



# Bennett Events Portal

A REPORT

First Semester Project  
ECSE103L: Computational Thinking and Programming

## Abstract

Bennett University is an activity-oriented institution. Believing firmly in the value of exposure, it strives to host a magnitude of events, ranging from guest speaker sessions to intensive hands-on workshops. With around 10 student clubs in domains from Computer Science to Finance to Photography, there isn't a single dull day in Bennett.

However, in order to communicate the event details with the students, hosts have to rely on informal means, *such as WhatsApp groups, mess hall posters, and sometimes even just word-of-mouth*. Such a system is not only extremely unreliable, but also unstructured.

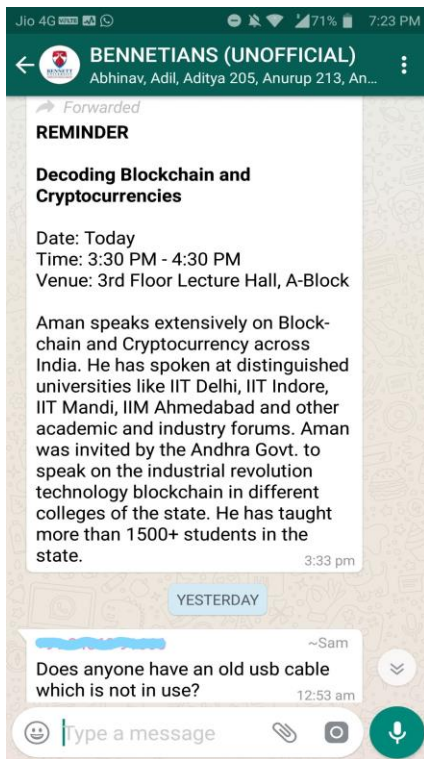


Fig 1. An example of a typical event invitation. Similar messages are shared multiple times a day on WhatsApp.

There are two main drawbacks to this system.

1. Since the groups are not meant just for invites, and *contain all other sorts of messages*, there is a high probability that users will miss the event invites amongst all the other noise, especially if they are skimming through unread messages quickly. Students also must *dig through their chat history* if they want to look at an event invite again.
2. There is *no central record* that keeps track of events being hosted in Bennett.

## Problem Statement

There is no reliable and structured way for event hosts to let students know about events and/or communicate event details with them. There is no convenient and quick way for students to check event details. There is no record of events.

## Why This Project

**Bennett Events Portal** is our solution to the above problems. It is an online notice board, accessible by users through the Internet, anywhere, anytime. It aims to make the PR part of hosting an event as painless as possible, by allowing event hosts to add an event to the Portal by just filling up a small form of around 6 fields. Once an event is added, an invitation is emailed to all students on their college accounts, immediately. The Portal serves as an online record of all events happening at Bennett. The Portal hence allows students to look up what events are scheduled for which dates, conveniently and quickly.

## Background Search/ Study of Similar Projects

An online event board is hardly a revolutionary or new idea. Similar projects are abundant on the Internet, varying in fields from design to implementation. We also went through the codebase of dozens of other web applications written in Flask, in order to be better acquainted with the nuances of the library.

The team at Real Python has a repository (<https://github.com/realpython/flask-by-example>) which contains code samples that were helpful while learning Flask.

## Literature Survey

“When in doubt, check the documentation”.

We were fortunate enough to have great documentation guiding us throughout the course of building our application.

Our sincere gratitude to the authors and editors of

- <https://docs.python.org/3/>
- <http://flask.pocoo.org/docs/1.0/>
- <http://jinja.pocoo.org/docs/2.10/>
- <https://firebase.google.com/docs/firestore/>

