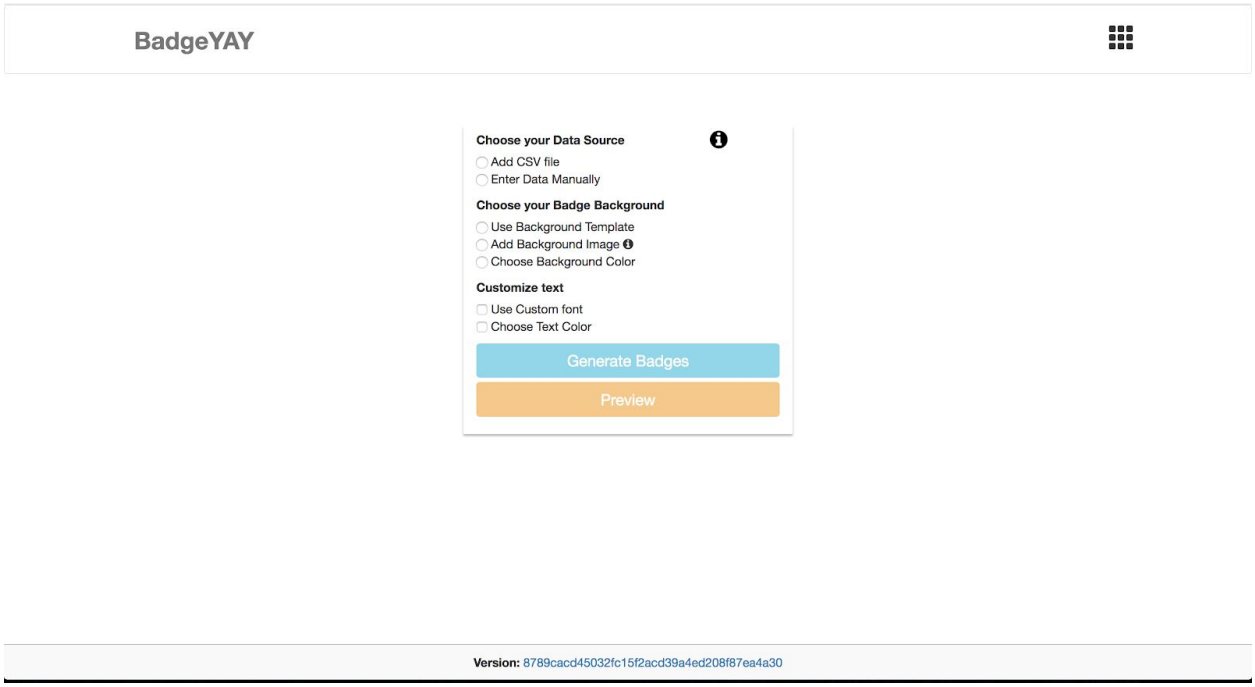


Badgeyay into full featured suite

By Vaibhav Singh

@vaibhavsingh97

Org: FOSSASIA

Overview

This is a project is a simple badge generator that generates different badges for events like conferences, summit, meetups etc. As badges are an essential items for such events so this project serves to help these events.

Problem Statement

Badge generation is the hectic task for the organization who is organization organizing conference, summit, meetup,etc. Although, paid webapps are there but it's too expensive.

Current Status Of Project

The project is currently in a functional state but still there are some bugs in the project which needed to be fixed. This project is also an extension to the EventYay where people can create and maintain events. The project contain lot's of unnecessary code and the project also lacks some basic badge generator functionality (some of which are mentioned in implementation below) which end-user might expect.

Currently user can:

- Choose background template, background image and background color.
- Choose different fonts and different font color
- Preview badge before generating actual badge
- Generate badge

Goals

My goal is to enhance the current version of Badgeyay by adding functionalities and features for end users so that our badgeyay can widely be used as an open source solution to generate badges in events, conferences, etc.

