

**Project Report**

# **Design Thinking and Innovation**

**(ECSE235P)**

**Savo**



**Bennett University**

**School of Engineering & Applied Sciences**

**Submitted by:**


**Savo (EB05)**

PARITOSH TRIPATHI (E20CSE067)

ALTAMASH ALAM (E20CSE055)

DRISHTI MAKHIJANI (E20CSE070)

ANANT GOVIL (E20CSE007)



November 2021

## Declaration

This report has been prepared based on our work. Where other published and unpublished source materials have been used, these have been acknowledged.

## Table of Contents

[Chapter 1: Introduction](#)

[1.1 A Point worth mentioning](#)

[Chapter 2: Data Collection](#)

[Chapter 3: Data cleaning](#)

[Chapter 4: Recommendation system](#)

[4.1 Technologies](#)

[4.2 Algorithms](#)

[Chapter 5: Personalization for Every User](#)

[5.1 Each User, Unique Recommendation](#)

[5.2 Source Code](#)

[Chapter 6: Table of Contents and References](#)

[Chapter 7: Project Information and Rules](#)



## Abstract

The arduous work of students and the time they give away while searching for the right course take up most of their time. In return, they are left with a lot less time to study the course in the most efficient manner. Savo is the one-stop solution for that. The task of browsing through tens of websites for the courses about the required topic is on Savo. It helps you to filter your search for courses across different websites to be on one spot and provides the students with the necessary details along with the search. All of this makes Savo the right choice



## Project Specification

### Requirements –


1. The most basic resource that the project requires is the data of courses that are being hosted across different websites. The data should be well maintained in a proper manageable way. The best bet on the kind of data would include categories like the instructor, the institution providing the course, the price, etc.
2. Another requirement would be to provide a better GUI for the keen learners. For this, we use *Django* to define the frontend, giving users a seamless experience when they are browsing through the courses.
3. Finally, our project would require investors to initiate or kickstart the project. The funds further will be used in the marketing and advertising of our platform.

### So, the project can be divided into 4 parts –

4. Web platform - (react for frontend and Django for backend) these two technologies are new to us and will take considerable time to learn and implement.
5. Data collection - We will be needed to collect data from various websites like reviews of courses, length of courses, etc, and since only 1 member knows about web scraping.
6. Data cleaning - so the data collected will be cleaned before putting it into the recommendation system or the rating algorithm.
7. Recommendation system and Rating algorithm - these are two separate skills that will be needed to be worked upon.

### Workflow –

- Month 1 -
  1. We will be researching the most optimized way to implement for backend and frontend using API or some other methods.

- 
- 2. Work on the algorithm to rate the courses as fast as we can since a delay in showing results might lead to a loss in the user base.
    - 3. Will start collecting data from various sources.
  - Month 2 –
    - 1. After the research is done, we will be starting to set up the database.
    - 2. Start working on cleaning the data.
    - 3. Start working on setting up the pipeline for model building.
  - Month 3 –
    - 1. Development of the website starts and the building of UI/UX starts.
    - 2. Backend development starts.
    - 3. Data cleansing and Training of the model.
  - Month 4 –
    - 1. Implementation and integration of all the elements start.



# Chapter 1: Introduction

Savo aims to provide a one-stop solution to one of the most prominent problems among college/school students, i.e. Which online certification course would be best suited for them? Often it is confusing for students to figure out which course to take up when there are about tens of them teaching the same topic. Savo would provide them with a single platform that helps them to compare different courses on the same topic based on specific filters. No more would it be a hassle for students to get the right kind of guidance in choosing a course once they get their hands on Savo.

Simply put, students would find themselves at ease once they get their hands on Savo. The courses, the cost of it, the ratings. They all would be at one spot itself hence they know that this is the right course for them.

## 1.1 A Point worth mentioning

Many of you may be able to relate this project with 'Trivago', a similar kind of website which is used to compare hotels. The key difference here is that 'Trivago' is meant for finding the cheapest price for the same kind of hotel. On the other hand, Savo is meant for giving you a list of courses across different websites and on top of that also gives you a better recommendation rating which is personalized from user to user.



## Chapter 2: Data Collection

For this Data collection is the most important part of the project. While none of the course platforms provides API for data collection, we need to do data collection using web scraping.