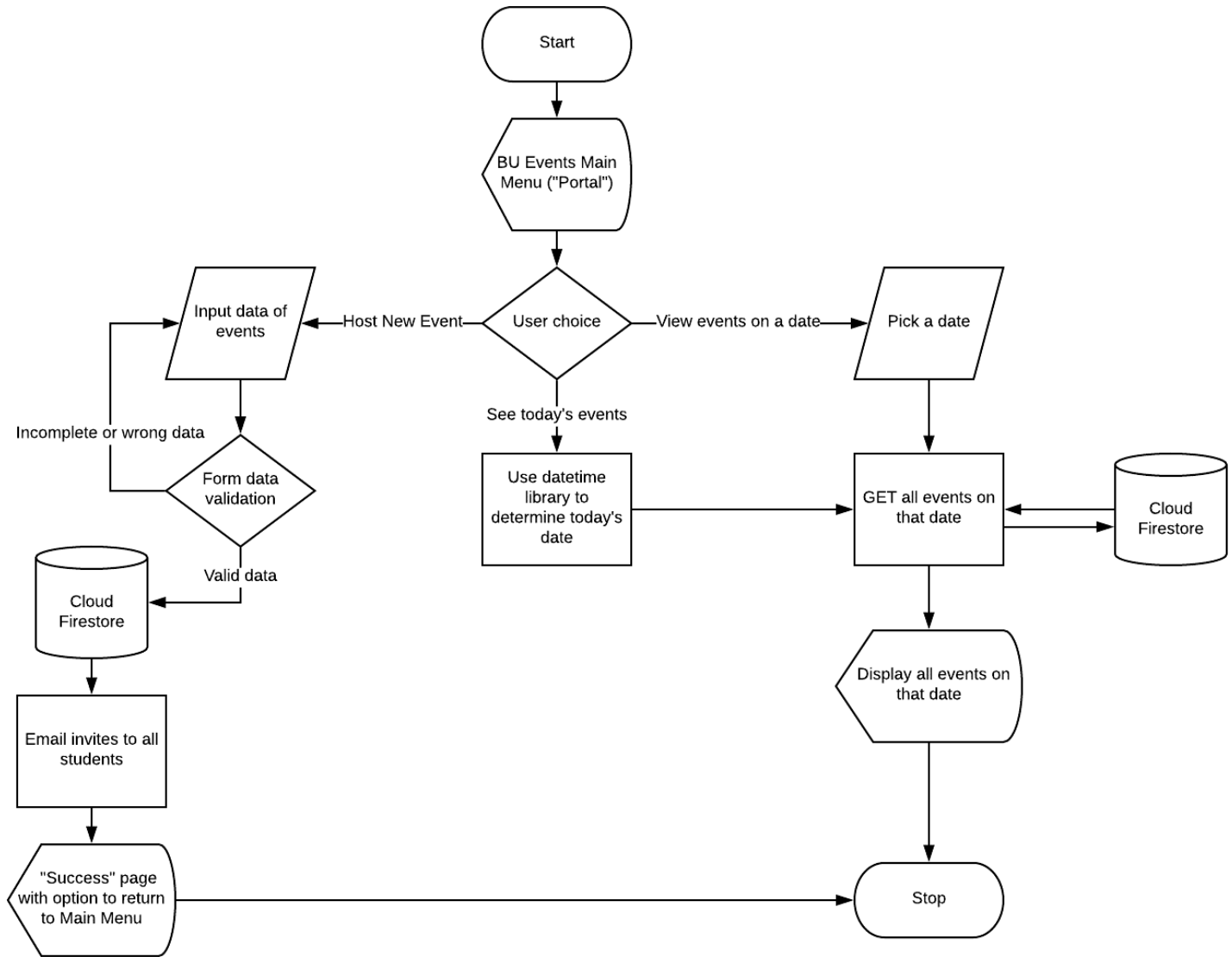
## Introduction to Web Apps

The domain area of this project is web application development. Web apps refer to applications that reside on remote servers and are accessed by the user over the Internet. Web apps are different from native apps, which are installed natively on users' devices. Therefore, web apps are easier to develop and maintain, as code written once can run platform-independently on all devices. Web apps are also easier to access, from a user's perspective.