1. Badgeyay App (Main Functionality of the app)

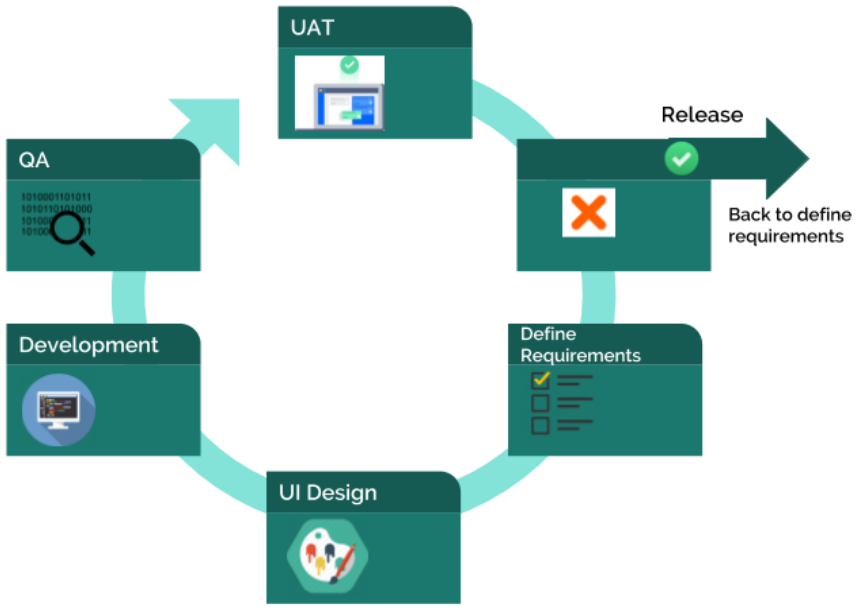
a. Improve the badgeyay UI
b. Improve the BadgeYay API and resolved the known bugs
c. Can produce High quality PDFs with the user configuration like different Paper Size and Badge Size.
d. User can upload custom background and after successfully uploaded, if background is not according to guidelines than app should provide user feedback so that user can re-upload badge.
e. Provide simple d*rag and drop* interface and can be customize according to user's need.
f. Improve the Badgeyay UI

2. Enhancement of Badgeyay
    a. User can customize each and every section like Name, Organization, social handle, etc
        i. User can change font of all of these fields
        ii. User can customize the font like making bold, italic, underline, change color
    b. User can add event logo and can change position for the same
    c. User can either contact details either by adding barcode or QR code
    d. User can choose between different size of badges or also can choose custom size of badges.
3. Authentication
    a. Adding OAuth 2 as an authorization framework so that the app can have limited access to the user account on an HTTP service, such as Facebook, Github, etc. and user can save their badges configurations
    b. User can authenticate and save their current badge configuration
    c. User can visit their profile and access the previous badges made by them and able to regenerate

4. Theme for Badges
    a. User can add different themes which will be available for different users globally. User have the choice to make their badge theme public/ private.
5. Analytics in Badge
    a. Basic analytics in backend such as number of badges generated, number of unique user used our service etc
6. Test Cases
    a. Adding test cases with each enhancement and bug fix. This will make sure our code doesn't break any point of time.

Project Development Lifecycle

I will follow agile methodology for Badgeyay development. The iterative process is dominated by Agile software development life cycle. The result of each iteration in the next piece of software development process- the software that supports working and supporting elements such as documentation, available for use by customers - until the final product is completed.



A typical process flow can be visualized as follow:

1. **Requirements:** Defining the requirements with discussion with the community and mentors to make badgeyay stable and working.
2. **Development:** Design and develop software based on the requirements
3. **Testing:** Continuous testing of the small patch by writing test cases and user testing (which is done by mentor and community)
4. **Delivery:** Integrate into the master codebase and working on next iteration
5. **Feedback:** Record feedback from the mentor and community and improve in the next iteration of the project.

Implementation Details

1. Code refactoring

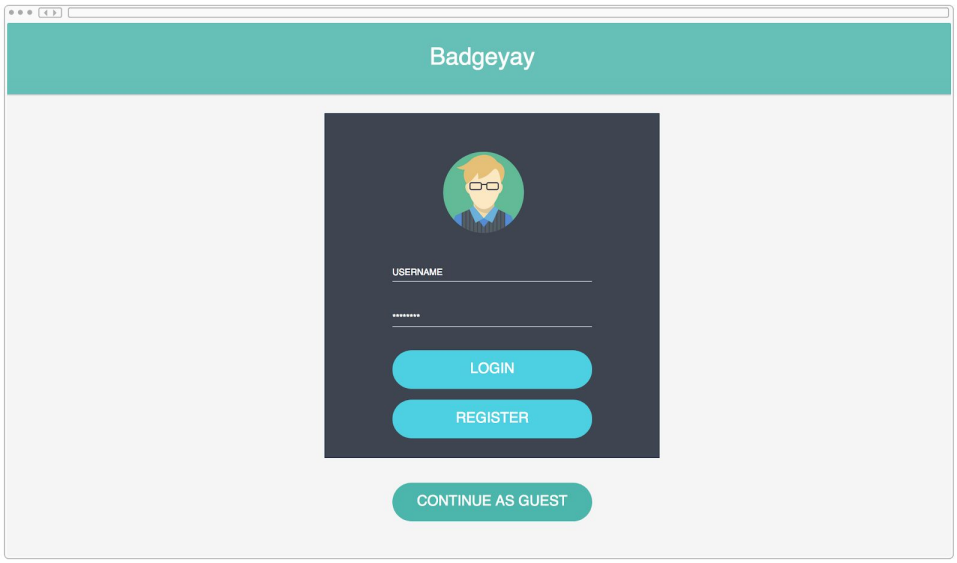
- Solving existing bugs in the latest version of the app and ensure the basic functionality of badgeyay is working.
- Removing unnecessary code from the codebase. Currently, Badgeyay repo contains some code which is either deprecated or not used anymore
- Add test cases after solving existing bugs

