Tribhuvan University

Faculty of Humanities and Social Sciences

SPOTLIGHT: A VIRTUAL CROWDFUNDING PLATFORM

A Project Report

Submitted to

Department of Computer Application

Ojashwi College

In partial fulfillment of the requirements of Bachelors in

Computer Application

Submitted by

Kshitiz Wagle (6-2-346-13-2019)

October 2023

Under the supervision of

Ganesh Prasad Bhatta

Tribhuvan University

Faculty of Humanities and Social Sciences

Ojashwi College

Supervisor's Recommendation

I hereby recommend that this project be prepared under my supervision by Kshitiz Wagle entitled "**Spotlight**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

……………………..

SIGNATURE

Ganesh Prasad Bhatta

SUPERVISOR

Department of Computer Application

Ojashwi College

Gwarko, Lalitpur

Tribhuvan University

Faculty of Humanities and Social Sciences

Ojashwi College

LETTER OF APPROVAL

This is to certify that this project prepared by Kshitiz Wagle entitled "**Spotlight**" in partial fulfillment of the requirements for the Bachelors of Arts in Computer Application has been evaluated. In our opinion it is satisfactory in scope and quality as a project for the required degree.

SIGNATURE of Supervisor …………………………………………... Mr. Ganesh Prasad Bhatta, Project Coordinator Department of Computer Application Ojashwi College Gwarko, Lalitpur	SIGNATURE of HOD/Coordinator ………………………………………….. Mr. Santosh Rijal Department of Computer Application Ojashwi College Gwarko, Lalitpur
SINGATURE of Internal Examiner	SIGNATURE of External Examiner

Abstract

Spotlight has been set to target the transformation of Nepal's entrepreneurial landscape. In a world where innovation is crucially driven by crowdfunding, Nepal's emerging entrepreneurs find themselves facing unique challenges. Existing global platforms often fail to cater to their specific needs. This project is in the process of developing a crowdfunding platform called Spotlight, which is tailored to the Nepali market and prioritizes cost-effectiveness and user-friendliness. The objective is to facilitate swift and efficient fundraising, offer expert guidance, and provide incentives for startup growth, all aimed at bridging the gap between entrepreneurs and investors. This, in turn, enables Nepali innovators to be empowered on a global scale. Through Spotlight, entrepreneurs gain access to the support and resources needed to bring their visionary ideas to fruition, ultimately contributing to the growth and prosperity of Nepal's entrepreneurial ecosystem.

Keywords: crowdfunding, online funding, Spotlight, software development, virtual crowdfunding

Acknowledgement

This project report is prepared in the partial fulfillment of the requirements for the Bachelor of Arts in Computer Application. I want to thank everyone who played a role in making it possible.

I'd like to express my gratitude to Mr. Ganesh Parsad Bhatta, my supervisor at Ojashwi College, for guiding and supporting me throughout. I also appreciate Mr. Santosh Rijal, our coordinator, for his ongoing help.

This project wouldn't exist without the kindness of people who are always ready to help others and support good ideas. We owe them a big thank you for giving this project its purpose.

Table of Contents

List of Abbreviations

API	Application Programming Interface
DB	Database
DFD	Data Flow Diagram
ER	Entity Relationship
JS	JavaScript
MVT	Model View Template
SQL	Structured Query Language
UML	Unified Modeling Language
DF	Dataframe
NLTK	Natural Language Toolkit

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

Spotlight is a web-based crowdfunding platform designed for entrepreneurs. It gives them one place to create, display, and share their project ideas. Spotlight connects entrepreneurs with potential investors, offers flexible fundraising options, and helps validate their concepts. Plus, it's a platform to promote their projects.

Our application is designed to minimize errors when entering data and provides helpful error messages for invalid inputs. You don't need to be a tech expert to use it, making it user-friendly. This system aims to be error-free, secure, reliable, and efficient, allowing users to focus on their main tasks instead of record-keeping.

Traditionally, entrepreneurs would spend a lot of time and effort searching for investors. Crowdfunding platforms like IndieGoGo, Crowdcube, and Seedrs have simplified this process, helping many beginners and fundraisers succeed. Other well-known platforms include GoFundMe and Boost Turku.

1.2 Problem Statement

In Nepal, as the entrepreneurial and startup scene continues to grow, raising funds has become a significant challenge. Entrepreneurs often must meet with investors and venture capitalists in person, which requires setting up appointments and navigating through complex and uncertain bank loan procedures. While platforms like Kickstarter and Indiegogo exist for crowdfunding, they are primarily aimed at the global market and may not be the perfect fit for a developing country like Nepal.

This gap in the market means that there are limited modern platforms in Nepal for showcasing startup projects and securing funding. Currently, crowdfunding platforms in Nepal are predominantly used for collecting funds for social causes and company initiatives. In today's technology-driven era, networking and information flow are crucial, and the absence of such platforms only widens the communication gap between startup founders and potential investors or audiences.

It's been observed that the dissemination of updates and information about services and events by various profit and non-profit organizations has not been very effective. This lack of effective communication means that startups struggle to reach their projects and ideas to their potential audience, which limits their overall impact.

1.3 Objectives

Crowdfunding serves to secure funds for artistic, creative projects, and startups. The objectives of this project can be summarized as follows:

- To offer a quick and cost-effective method of raising funds without any initial charges, while also providing a platform to promote projects for marketing and media exposure.
- To serve as a platform where individuals can receive expert guidance and constructive feedback on their concepts, in addition to obtaining funding for a wide range of innovative ideas.
- To encourage and incentivize the emergence of numerous new startups within the country by making funding and resources more accessible to aspiring entrepreneurs.

1.4 Scope and Limitations

1.4.1 Scopes

Successfully crowdfunded projects can get huge amounts of attention, on social media and elsewhere, which can help them grow beyond what the money raised alone could have done. As part of the crowdfunding process, the business can get feedback about their idea and how to improve it. One of the best things about online crowdfunding is its ability to centralize and streamline your fundraising efforts.

The major scope of the crowdfunding project can be listed as follows:

- It can be a fast way to raise finance with no upfront fees.
- Pitching a project or business through the online platform can be a valuable form of marketing and result in media attention.
- It is a good way to test the public's reaction to your product/idea - if people are keen to invest it is a good sign that your idea could work well in the market.
- Ideas that may not appeal to conventional investors can often get financed more easily.
- It's an alternative finance option if you have struggled to get bank loans or traditional funding

1.4.2 Limitations

Crowdfunded projects are visible, finite, and understandable. If your project isn't all three, it's unlikely to succeed. If the target amount isn't reached, potential investors get their money back and the business goes away empty-handed. Such a public display of an idea risks others copying it. A company that has a limited network, no digital or social media presence, or a very complicated product will find it harder to crowdfund.

Some of the limitations of the crowdfunding project can be listed as:

- No algorithm that will reveal the money used to fund is legal money.
- No guarantee that all projects uploaded here will get 100% funded.
- If the creator hasn't protected his/her business idea with a patent or copyright, someone may see it on a crowdfunding site and steal the concept.
- Unusable payment gateway.