The tools and tech we are going to use for data collection –

1. Python
2. Selenium
3. BeautifulSoup
4. Pandas
5. NumPy

Websites that we need to get data from-

- YouTube
- Udemy
- Coursera
- Edx
- Harvard and Stanford
- IBM
- Google
- Microsoft
- Freecodecamp

Here is one of the following python scripts used for scraping Harvard –

```
def Data():
    for i in range(0, page):

        # this is the main link dont touch this

        print("Loading Page number ", i)

        driver.get(
            'https://online-learning.harvard.edu/catalog?keywords=&start_date_range%5Bmin%5D%5Bdate%5D=&start_date_range%5Bmax%5D%5Bdate%5D=&page={}'.format(
                i))

        content = driver.page_source.encode('utf-8').strip() # this get content of a page in normal way

        soup = BeautifulSoup(content, 'lxml') # this is used to find the things we need in the source code

        link_class = soup.findAll("div", class_="field field-name-title-qs") # titles of youtube videos is stored here

        for j in link_class[0:]:
            link_tag = j.find("a")
            base = 'https://online-learning.harvard.edu'
            try:
                if 'href' in link_tag.attrs:
                    link = link_tag.get('href')
            except:
                pass

            url = urljoin(base, link)
            urls.append(url)

Data()

for links in range(0, len(urls)):
    driver.get(urls[links])

    new_soup = BeautifulSoup(driver.page_source, "lxml")

    title = new_soup.find("div", class_="field field-name-title")

    subject = new_soup.find("div", class_="field field-name-subject-area field-type-ds field-label-inline clearfix")
    subject_tag = subject.find("a")

    length = new_soup.find("div", class_="field field-name-field-duration")

    difficulty = new_soup.find("div",
                               class_="field field-name-field-difficulty field-type-list-text field-label-inline clearfix")
    difficulty_tag = difficulty.find("div", class_="field")

    description = new_soup.find("div", class_="field field-name-body field-type-text-with-summary field-label-above")
    description_tag = description.find("p")
```



## Chapter 3: Data cleaning

Data processing occurs when data is collected and translated into usable information. Usually performed by a data scientist or team of data scientists, it is important for data processing to be done correctly so as not to negatively affect the product or data output.

Data processing starts with data in its raw form and converts it into a more readable format (graphs, documents, etc.), giving it the form and context necessary to be interpreted by computers and utilized by employees throughout an organization.

Tools and technologies used for data processing –

1. Pandas
2. NumPy
3. Seaborn
4. Jupyter notebook

Following is a script used for data cleaning –

```
In [1]: import pandas as pd

In [2]: udey = pd.read_csv("Udey_Programming_courses.csv")

In [3]: youtube = pd.read_csv("youtube_csv_.csv")

In [4]: harvard = pd.read_csv("Harvard_courses_csv.csv")

In [5]: udey.columns
Out[5]: Index(['url', 'Title', 'Course Headline', 'Instructor', 'Rating', 'Price', 'length', 'number_of_lectures', 'difficulty'], dtype=object)

In [6]: youtube.columns
Out[6]: Index(['NULL', 'links', 'names', 'Views', 'likes'], dtype=object)

In [7]: harvard.columns
Out[7]: Index(['NULL', 'links', 'names', 'Type', 'Duration', 'Price', 'difficulty', 'Description', 'platform'], dtype=object)

In [8]: youtube.drop(['NULL', 'Views', 'likes'], axis = 1, inplace=True)

In [9]: harvard.drop(['NULL', 'Type', 'Duration', 'difficulty', 'Description', 'platform'], axis = 1, inplace=True)

In [10]: udey.drop(['Course Headline', 'Instructor', 'Rating', 'length', 'number_of_lectures', 'difficulty'], axis = 1, inplace=True)

In [11]: udey = udey.dropna()

In [12]: youtube = youtube.dropna()

In [13]: youtube['names'] = youtube['names'].str.replace(r'\n', '')
FutureWarning: The default value of regex will change from True to False in a future version.
youtube['names'] = youtube['names'].str.replace(r'\n', '')

In [14]: udey
Out[14]:
```

	url	Title	Price
0	https://www.udemy.com/course/python-for-finance...	Python for Finance and Algorithmic Trading with...	Current price\$455
1	https://www.udemy.com/course/deeplearning_ai/	A deep understanding of deep learning (with Py...	Current price\$455
2	https://www.udemy.com/course/the-python-mega-c...	The Python Mega Course: Build 10 Real World Ap...	Current price\$455
3	https://www.udemy.com/course/the-ultimate-mysql...	The Ultimate MySQL Bootcamp: Go from SQL Begin...	Current price\$455
4	https://www.udemy.com/course/the-advanced-web-...	The Advanced Web Developer Bootcamp	Current price\$455
...	...	...	...
3966	https://www.udemy.com/course/sql-bootcamp-hand...	SQL Bootcamp - Hands-On Exercises - SQLite - P...	Current price\$455
3967	https://www.udemy.com/course/ssas-sql-server-a...	SQL Server Analysis Services - SSAS, Data Mini...	Current price\$455