2. Improve UI/UX

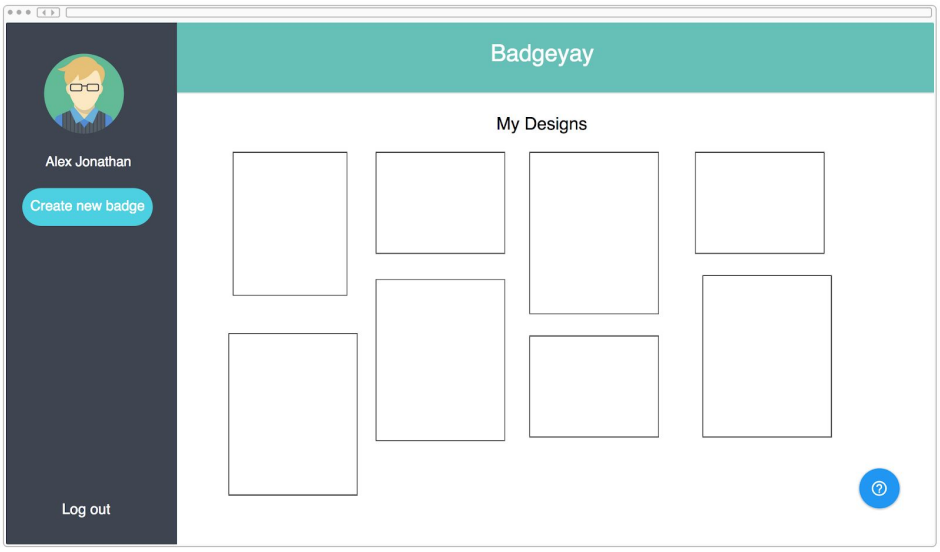
Current design of Badgeyay is not User centric design. User centric designs consists of two important parts, i.e., usability testing which is based on goal users' psychological research (user model, user needs, procedures etc.). Second, in the behavior of the design, it is cleverly used to combine cognitive psychology, ergonomics, industry psychology and other topics of basic principles. Here I have added some of my proposed designs to make it User Centric Design.

1. User can login/ register to the badgeyay app, and can design badges. Registered users can generate badges and their badges configuration will be saved so user can revisit anytime to get their badges printed.

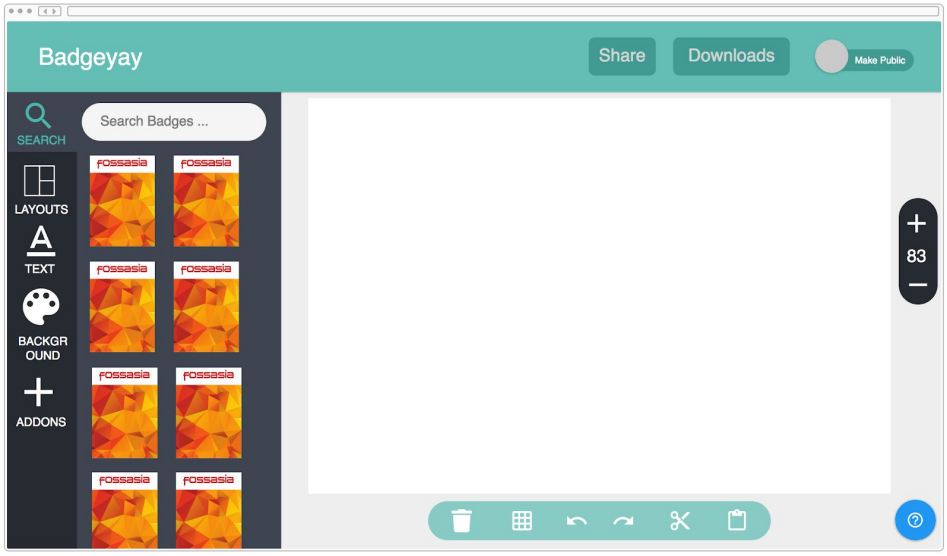
User can also use our system without login, it's just they can't use some of the premium feature which will be available after registering with us.