1.5 Development Methodology

1.5.1 Spiral Model

The Spiral Model represents a distinctive approach to software development, characterized by its emphasis on managing project risks. This model acknowledges that different projects come with unique risk patterns, and it offers a flexible framework for teams to adapt accordingly.

In the Spiral Model, the development process is guided by an iterative and cyclical approach. It involves several phases, including planning, risk analysis, engineering, and evaluation. The key feature of this model is its ability to integrate elements from various other process models as needed. For instance, a project might incorporate aspects of incremental development, the traditional waterfall model, or evolutionary prototyping, depending on its specific requirements and associated risks.

This adaptability makes the Spiral Model particularly useful in scenarios where uncertainties and risks are high, and a more flexible development approach is warranted. It allows teams to make informed decisions based on ongoing risk assessment and project evolution, ultimately leading to more effective and successful software development.

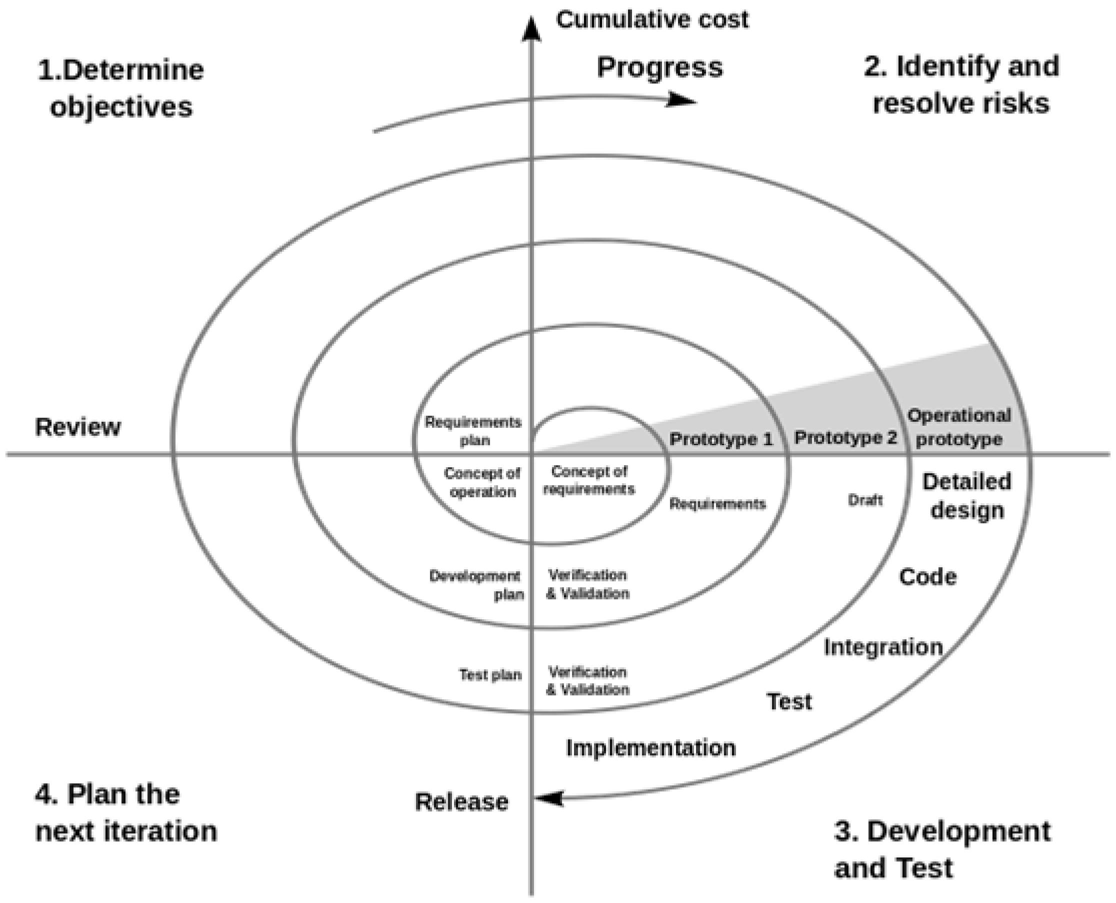


Figure 1.1 - Spiral Model

1.5.1.1 Determine Objectives

The objective is to develop a web-based crowdfunding platform to arrange funds for artistic and creative projects and startups. To make Spotlight, we used Django; the web development framework in Python, SQLite for database management and CSS3, and Bootstrap for the front-end development.

1.5.1.2 Identifying and resolving risk

In this phase, identification of potential risks while developing the project and figure out the optimal way to resolve them to eliminate or decrease the risk impact of this project is carried out along with estimation while monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun, and so on.

1.5.1.3 Development and Test

In this phase, the overall work and modules for the initial working of the application is completed. The prototype model is then tested along with its corresponding dataset in the database (which is stored locally). The datasets are separated and cleaned as well to remove missing data portions.

1.5.1.3.1.1 Prototype I

In the initial Prototype I, the authentication and security for the users to access the system through the proper login credentials (email and password; email verification to activate the account through an activation link) is built. To showcase the projects and to add the new projects to the system's database, modules are developed accordingly. Integrating the PayPal API using Sandbox accounts for the project fundings and storing their records into the database are in the pipeline.

1.5.1.3.1.2 Prototype II

In the second prototype, focus is on informing both the parties; creator and investor about the fundings being made. To make the fundings, authentication is required. The custom amount is taken from the user as input. By entering the login credentials from the PayPal Sandbox account, the investor can successfully pledge certain amount in the creator's concerned projects. As soon as the payment is made, the investor gets the alert in the browser mentioning the payment details. While the creator receives an automated email to inform about the payment. Besides that, the search functionality is also included in the homepage. All the transactions are reflected in the user's profile under Projects Created and Projects Funded.

1.5.1.3.1.3 Operational Prototype

Integrates the overall system from the Prototype I & Prototype II

1.5.1.4 Plan the next iteration

After building and testing, at the end of the first iteration, feedback report of previous model is created for any changes or modifications, the next iteration is then carried out addressing the modifications.

1.6 Report Organization

Chapter 1: Puts brief emphasis on background and overview, problem statement, objectives, scopes, development methodology and limitations of the project.

Chapter 2: Defines and describes background study and literature review.

Chapter 3: Presents the system analysis and design including requirement analysis and feasibility analysis.

Chapter 4: Presents the implementation and testing, it clarifies the system workflow and provides what tools are used.

Chapter 5: Presents the conclusion and recommendation.

Chapter 6: References taken

Chapter 7: Appendix which includes certain source codes and screenshots

Chapter 2: Background Study and Literature Review

2.1 Background Study

In today's digital age, crowdfunding has become a convenient way to gather funds for various purposes, such as business ventures, nonprofit initiatives, or personal financial needs. To help people make informed choices, we conducted research and evaluated several crowdfunding platforms. Our selection criteria included factors like user-friendliness, a track record of successful fundraising, pricing options, integration with social media, and more.