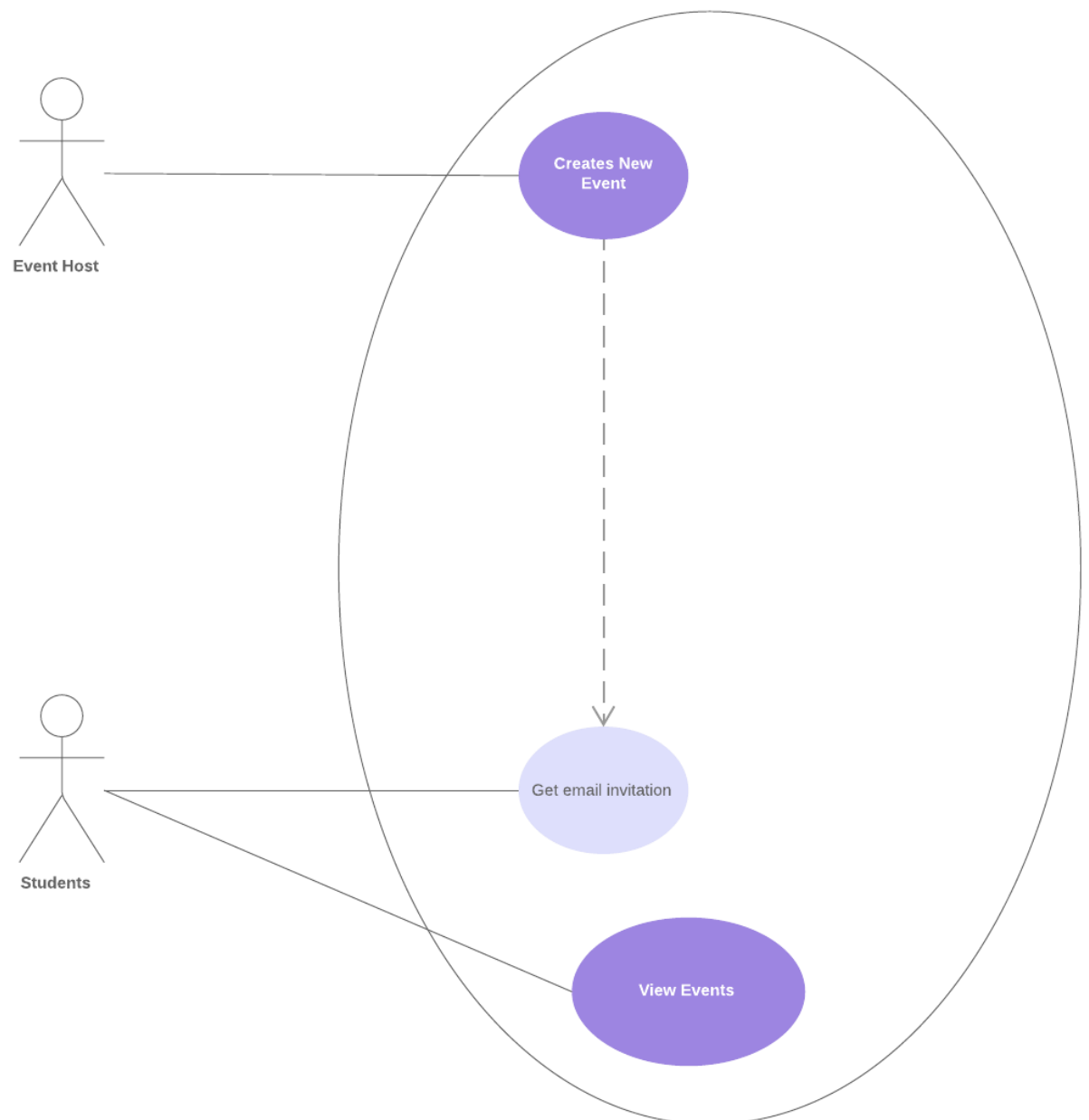
A typical full-stack web application consists of two main parts,

- Front-end: This refers to the part that is seen by the user and which the user interacts with.
- Back-end: This refers to the part that is not accessed directly by the user, but which supports the front-end by handling input/output and running processes behind the scenes.