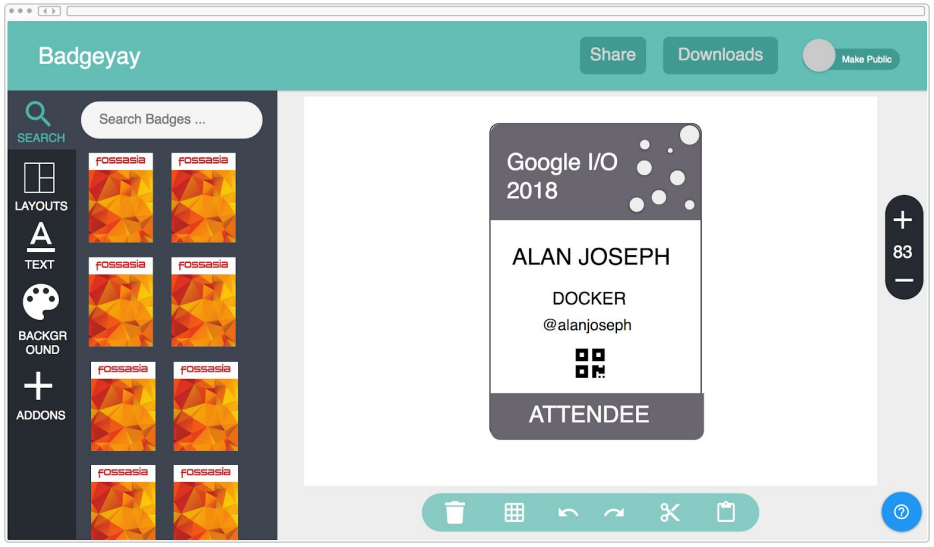
2. On successful login, User will able to see the simple dashboard which consists of simple design made by them. (This can be complete badges/ incomplete badges).



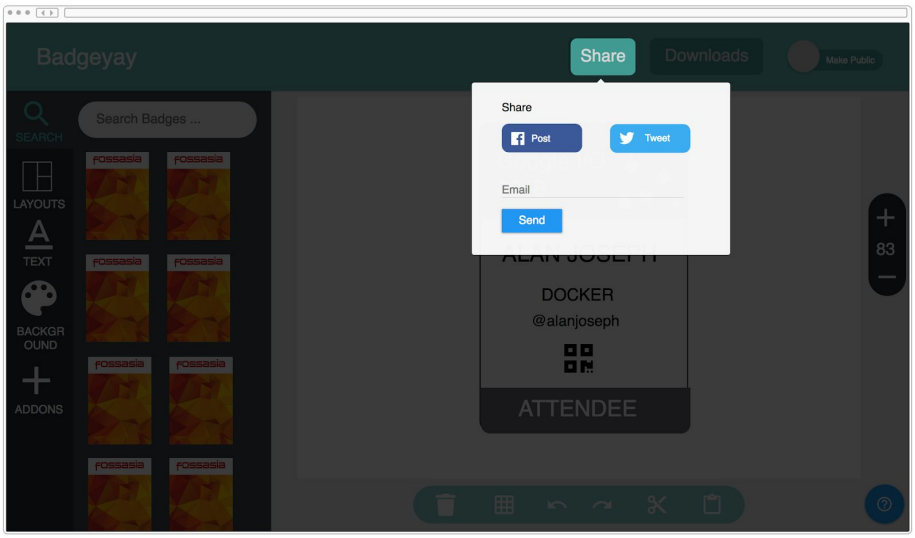
3. After clicking on ***Create new badge*** , User will be landed to the designing area. User can either search pre-made badges created by us to start with or can create entire new design.