## Chapter 4: Recommendation system

### 4.1 Technologies

- Neat text – for cleaning of descriptions and titles
- Pandas – for data frame handling
- Sklearn CountVectorizer, TfidfVectorizer
- Sklearn cosine\_similarity, linear\_kernel

### 4.2 Algorithms

- Approach
1. Task 1 - First we will get all the data and use description and title for getting keywords. We will use neat text for getting rid of all prepositions and other characters from the description.
  2. Task 2 - Building a new column of clean titles and descriptions from original columns to help convert them.
  3. Task 3 - now we will convert the cleaned titles into Vector using sklearn CountVectorizer, TfidfVectorizer.
  4. Task 4 - Now we will use cosine similarity to generate a similarity score between courses and other parameters
  5. Task 5 - Now we will sort the scores to get the top 10 scores when a keyword is entered.

```

1: # Cosine Similarity Matrix
cosine_sim_mat = cosine_similarity(cv_mat)

2: cosine_sim_mat

3: array([[1.         , 0.1490712 , 0.15811388, ..., 0.         , 0.         ,
0.         ,
0.1490712 , 1.         , 0.11785113, ..., 0.         , 0.         ,
0.         ,
0.15811388, 0.11785113, 1.         , ..., 0.         , 0.08574929,
0.         ,
...,
0.         , 0.         , ..., 1.         , 0.12126781,
0.         ,
0.         , 0.         , 0.08574929, ..., 0.12126781, 1.         ,
0.32539569],
0.         , 0.         , ..., 0.         , 0.32539569,
1.         ]])

4: df.head()

5:
   Unnamed: 0      Title      clean_course_title
0      0  Python for Finance and Algorithmic Trading wit...  Python Finance Algorithmic Trading QuantConnect
1      1  A deep understanding of deep learning (with Py...  deep understanding deep learning with Python L...
2      2  The Python Mega Course: Build 10 Real World Ap...  Python Mega Course Build 10 Real World Applica...
3      3  The Ultimate MySQL Bootcamp: Go from SQL Begin...  Ultimate MySQL Bootcamp SQL Beginner Expert
4      4  The Ultimate MySQL Bootcamp: Go from SQL Begin...  Ultimate MySQL Bootcamp SQL Beginner Expert

6: # Get Course ID/Index
course_indices = pd.Series(df.index, index=df['Title']).drop_duplicates()

7: df.Title[1]

8: 'A deep understanding of deep learning (with Python intro)'

9: idx = course_indices['A deep understanding of deep learning (with Python intro)']

10: scores = list(enumerate(cosine_sim_mat[idx]))

11: # Sort our scores per cosine score
sorted_scores = sorted(scores, key=lambda x: x[1], reverse=True)

12: # Selected Courses Indices
selected_course_indices = [i[0] for i in sorted_scores[1:]]

13: # Selected Courses Scores
selected_course_scores = [i[1] for i in sorted_scores[1:]]

14: recommended_result = df['Title'].iloc[selected_course_indices]

15: rec_df = pd.DataFrame(recommended_result)

16: rec_df.head()

17:

```

```

def recommend_course(title, num_of_rec=10):
    # ID for Title
    idx = course_indices[title]
    # Course Index
    # Search inside cosine sim mat
    scores = list(enumerate(cosine_sim_mat[idx]))
    # Scores
    # Sort Scores
    sorted_scores = sorted(scores, key=lambda x: x[1], reverse=True)
    # Recomm
    selected_course_indices = [i[0] for i in sorted_scores[1:]]
    selected_course_scores = [i[1] for i in sorted_scores[1:]]
    result = df['Title'].iloc[selected_course_indices]
    rec_df = pd.DataFrame(result)
    rec_df['similarity scores'] = selected_course_scores
    return rec_df.head(num_of_rec)

recommend_course("Nonprofit Financial Stewardship Webinar: Introduction to Accounting and Financial Statements", 20)