2.1.1 Kickstarter

Kickstarter stands out as one of the most prominent players in the crowdfunding arena. It has gained renown for its role in assisting tech innovators and creative entrepreneurs in raising funds for their projects, allowing them to circumvent the traditional avenues of securing loans or venture capital. Since its inception in 2009, Kickstarter has facilitated the raising of a staggering $6.5 billion. This platform encompasses a diverse range of verticals, encompassing everything from arts and film to publishing.

Kickstarter employs a straightforward approach to fundraising. Creators begin by establishing a funding goal and a defined timeframe for achieving it. However, it's important to note that Kickstarter requires campaign approval before launch. A distinctive feature of Kickstarter is its all-or-nothing fundraising model, meaning that project creators only receive their funds if they meet or exceed their funding goal. This also ensures that a funder's credit card is charged only if the campaign successfully reaches its objective. Kickstarter charges a 5% fee on top of transaction processing charges, which typically range from 3% to 5%. Additionally, there is a 14-day waiting period to access the funds after a successful campaign. [2]

2.1.2 Indiegogo

Indiegogo, much like Kickstarter, serves as a platform for creators across various domains, including tech innovations, creative works, and community projects. It shares similarities with Kickstarter but distinguishes itself by offering both fixed and flexible funding options. Creators can choose between these two funding models based on their project's specific needs.

In the case of fixed funding, this option is ideal when a project requires a predetermined amount of funds for success. Conversely, flexible funding allows creators to retain any funds raised, regardless of whether they reach their initial funding goal or not. Notably, Indiegogo does not impose fixed funding fees for campaigns that fall short of their objectives, in contrast to Kickstarter. The fee structure on Indiegogo includes a 2.9% processing fee and an additional $0.30 per transaction. The minimum funding goal for both funding types starts at $500. [3]

2.1.3 Patreon

Patreon caters primarily to digital creatives, encompassing content creators like YouTubers, podcasters, and bloggers. Unlike traditional crowdfunding platforms that facilitate one-time donations, Patreon operates on a subscription model. Here, patrons commit to contributing a set amount of money either monthly or per creation by the content creator.

Patreon fosters a unique relationship between artists and their fan base. Creators can provide exclusive content to their Patreon subscribers as an incentive for ongoing funding. However, the effectiveness of Patreon is closely tied to how consistently creators share content on their personal platforms. Patrons retain the option to cancel their subscriptions if content production lags.

Patreon offers creators three plan tiers, each of which takes a percentage of the monthly income generated on the platform. These tiers include Lite (5%), Pro (8%), and Premium (12%). Additionally, Patreon charges payment processing fees that vary based on the selected plan tier and the patron's chosen payment method. While Patreon boasts over 8 million active patrons and 250,000 creators, one drawback is that it may not provide as much marketing exposure for creators compared to platforms like Indiegogo or Kickstarter, which have dedicated sections for potential donors to explore. [4]

2.1.4 GoFundMe

GoFundMe is a well-known crowdfunding platform recognized for its use in charitable campaigns and support for individuals with medical needs or emergency relief requirements. It operates under a model where you retain all the funds raised, regardless of whether you meet your funding goal or not. This differs from all-or-nothing platforms like Kickstarter.

One of GoFundMe's notable features is its minimal fee structure. It imposes a 2.9% processing fee on each donation, along with an additional $0.30 per transaction. Notably, for campaigns based in the United States, GoFundMe does not charge personal campaign funding fees. This fee structure makes GoFundMe an attractive option for individuals and small-budget projects.

While GoFundMe has seen numerous successful campaigns, such as the Las Vegas Victims Fund and the Time's Up Legal Defense Fund, it's important to acknowledge that conventional startups may not generate as much capital on this platform. Additionally, it's essential to be aware that only about one in ten campaigns typically achieve full funding on GoFundMe.

2.2 Literature Review

Kickstarter is widely recognized as a crowdfunding giant. It's renowned for its role in aiding creators and innovators in securing funds for their projects, whether they're in the realms of technology, creative arts, or various other fields. Notably, Kickstarter operates on an "all-or-nothing" model, which means that campaigns must meet their specified funding goal to access the funds pledged. This also ensures that backers' credit cards are only charged if the campaign successfully reaches its objective. [2]

Indiegogo, in contrast, shares similarities with Kickstarter in its diverse project offerings. It's a hub for tech, creative endeavors, and community projects. What sets it apart is its flexibility; Indiegogo doesn't strictly adhere to the "all-or-nothing" model. This means that even if a campaign falls short of its funding target, the collected funds can still be accessed. Moreover, Indiegogo doesn't impose fixed funding fees for campaigns that don't meet their goals, setting it apart from Kickstarter. [3]

Patreon occupies a unique space in the crowdfunding landscape, catering primarily to digital creators like YouTubers, podcasters, and bloggers who engage with their audiences regularly. Unlike traditional crowdfunding platforms, Patreon operates on a subscription basis. Supporters commit to contributing a set amount of money either monthly or per creation by the content creator. This approach fosters ongoing relationships between creators and their patrons, but creators must maintain content production to keep their patrons engaged. [4]

GoFundMe serves as a versatile platform, ideal for both charitable initiatives and smaller budget projects with various funding needs, including medical expenses and emergency relief. One of its key advantages is its flexibility. Campaign creators retain all funds raised, regardless of whether they meet their funding goal. This flexibility makes GoFundMe an appealing choice for projects with more modest funding objectives. Moreover, GoFundMe maintains a straightforward fee structure, including a 2.9% processing fee per donation, along with a $0.30 transaction fee. [5]

Chapter 3: System Analysis and Design

3.1 System Analysis

Given that the application involves designing and implementing a software system, it's important to mention the development model used. This choice is crucial because it sets the framework for how the entire project is organized and executed. The development model determines how tasks are managed, progress is tracked, and stakeholders are involved throughout the project's life. In essence, it's the roadmap that ensures a smooth and efficient software development process. Therefore, let's discuss the development model chosen for this project and its implications.

3.1.1 Requirements Analysis

Requirements analysis is a crucial step for determining the success of a system or software project. Requirements are generally split into two types:

i. Functional Requirements
ii. Non-functional Requirements

3.1.1.1 Functional Requirements

This section provides the requirement overview of the application. Various modules implemented by the system are:

- User Module
  - Users can register and login
  - Users can create project
  - Users can invest/fund
- Admin Module
  - Admin can login
  - Admin can view projects
  - Admin can view users
- Login Module
  - Only verified registered users can login to the system.