4. After designing badge. It will look something like this.



5. User can also share their designs on social platform and encourage community to use our platform to generate badges.



3. Adding Attendee/Speaker contact details on the badge either by adding QR Code/ Barcode

Generally, people in events like conferences/ meetups came for learning new technologies and make networking. This feature in badgeyay app will help the organizers to ease the process of attendees, speakers, etc. QR code/ Barcode will will store basic contact details like his name, working area, company, social handles, etc given at the time of registration for the event. This feature will be in closed integration with the eventyay and hence, we can fetch details from the eventyay for details to be included in the qr code or organizer can add these details in the CSV itself and Badgeyay app will add QR code/Barcode according to the given detail.

```
import qrcode
qr = qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_L,
    box_size=10,
    border=4,
)
qr.add_data('Some data')
qr.make(fit=True)

img = qr.make_image()
```

In this way attendee/speaker either scan the QR code/ barcode to save details or can capture photo and can extract details later.

```
import qrcode
import qrcode.image.svg

if method == 'basic':
    # Simple factory, just a set of rects.
    factory = qrcode.image.svg.SvgImage
elif method == 'fragment':
    # Fragment factory (also just a set of rects)
    factory = qrcode.image.svg.SvgFragmentImage
else:
    # Combined path factory, fixes white space that may occur when zooming
    factory = qrcode.image.svg.SvgPathImage

img = qrcode.make('Some data here', image_factory=factory)
```

For adding barcode, we can use the following code snippets.

```
import barcode
from barcode.writer import ImageWriter
ean = barcode.get('ean13', '123456789102', writer=ImageWriter())
filename = ean.save('ean13')
```

4. Tackle long names

The next problem arises with the current version of Badgeyay is that when we have to generate badges with long names than those name simply overflow. The possible solution can be we can add their name with short forms. We can use ***python-nameparser.*** It's a simple python module for parsing human names into their individual components. We will then either make first name and middle name as initials and can add surname to shorten the name.

```
from nameparser import HumanName
name = HumanName("Dr. Juan Q. Xavier de la Vega III (Doc Vega)")
print(name)
'''
===== Output=====
<HumanName : [
    title: 'Dr.'
    first: 'Juan'
    middle: 'Q. Xavier'
    last: 'de la Vega'
    suffix: 'III'
    nickname: 'Doc Vega'
]>
================
'''
print(name.last)
# 'de la Vega'
print(name.as_dict())
# {'last': 'de la Vega', 'suffix': 'III', 'title': 'Dr.', 'middle': 'Q. Xavier', 'nickname': 'Doc Vega', 'first': 'Juan'}
print(str(name))
# 'Dr. Juan Q. Xavier de la Vega III (Doc Vega)'
name.string_format = "{first} {last}"
print(str(name))
# 'Juan de la Vega'
name.string_format = "{first[0]} {last[0]}"
# 'J d'
```

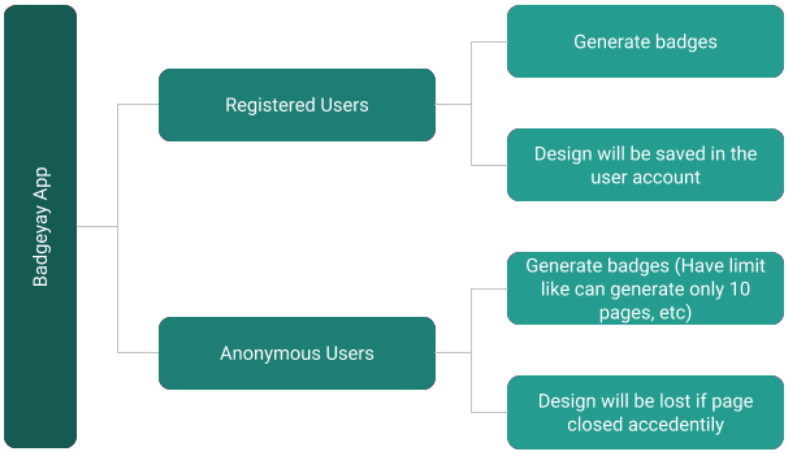
5. Anonymous User

We can have one extra method in the *User* class which will tell us that user is anonymous or authenticated. Also, if it returns True than we can block some features in the web app like user will lost all the configuration as soon as he/she exits the session, etc.

```
from flask.ext.sqlalchemy import SQLAlchemy

db = SQLAlchemy()

class User(db.Model):

    __tablename__ = 'user'

    email = db.Column(db.String, primary_key=True)
    password = db.Column(db.String)
    authenticated = db.Column(db.Boolean, default=False)

    def is_active(self):
        """True, as all users are active."""
        return True

    def get_id(self):
        """Return the email address"""
        return self.email

    def is_authenticated(self):
        """Return True if the user is authenticated."""
        return self.authenticated

    def is_anonymous(self):
        """True for anonymous users"""
        return False
```

6. Basic Badgeyay System

Badgeyay app will allow registered users to generate badges and they can access previously generated badges. All the badges will be shown on the main dashboard under "My Designs" page. Anonymous users can generate badges but only with limited pages, for unlimited badge generation they have to register in our system. Also, design will be lost if they close the page accidently. Hence, no recovery is possible.