```

	Title	similarity scores
4095	Financial Accounting	0.670820
4096	Financial Accounting	0.670820
2750	SAP S4/HANA -Financial Accounting C_TS4FI_1909...	0.387298
2672	Introduction to R	0.316228
2673	Introduction to R	0.316228
2674	Introduction to R	0.316228
1024	Python for Financial Analysis and Algorithmic ...	0.282843
1222	Financial Engineering and Artificial Intellige...	0.282843
1540	Introduction to R Programming - Must See Intro...	0.282843
1581	Python & Machine Learning for Financial Analysis	0.282843
3296	Python and Machine Learning in Financial Analysis	0.282843
3925	New Ideas for Nonprofit Leaders Webinar	0.282843
1913	Python for Finance 2021: Financial Analysis fo...	0.258199
4121	Health Care Financial Management for Physician...	0.258199
1700	Introduction to R Programming	0.223607
1884	Introduction to Statistics with R	0.223607
2511	Introduction to Clustering using R	0.223607
3475	Introduction To The C Language	0.223607
3476	Introduction To The C Language	0.223607
3713	Introduction to SQL	0.223607



## Chapter 5: Personalization for Every User

### 5.1 Each User, Unique Recommendation

Every user is given personal care as they are recommended with the courses which are based on their past search history. Suppose a person searches for ML courses. In such a case the user would be given more recommendations on ML courses themselves. And that too

.

### 5.2 Source Code

A *Code* style has been prepared for formatting short excerpts of source code. It is a simple indented, single-spaced style using a fixed font (Courier New) to produce code that appears like the following:

Find the code for all scripts at - <https://github.com/paritoshtripathi935/Savo>

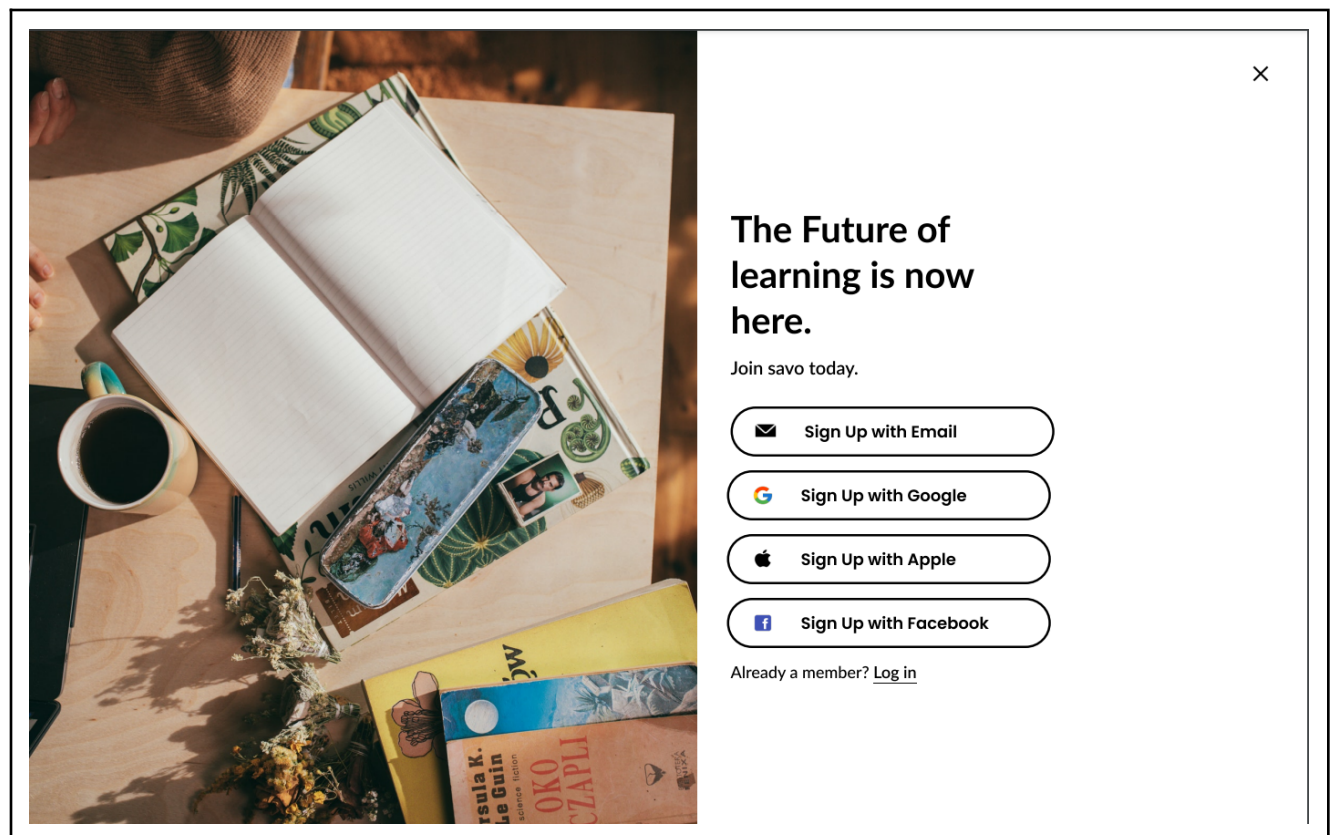
## Chapter 6: User Experience and User Design