Use Case Diagram

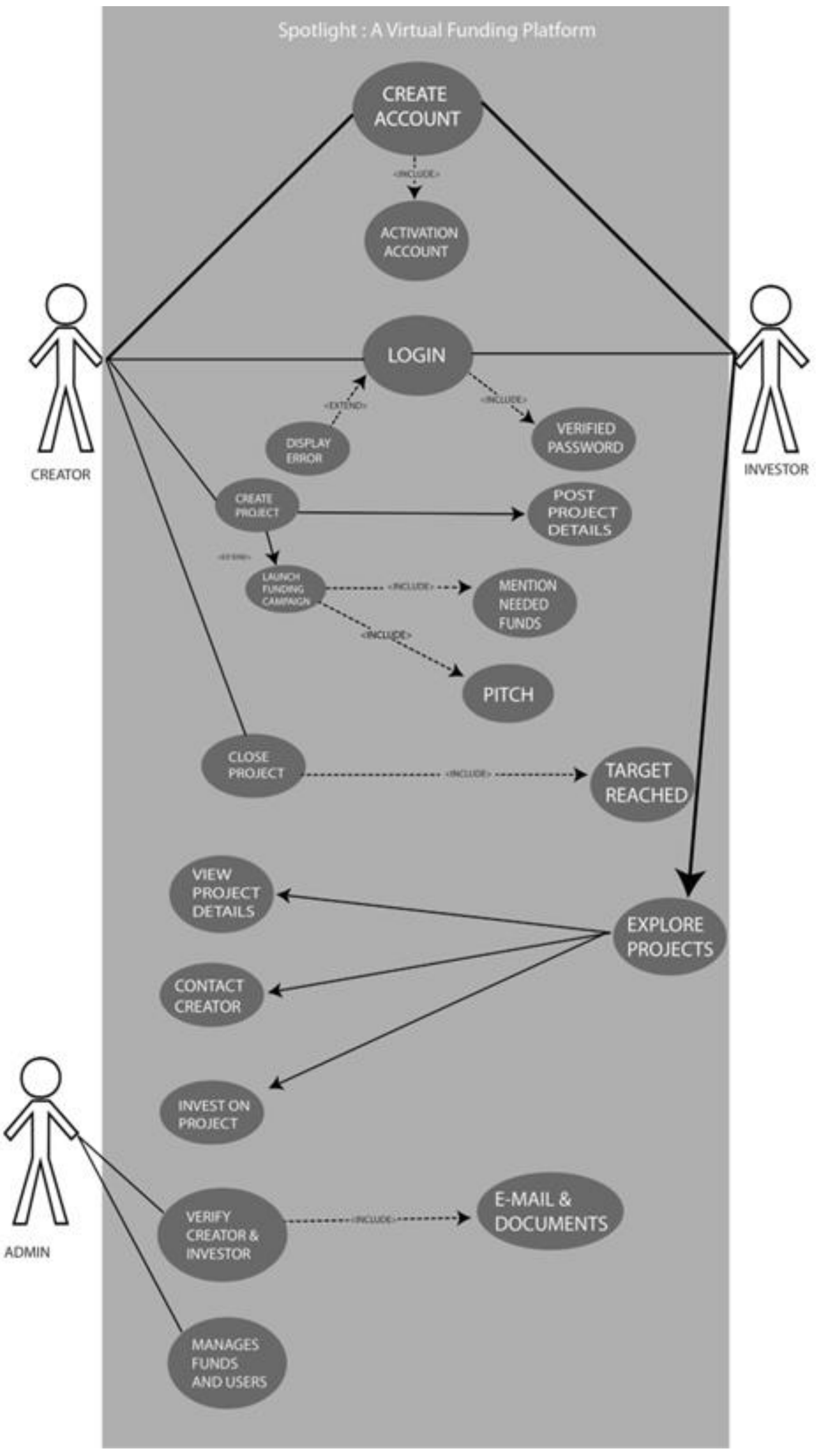


Figure 3.1 - UML Use Case Diagram

3.1.1.2 Non-functional Requirements

Non-functional requirements of the system are identified as efficiency requirements. reliability requirements, usability requirements, and implementation requirements. The non-functional requirements included in the project are:

- Efficiency Requirements
  The efficiency of a software system refers to how well it handles capacity, throughput, and response time. With the deployment of the virtual crowdfunding platform, both users and admin may easily access the app and users are able to create projects/fund projects.
- Reliability Requirements
  The degree to which the software system regularly executes the stated functions without failures is referred to as reliability. User registration, login, project creation, etc. inquiry is all performed accurately by the system.
- Usability Requirements
  The system's usability criteria state how simple it must be to use. The system was created in a user-friendly environment so that the users and administrators could easily and successfully complete various activities in the system.
- Implementation Requirements
  The process of turning strategies and plans into actions to achieve strategic objectives and goals is known as implementation. The frontend was created using HTML, CSS, JavaScript, Bootstrap, Jinja2, etc. with Python Django framework serving as the server-side programming language for database connectivity at the backend, SQLite is utilized.

3.1.2 Feasibility Analysis

a. Technical Feasibility

These include hardware, software, and technologies. The suggested system is technically possible because it requires access to use of an internet browser.

b. Operational Feasibility

Reliability, maintainability, usability, and supportability are among them. The suggested system is operationally practical since it is reliable for all the types of users, regardless of whether they are computer literate. For a small to large-scale organization, the proposed system is supported. It is simple and straightforward to use.

c. Economic Feasibility

The project resource was freely available, and no additional obligation required. The creation of this software does not necessitate the use of expensive hardware or software.

d. Schedule Feasibility

Among various phases of the project development took longer time as a new framework and programming language was used.

Table 3.1 - Time Schedule

Task	Start Date	Duration (in days)	End date
Requirement Analysis	2023 May 15	12	2023 May 30
System Design	2023 May 30	15	2023 June 15
Implementation	2023 June 13	40	2023 July 23
Testing	2023 July 24	10	2023 August 3
Maintenance	2023 August 4	-	-
Documentation	2023 August 5	18	2023 August 23