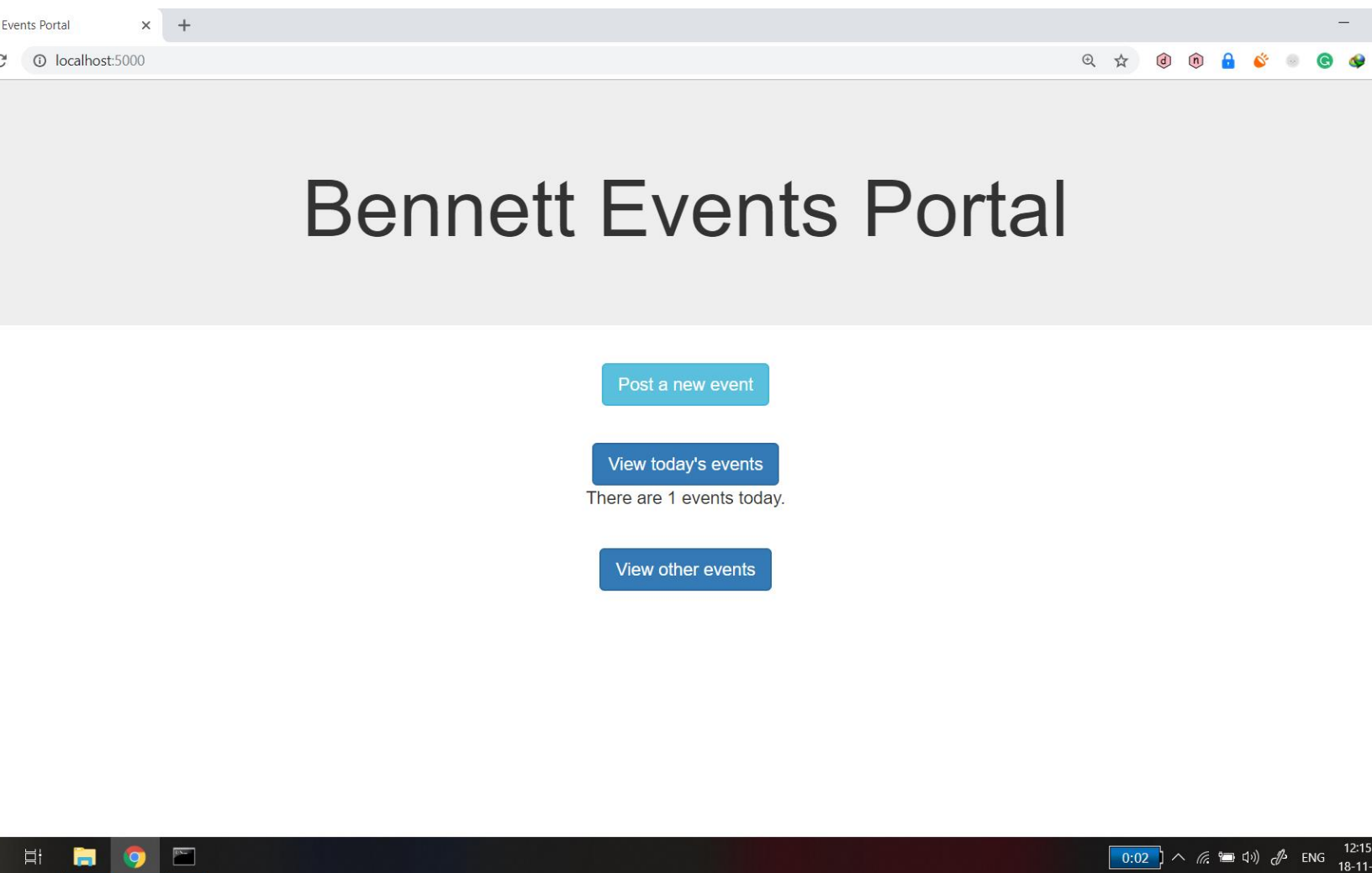
## Flow Diagram



## Use Case Diagram



## Snapshots





# Bennett Events Portal - New Event

Ahoy, host. Welcome to the invites dashboard. With the Bennett Events Portal, hosting an event in Bennett is quicker and easier than ever before. Just fill in the details, and we'll take care of the PR. So that you can focus on more important stuff. Like arranging snacks.

Club Name

Name of Event

Name of Hall

Date of Event

Time

Event Details

Submit

Club: ACM

What: CodeSenso

When: 2018-11-18, **at** 6 PM

Where: Orientation Hall

Details: Workshop

Club: ISAC

What: Mobile Photography

When: 2018-11-18, **at** 8 PM

Where: Lecture Hall

Details: Guest Session

[Try another date](#)