7. Badge storage

All the badges will be stored in the Postgres. My reason to choose Postgres because PostgreSQL's query optimizer is superior to many others. It has built-in NoSQL key-value store. Also, It has built-in binary JSON storage like Mongo, Good functions for storage and manipulation, Syntactic sugars for querying, cloud-hosting with extra features like data clips, snapshots, etc. These features help us to built extra features like recovering of designs made by user if accidently closed, etc. Also, each user badge designs will be stored in the database with some extra metadata like time of file creation, type of badge, Type of event, etc. This metadata will help us in

developing search functionality in case the user choose to make their badge design public then we can use this metadata to index and respond to user search queries.

8. Adding API which can be integrated to eventyay

As badgeyay is extension of eventyay, We will incorporate API using python-restful and python-flask, to gather event data and details of attendees to create badges for them. Hence, Badgeyay will help eventyay to generate badges automatically.

Designing GET, POST, OPTIONS functions which will help eventyay in generating badges for an event.

9. Adding Test Cases

Testing for the badgeyay app will be done using Unit Test approach. We can use either of the option for adding test cases for backend.

- Unittest Python module
- Assertions and Exceptions for Python

```
from unittest import TestCase
from coloredlogs import install
from psycopg2 import connect
class BaseSettings( TestCase ):
db = None
@classmethod
def setUpClass(cls):
    install( level='DEBUG' )
    host = 'localhost'
    cls.ApiUrl = "http://localhost:8000/"
    # database connection credentials
    cls.db = connect( host=host,
                      user='username',
                      password='password',
                      database='password' )
@classmethod
def tearDownClass(cls):
    cls.db.close( )
    print("------test is over------")
```

For frontend, Testing is made particularly in Ember through its use of the ember generate acceptance-test and QUnit

Project Timeline

Google Summer of Code is a 12 week long program. I have divided my proposal into three milestones and my main focus is on adding functionality and improving backend without breaking the functionality of the app. Along with the coding, I will be writing weekly blogs and daily scrum to update my progress throughout the program.

Below is the timeline that has been prepared for this GSoC 2018

Time	Work
24 April 2018 - **13** May 2018	**Community Bonding Period** • Discussing UI with mentor and improve the current UI according to feedback • Lock final requirements after discussion with mentors • Lock the wireframes and learn new technology if needed • Learning more about ember JS
7 May 2018 - **20** May 2018	• Fix bugs in current version and make it stable. • Fix any UI bug, and make snapshot and deploy on Heroku • Adding simple authentication in the backend.
14 May 2018 - **28** May 2018	**Coding Period** • Refactoring the current backend code and make it modular
29 May 2018 - **11** June 2018	

	• Add dummy UI according to the final wireframes • Adding different models in database • Adding feature of generating different size badges • Add feedback from mentors • Fix Bugs
Extra Deliverables	
Blog Post about my journey every week Daily Scrum in the morning	
16 June 2018 - **28** June 2018	• Add option to customize Name, Organization, social Handle • Add feature to add the custom logo • Add features to customize position of logo • Add QR code to the badges where information like Name, social handle of person will be stored • Add documentation in the project • Add feedback from mentors • Fix Bugs
29 June 2018 - **9** July 2018	• Modifying the ember JS frontend fro login • Add Oauth modules in the project • Add documentation in the project • Adding test cases for backend • Add feedback from mentors • Fix Bugs
Extra Deliverables	

