

Web Technology 2 Project

Emblaze

A simple to use web app to create or beautify existing resumes.

Submitted By:

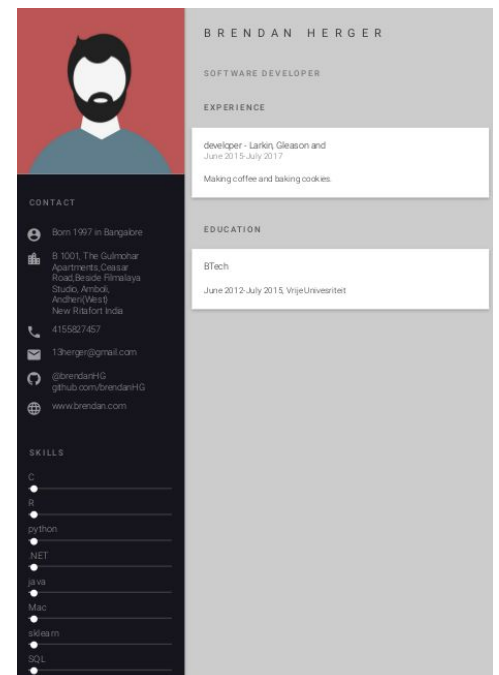
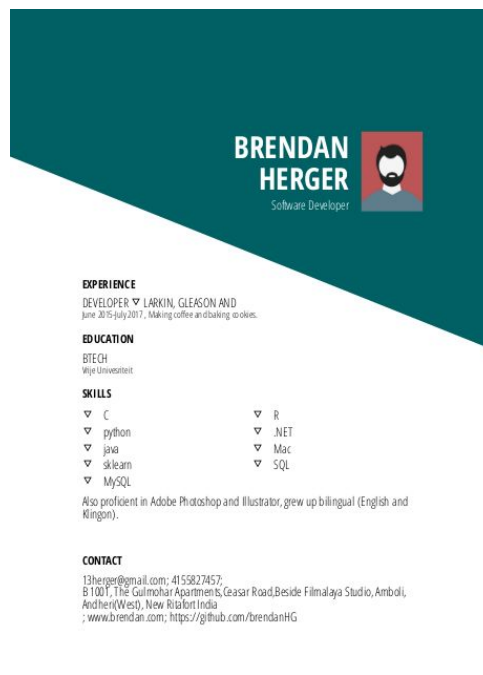
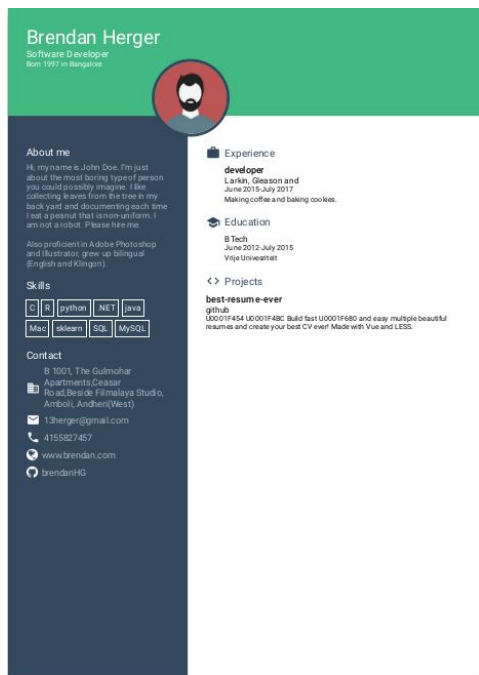
Rahul R Bharadwaj	01FB15ECS366
Chirag Ramesh	01FB15ECS080
Gurunandan N	01FB15ECS112

Synopsis

Emblaze allows you to create beautiful and breathtaking resumes. You can either:

- Create a resume from scratch.
- Create a resume by uploading a PDF and letting our system to intelligently guess the details by parsing your resume with industry standard NLP package: spacy.

Here's a simple demo of what our resumes look like:



Technologies

The following technologies were used. The reason for using the same are mentioned. Internal modules if used are specified with their reasons.

1. Flask:

- a. Reason: Used as server number 1. Had following functionalities:

- i. Getting PDF uploaded and passing it onto resume-parser module.
- ii. Placing the results in appropriate boxes in the common input form.
- iii. Handling the form input and passing it onto Node server for placing the input into the resume templates (via VueJS).

b. Module 1: **Jinja2**, Reason:

- i. Flask uses Jinja2 as its templating engine.
- ii. Jinja2 replaces all “mustaches” appearances in a document with data/variables mentioned within them.
Eg: `{{ container.value }}`

c. Module 2: **MySQL connector**, Reason:

- i. Used to connect to MySQL server.
- ii. Server stored city, state and company names.