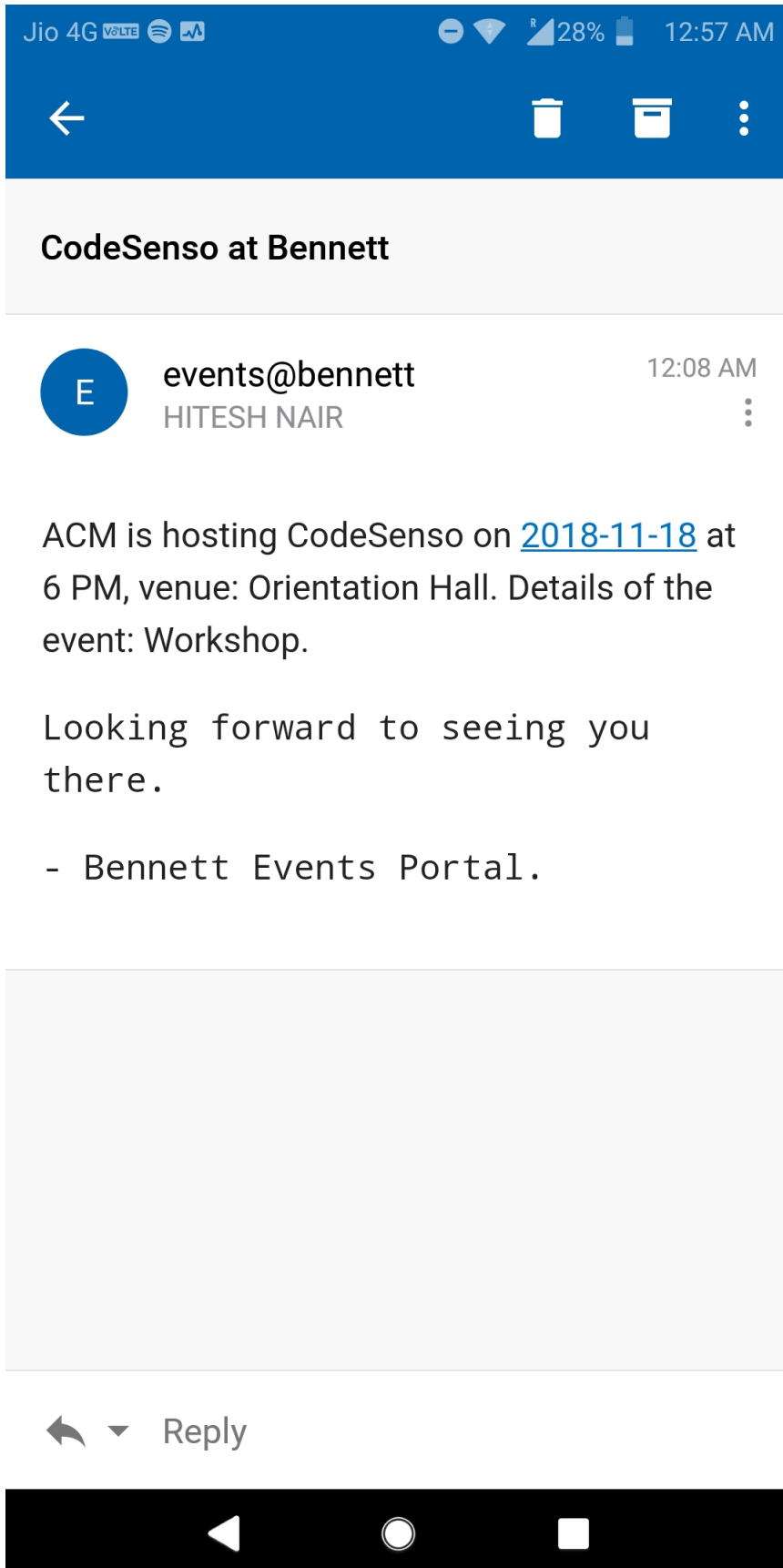
[Main Menu](#)

# Bennett Event Board - See events on a specific date

Choose date

Submit





## Implementation Details

- Written in Python 3.x
- Libraries:
  - Flask – *Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions.*
  - WTForms – *WTForms is a flexible forms validation and rendering library for Python web development.*
  - Flask\_WTF – *Integration with WTForms.*
- Cloud Firestore:

*Cloud Firestore is a NoSQL document database for mobile, web, and server development from Firebase and Google Cloud Platform, that lets you easily store, sync, and query data for your apps - at global scale.*
- SendGrid:

*SendGrid is a leader in trusted email delivery. Over 78,000 paying customers like Spotify, Uber and Airbnb trust SendGrid to send over 45 billion emails every month.*

## Implementation Overview

In this project, we use Flask, a Python library used for web app development, as well as Jinja, a template engine, to render our front-end to the user. We're using Python code in the backend to extract user input and store it in our NoSQL database, Cloud Firestore. We're then using SendGrid's API to send e-mails to students.

## Limitations & Future Enhancements

- The invitation emails sent sometimes land in the spam/junk folder of the students. This can be fixed by improving our sender reputation but that is beyond the scope of this project.
- This is just a working model/prototype. It hasn't been deployed online yet.
- The UI is very basic and needs to be made more appealing.

## Learnings from this project

- Rendering web pages with Flask and Jinja
- Using Python in web backends
- Using APIs with Python

## Team Members



**Hitesh Nair (E18CSEo64)**

hello@hiteshnair.com



**Isha Puthige (E18CSEo66)**

ishaputhige@gmail.com



**Chahat Bindra (E18CSEo37)**

chahat.bindracb@gmail.com



**Harsh (E18CSEo61)**

geemax.harsh@gmail.com



**Gautam Jain (E18CSEo56)**

mvgt.gautam@gmail.com