A table of contents (TOC) page has also been included in this report template. Before delivering your report, remember to update your existing table of contents by right-clicking it and selecting the *Update field* option. If there are new sections since the last update, you should select the *Update entire table* option.

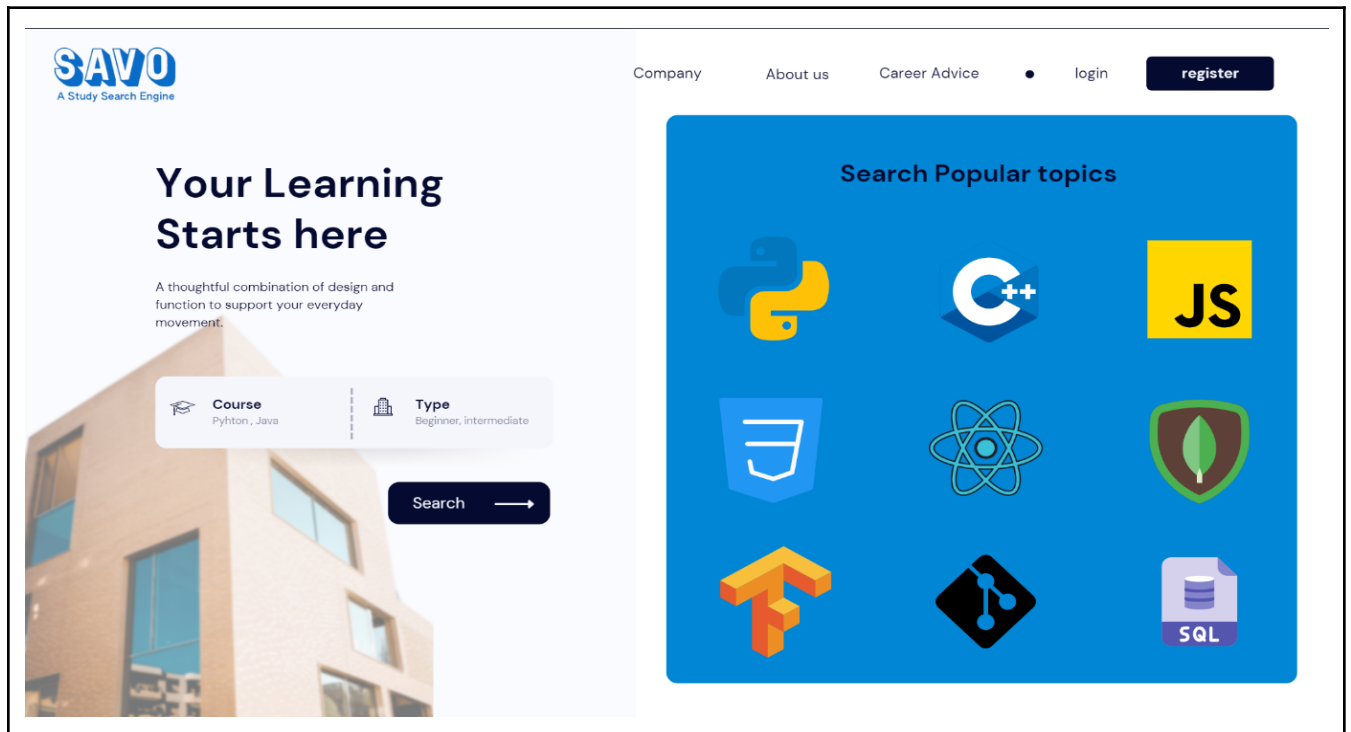
A word of warning on this feature – the table of contents is automatically generated by compiling a table of all the level 1 and 2 headings in your document. This means that every line with one of these styles will appear in the table. If you use these styles for non-headings (of course you should not do this) then these non-headings will also appear in the table.