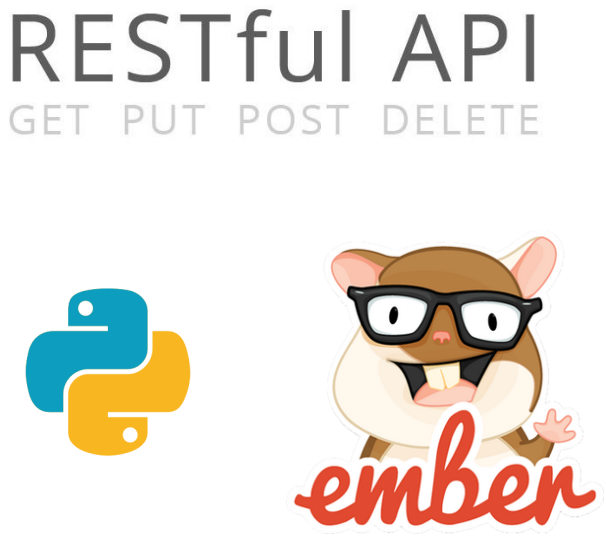
<table>
<tr><td colspan="2">Blog Post about my journey every week
Daily Scrum in the morning</td></tr>
<tr><td>14 July 2018 - 23 August 2018</td><td>Separating Registered users and anonymous users
Add option to save current workbook configuration so that user can redo and undo
Add option to save user badges in databases
Add option to make badges public, this will help users to design badges and use their badges design for inspiration</td></tr>
<tr><td>24 July 2018 - 6 August 2018</td><td>Add basic analytics in project like number of badges generated etc
Add Test cases and documentation
Fix existing bugs</td></tr>
<tr><td colspan="2">Extra Deliverables</td></tr>
<tr><td colspan="2">Blog Post about my journey every week
Daily Scrum in the morning</td></tr>
<tr><td>6 August 2018 - 14 August 2018</td><td>Adding Test Cases
Adding documentation
Adding scripts to prevent from crashing and look for solution for timely snapshot</td></tr>
<tr><td colspan="2">After GSoC 2018</td></tr>
<tr><td colspan="2">Continue to contribute
Help the organization by acting as an mentor to guide newcomer</td></tr>
</table>

Contribute as an mentor in codeheat Represent projects in conferences and meetups Help people to contribute in FOSSASIA and invite them to FOSSASIA

I will devote 40-50 hours per week during GSOC.

Languages and Technologies

In this project, we will have frontend and backend, so for the development of this project we will be using Python-FLASK to enhance the REST API for the backend, It will also be used for communicating with the Database Service. I will use PostgreSQL as our Database Service due to its flexibility and ease-of-scaling



For front end, we will be using EmberJS along with AJAX to communicate with FLASK API. We will be using Materialize CSS for styling as it's closely matches with Google beautiful Material Design.

About the Developer

I am Vaibhav Singh pursuing Bachelor of Technology in Computer Science from BML Munjal University, Gurgaon. Currently I am in my final year and I am active contributor in open source and helped many new contributors to get in to FOSS. I am also involved in bunch of projects which uses python in FOSSASIA. I have worked in Badgeyay during codeheat as my main project. It has been a long time since I started contributing with open source and FOSSASIA, so far it has been a wonderful journey. I have learned a lot during this period which may not be possible if I have not been given an opportunity in CODEHEAT, or mentoring for Google Code-In and coaching for RGSOC and the results of these opportunities are well described below.

I am always devoted to my goals and proved them well in my previous jobs. I can prove my ability again in GSOC 2018 if this opportunity has been given to contribute to FOSSIA in this summer.

Also, I am involved in various open source programs:

1. **Codeheat 2017**

   I had participated in codeheat organized by FOSSASIA and contributed to the various projects. I also helped some students to get into FOSS via codeheat.

CodeHeath is an online coding competition that was running for 5 months, September-January. After a period of 5 months, for my work with the organization, I was declared a finalist winner.

2. **Google Code-In**

   I was mentor in Google Code-in. My task include Guiding pre-university students, code reviews, teaching software development, reviewing multiple submissions by testing them and giving timely feedback to improve them.

3. **RGSOC coach 2018**

   I am remote coach for 4 teams. My task is to help teams during the contest.

I have around 800+ contributions in FOSSASIA and also I had reviewed lots of issues and PR's.

Some Open Source Project I am associated with:

1. **Badgeyay**

   Badgeyay is a simple badge generator with a simple web UI. One can generate badges by uploading wither CSV or entering data manually. I have contributed in Badgeyay project since beginning of codeheat. I solved some critical bugs, improve code quality and improved stability.

   Issues open/closed by me:
   https://github.com/fossasia/badgeyay/issues/created_by/vaibhavsingh97

   PR Open/Closed by me: https://github.com/fossasia/badgeyay/pulls/vaibhavsingh97

2. **Query Server**

   The query server can be used to search a keyword/phrase on a search engine (Google, Yahoo, Bing, Ask, DuckDuckGo, Baidu, Exalead, Quora, Parsijoo, Dailymotion, Mojeek and Youtube) and get the results as json or xml. The tool also stores the searched query string in a MongoDB database for analytical purposes.
   Issues open/closed by me:
   https://github.com/fossasia/query-server/issues/created_by/vaibhavsingh97

PR Open/Closed by me: https://github.com/fossasia/query-server/pulls/vaibhavsingh97

3. **Open Event**

   This repository holds the JSON Schema sample implementation in the sample folder, that is used across all Open-event projects for testing.