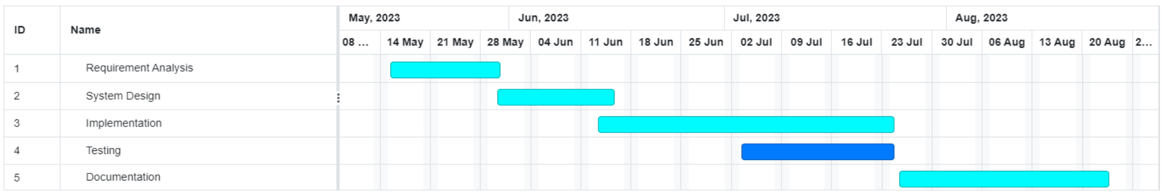


Figure 3.2 - Gantt Chart

3.1.3 Data Modeling: ER diagram

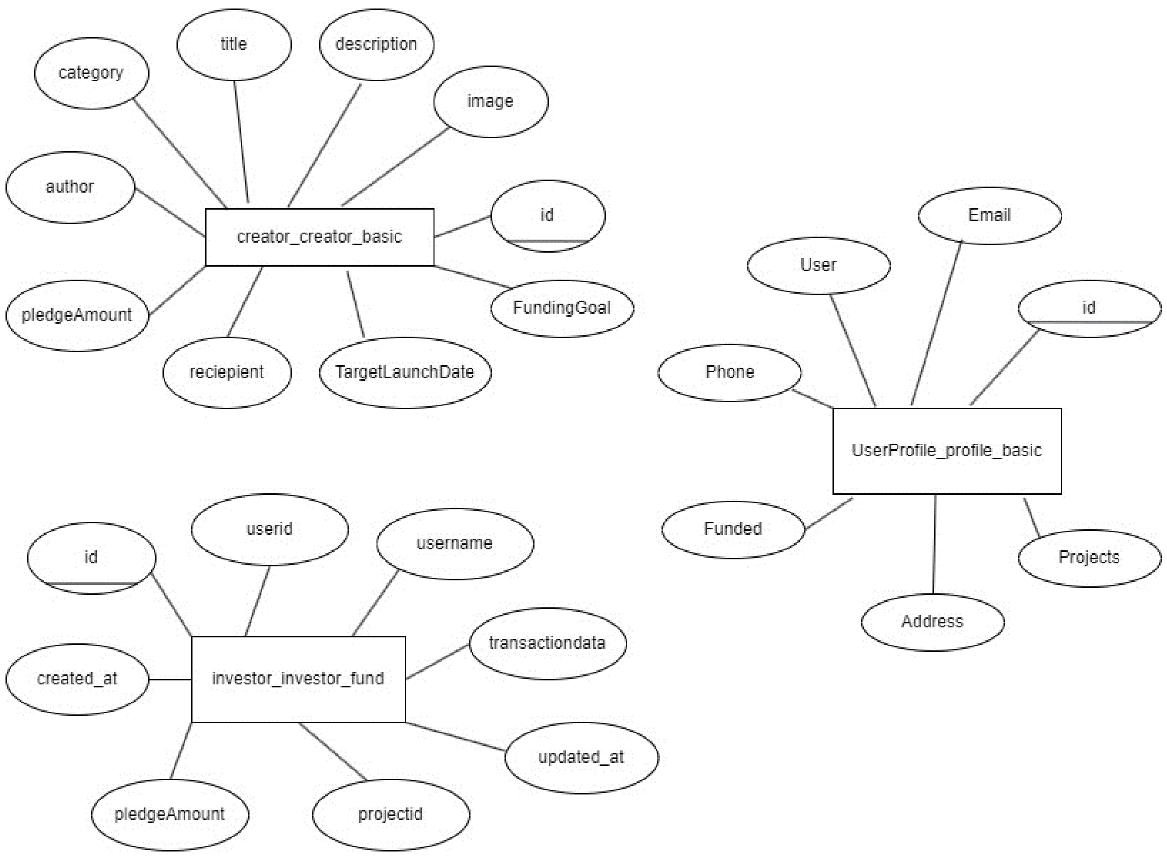


Figure 3.3 - ER Diagram

3.1.4 Process Modeling

3.1.4.1 Level 0 DFD

A process model describes the flow of work or activities, usually in a graphic format, that contributes to accomplishing a specific goal. Process models are typically used to represent and analyze a series of activities that occur repeatedly and on a regular basis.

A data flow diagram (DFD) is a graphical representation using a standardized set of symbols and notations to describe a business's operations through data movement.