2. Spacy:

a. Reason:

- i. Used for NLP. Spacy is a Industrial-strength Natural Language Processing module written in Python.
- ii. We used it to extract details from text version of resume PDF.
- iii. The details are obtained by running the module over with certain rules. Eg:
Rule for getting name of candidate:

```
def candidate_name_extractor(input_string, nlp):
    input_string = str(input_string)

    doc = nlp(input_string)

    # Extract entities
    doc_entities = doc.ents

    # Subset to person type entities
    doc_persons = [x for x in doc_entities if x.label_ == 'PERSON']
    doc_persons = [x for x in doc_persons if len(x.text.strip().split()) >= 2]
    doc_persons = [x.text.strip() for x in doc_persons]

    # Assuming that the first Person entity with more than two tokens is the candidate's name
    if doc_persons:
        return doc_persons[0]
    return "NOT FOUND"
```

While extracting simpler details like email and phone are done via regex:

- iv. The details are given by an extractor, which in turn reads a

```
EMAIL_REGEX = r"[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}"
PHONE_REGEX = r"(\d{3})?\d{3}(\d{3})?(\d{3})?(\d{4})"
```

```
extractors:
  experience:
    - [Teacher, teaching, tutor]
    - [developer, software developer, software engineer, dev]
    - trader

  platforms:
    - Linux
    - Windows
    - [Mac, MacOS]
    - Android
    - iOS

  database:
    - SQL
    - MySQL
    - [Postgress, Postgresql]
    - Oracle
```

yaml

template file. This file contains some data that helps the extractor to get data:

3. VueJS:

- a. Reason:

- i. It's an open-source JavaScript framework for building user interfaces and single-page applications.

- ii. We used this to fill template resumes with data entered by user.
- iii. VueJS recommends NodeJS to be the backend. This is beneficial **as both VueJS and Jinja2 use double mustaches** as their template matching syntax.
- iv. Thus we used VueJS with NodeJS

4. NodeJS:

- a. Reason: Server number 2
 - i. Has only one job: Run VueJS on the data input.
 - ii. Then render the new resumes onto users screen.

5. Bootstrap 4.0, jQuery:

- a. Reason:
 - i. Bootstrap used for easy to use CSS and javascript extensions.
 - ii. jQuery for cross browser compatible javascript.

6. Pandas, Textract, PyYAML:

- a. Reason:
 - i. Pandas used for reading and writing into CSVs.
 - ii. Textract used to convert PDF to text.
 - iii. PyYAML to convert dictionaries into YAML files.

Reasons for using above Technologies

1. Flask:

- a. Competitors: Django in python and other servers.
- b. Used mainly because NLP packages are mostly available in python.

- c. Also tools for reading from CSVs, writing YAML files, reading from PDF, are readily available in python.
- d. Thus we needed a python based server. Since our needs were small we didn't go with Django and went with Flask.

2. Spacy:

- a. Competitors: NLTK, which is another NLP toolkit in python.
- b. We chose Spacy as it's meant for industry grade NLP and many industries use it.
- c. Spacy's interfaces are also well defined and easy to use.

3. VueJS:

- a. Competitors: AngularJS or ReactJS.
- b. The Virtual DOM model is very helpful in terms of performance. Both React and Vue have a Virtual DOM. Due to a well-built structure, Vue delivers great performance and memory allocation.
- c. Vue is the cleanest in comparison to these three frameworks.
- d. We desired separation of concerns and thus settled on VueJS.

4. NodeJS:

- a. Reasons are as mentioned in previous section.

AJAX Techniques used:

1. Predictive Fetch:

- a. Used in homepage to fetch items only if user scrolls.
- b. This is important as homepage has a lot of images and javascript which is unnecessary to be loaded if user doesn't scroll.
- c. Thus Predictive fetch allows us to save bandwidth by transmitting only when needed.

2. Submission Throttling:

- a. Used to fetch commonly entered data as user enters data into a location.
- b. Eg: Shows suggestions of states, cities and companies as user enters their names.
- c. Reduces user input time and gives a better overall experience.

Other web technology considerations:

- Considering performance as a metric, we've used Content Delivery Networks to fetch some of our libraries.
- Also, we've used the minified versions to reduce bandwidth consumption.

Future Scope:

- Better conversion of PDF to text. The current textract module doesn't do a good job when PDF structure is complex.
- Using a NLP toolkit which can detect Indian names. Spacy being a US based product does not detect Indian names. This can be updated by contributing to their package or creating one for Indian purpose.

Acknowledgment

This project was part of Web Technology 2 course. We are thankful to our professor Mr. Sivaraman E and Srikanth R for their helpful guidance.