Use one consistent system for citing works in the body of your report. Several such systems are in common use in textbooks and conference and journal papers. Ensure that any works you cite are listed in the references section, and vice versa.

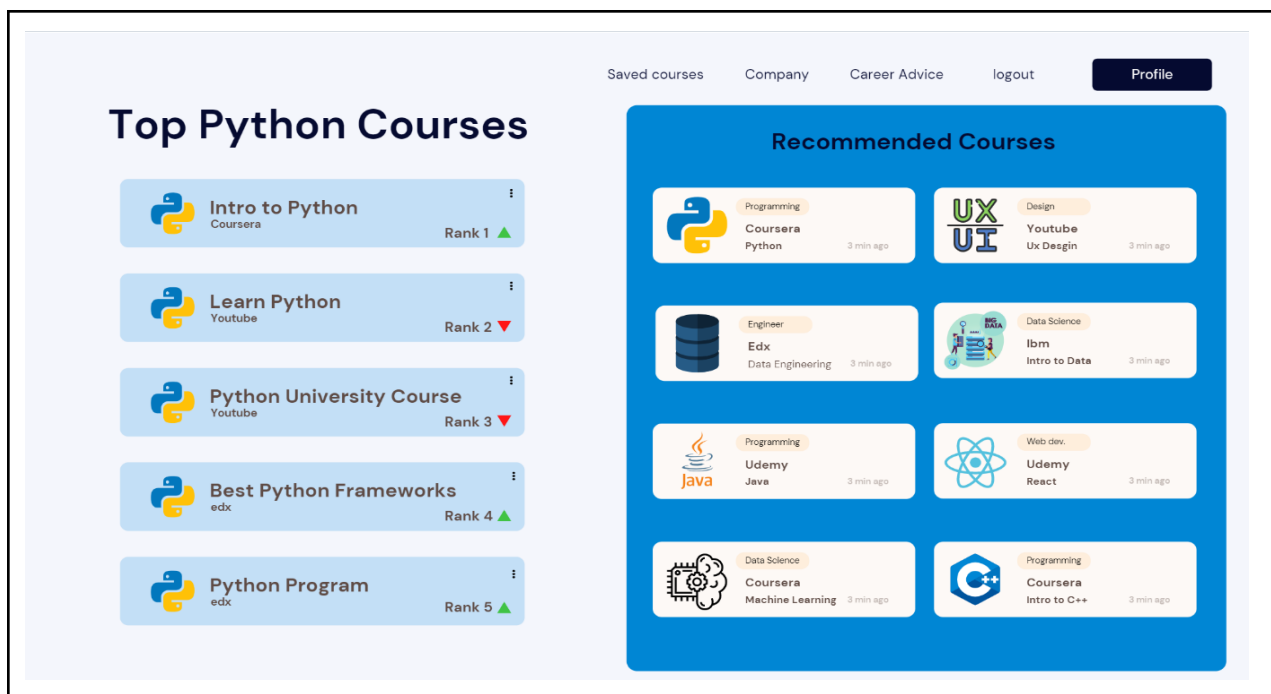
### Login Page for Desktop Website



## Homescreen Design



## Search Results Screen





# Chapter 7: Project Closure

## 7.1 Vision

Our vision for savo is to completely develop savo and show the world its full potential, it is a complex project and might need some more time to get it fully developed.

We might need to expand the team to our technical team to manage the massive amount of work and might also need to manage the team/marketing team.

## 7.2 Remaining work

The transformation of UI/UX designs into a frontend was a tedious task and might need a serious grind to complete basic files and designs apart from that we might need to automate the data collections and data storage.

Hosting the website is another task we need to set up the database and should keep security measures in mind.

## 7.3 Future Prospects

Like any desktop web application or a mobile application we need to regularly update the UI so that the user engagement is maximum. Hence, the UI of SAVO should regularly be updated. Further down, we would improve the recommendation system so that users would be more satisfied.