   - Zip files that include all JSON files with binary media data
   - The uncompressed files that can act as APIs substitutes to test applications

   Issue open/closed by me:

   https://github.com/fossasia/open-event/issues/created_by/vaibhavsingh97

   PR Open/Closed by me: https://github.com/fossasia/open-event/pulls/vaibhavsingh97

4. **Loklak Python API**

   Loklak Python API is a python wrapper written in python and it's a replacement of twitter API. Now one don't need OAuth Tokens in order to display tweets from twitter.

   Issues open/closed by me:
   https://github.com/loklak/loklak_python_api/issues/created_by/vaibhavsingh97

   PR Open/Closed by me:https://github.com/loklak/loklak_python_api/pulls/vaibhavsingh97

5. **Susper Search**

   Susper is a decentralized Search Engine that uses the peer to peer system yacy and Apache Solr to crawl and index search results.
   Issues open/closed by me:
   https://github.com/fossasia/susper.com/issues/created_by/vaibhavsingh97

   PR Open/Closed by me: https://github.com/fossasia/susper.com/pulls/vaibhavsingh97

6. **Meilix**

   Beautiful Linux OS for Hotels, Libraries and Public Event Kiosks
   Issues open/closed by me:

https://github.com/fossasia/meilix/issues/created_by/vaibhavsingh97

PR Open/Closed by me: https://github.com/fossasia/meilix/pulls/vaibhavsingh97

7. **SUSI Android**

   The main feature of the app is to provide a conversational interface to provide intelligent answers using the loklak/AskSusi infrastructure. The app also offers login functionalities to connect to other services and stored personal data.
   Issues open/closed by me:
   https://github.com/fossasia/susi_android/issues/created_by/vaibhavsingh97

   PR Open/Closed by me: https://github.com/fossasia/susi_android/pulls/vaibhavsingh97

8. **Mew**

   package manager translator

   Issues open/closed by me:

   https://github.com/fossasia/mew/issues/created_by/vaibhavsingh97

   PR Open/Closed by me:
   https://github.com/fossasia/mew/issues/created_by/vaibhavsingh97

Other Contributions

1. Maintainer of two projects i.e Badgeyay, query server.
2. Actively reviewed 1000's of PR and issues.
3. Organized outreach event for FOSSASIA

Availability

Total Hours per week: 40-60 Hours per week

Outreach Events

1. Planned to organize at least 2 outreach events
2. Will share information about this project in local meetups and encourage others to contribute in this project
3. Conduct small workshops in college to help students to get started with in the FOSS through this projects.

Personal Information

I. Background

Name	Vaibhav Singh
University	BML Munjal University
Major	Computer Science
Graduation Year	2018
Skills	**Languages-** C, Java, Python, Javascript, Golang **Web Technologi**es- HTML5/CSS3, Jquery, Bootstrap, MaterializeCSS, Django, Flask, MySQL, MongoDB **Other Skills**- Git, Bash, Linux, AWS, Docker

II. Time Constraints

Time Zone	UTC + 5:30
Country	India
State	New Delhi
Working Hour	At least 40 Hours - 60 Hours per week

Basic Task per Week	Coding for project
	Daily meetup with mentor and update about the daily progress. In case of any blockage, discuss with mentor and get it resolved.
	Daily SCRUM @ 10:00 everyday about the progress
	Weekly Blog every Sunday, storing about my progress in GSOC and discussing about functionality/bug fixed in Badgeyay.

III. Contact

Email	singh.vaibhav2011@gmail.com
Gitter Nickname	vaibhavsingh97
Telephone	+91-7082077548
Portfolio Website	https://vaibhavsingh97.me/
Github	https://github.com/vaibhavsingh97
Twitter	https://twitter.com/vaibhavsingh97
Blog	https://medium.com/@vaibhavsingh97
Linkedin	https://www.linkedin.com/in/vaibhavsingh97/

References

1. Badgeyay Badge Generator:- https://badgeyay.com

2. Open Source Event Management System:- https://eventyay.com
3. My Codeheat 2017 Contributions :- https://gist.github.com/vaibhavsingh97/feecfce95859ee20747c57be4673e58b
4. CODEHEAT 2017 Results:- https://blog.fossasia.org/announcing-the-fossasia-codeheat-winners-2017-2018/
5. Basic UnitTest Example:- https://docs.python.org/2/library/unittest.html