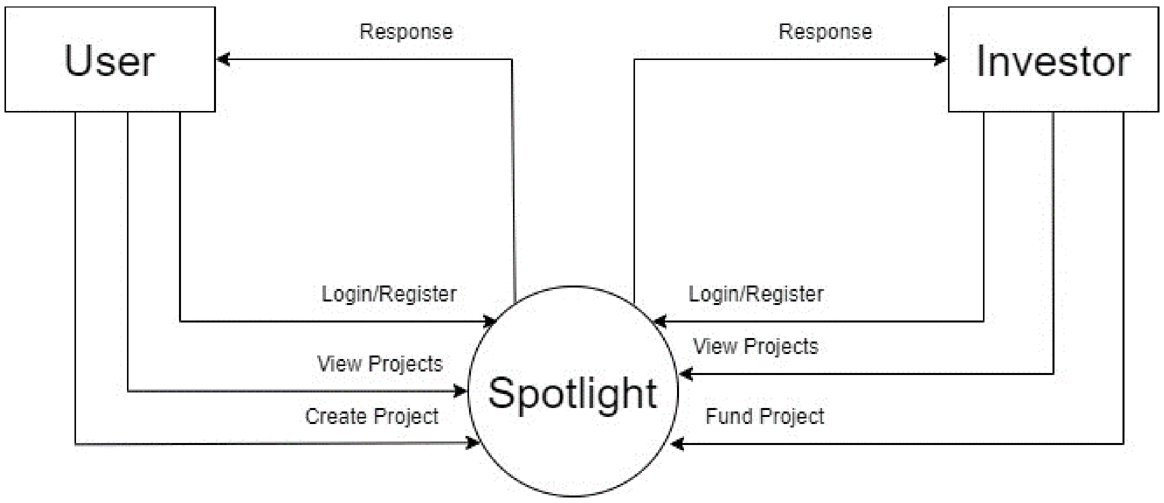


Figure 3.4 – DFD level 0

3.1.4.2 Level 1 DFD

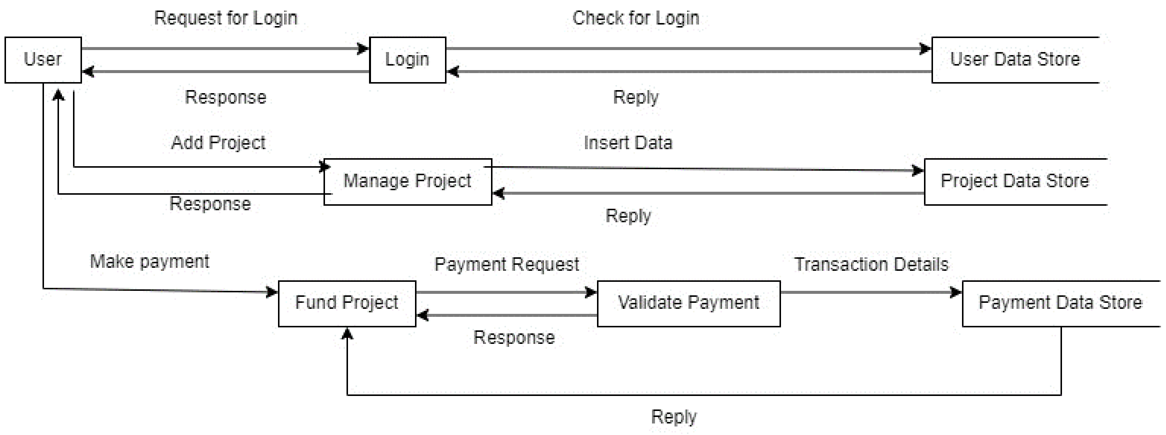


Figure 3.5 - DFD Level 1

3.2 System Design

3.2.1 Architectural Design

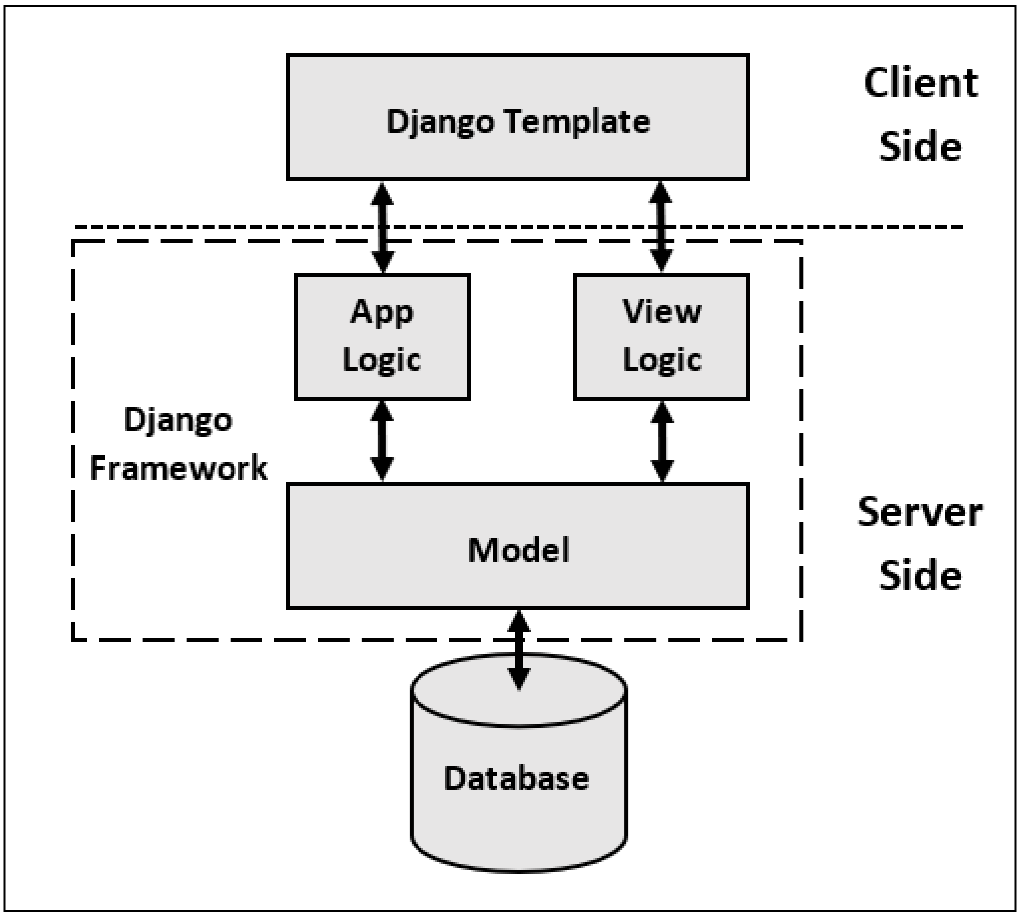


Figure 3.6 - Architectural Design

3.2.2 Database Schema Design

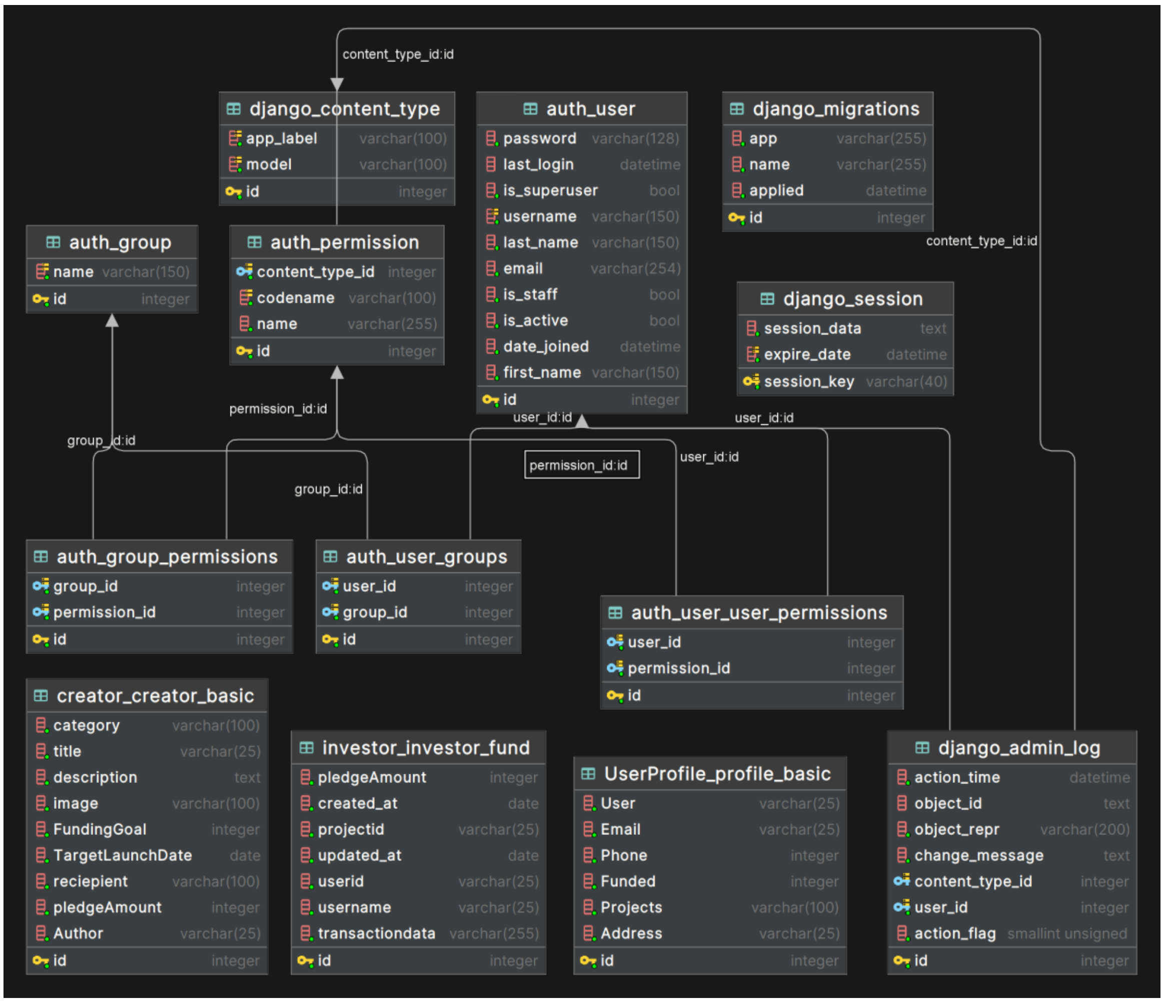


Figure 3.7 - Database Schema Design

3.2.3 Interface Design

Interface is one of the most important parts of the system as it determines how easy if is for a new user using the system to understand the different components listed and navigate through them in order to achieve the intended goal of using the system. The interface of this system will be presented according to following representations:

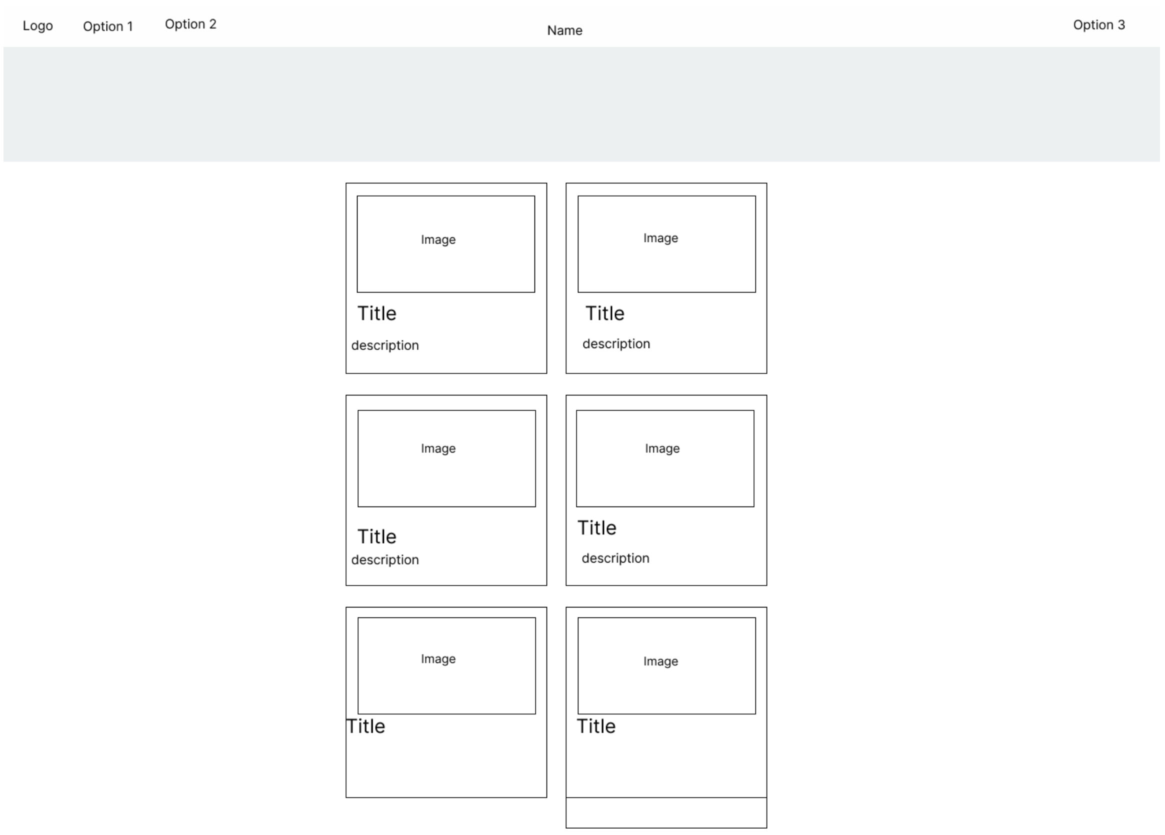


Figure 3.8 - Home View

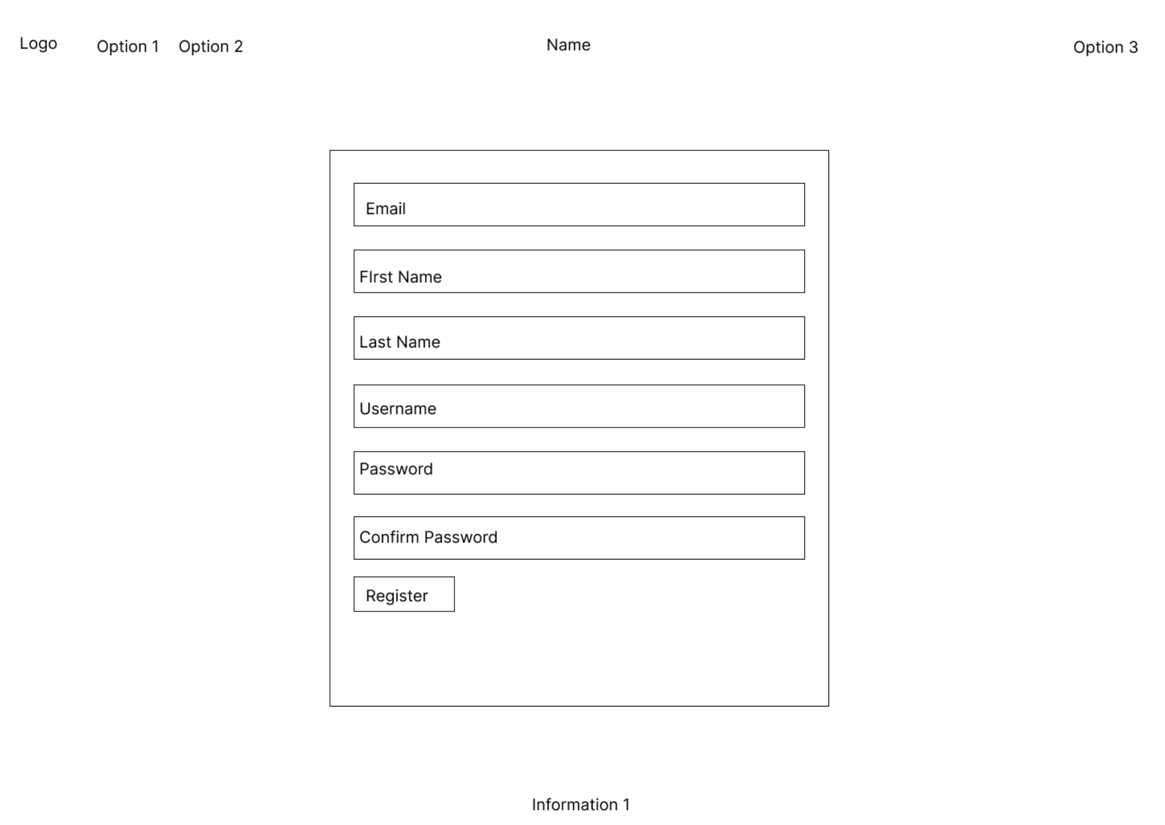


Figure 3.9 - Register View

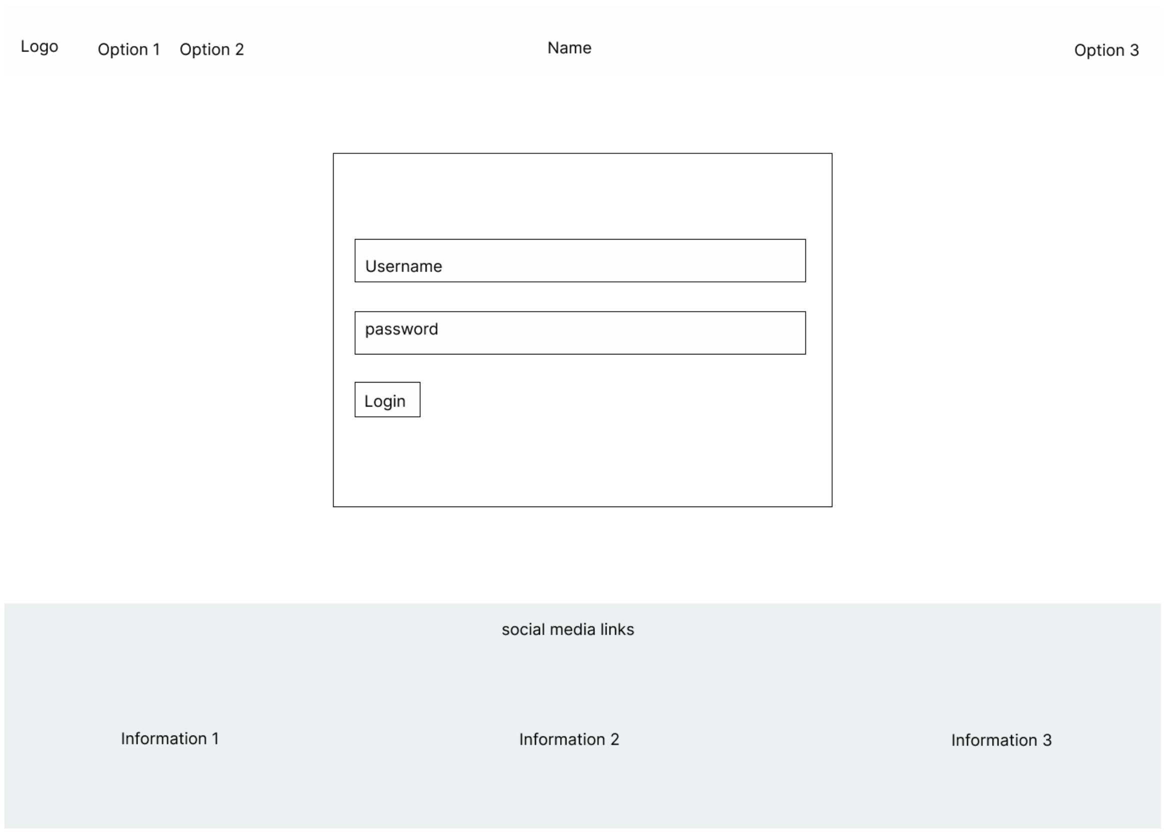


Figure 3.10 - Login View

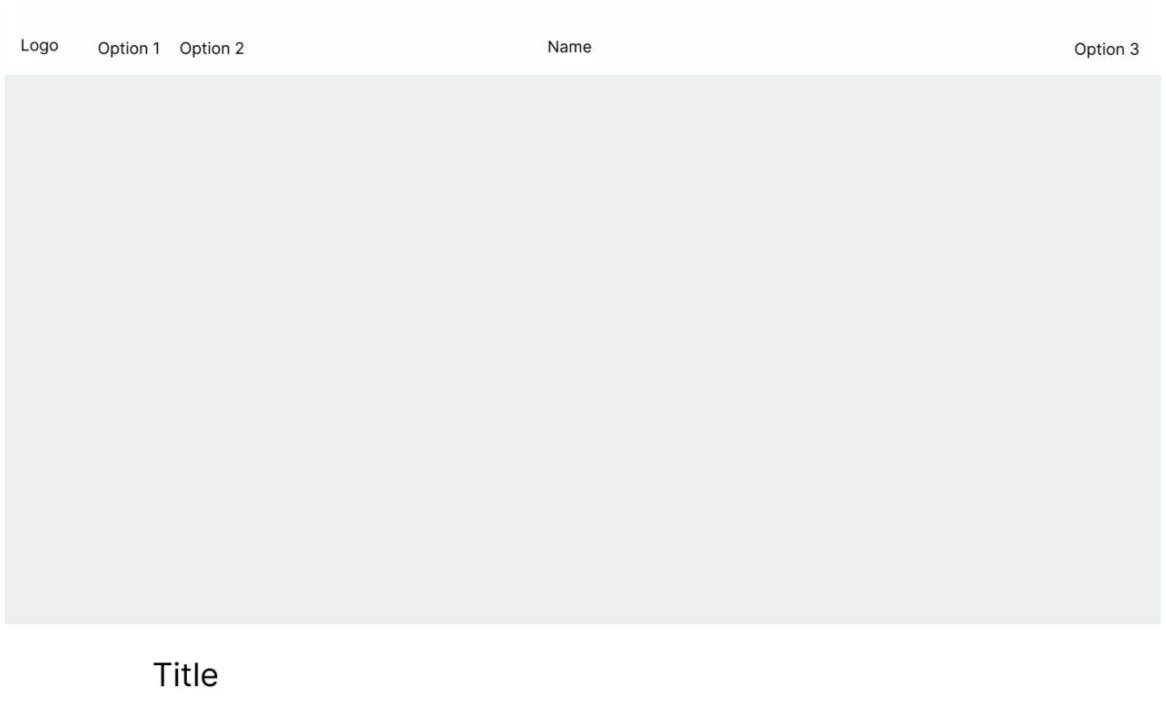


Figure 3.11 - Default Empty View

3.3 Algorithm details

3.3.1 Content-based filtering

A recommendation system technique called content-based filtering is used in information retrieval and e-commerce to suggest products to users based on the qualities and attributes of those products. One of the primary methods for creating customized recommendation systems is this. When making suggestions, content-based filtering focuses on the characteristics or content of the items and the user's preferences.

Here, content-based filtering is applied for recommending user with similar projects to the currently viewing projects based on cosine similarity. The cosine similarity is calculated from vectorized texts taken from description of projects created, the recommendation algorithm used is mentioned as below and source code of implementation is shown in appendix.

Recommendation algorithm

Step 1. Retrieve data from the database using creator_Basic.objects.all().order_by("-TargetLaunchDate").values("id", "description").

Step 2. Fill missing description values in the DataFrame with an empty string.

Step 3. Construct a vocabulary based on unique words in the descriptions.

Step 4. Create TF-IDF vectors for descriptions by looping over the descriptions and calling text_to_tfidf_vector.

Step 5. Calculate the cosine similarity matrix by iterating over the descriptions and calling cosine_similarity to fill the similarity values in the matrix.

Step 6. Create a DataFrame for the cosine similarities.

Step 7. Create a mapping between IDs and DataFrame indices.

Step 8. Return the cosine similarity DataFrame, the mapping, and the original DataFrame.

Step 9. Call custom_recommendation_engine.recommendation() to get the similarity matrix, mapping, and the projects' DataFrame.

Step 10. Retrieve the index of the input project using mapping[input_id].

Step 11. Calculate the similarity scores between the input project and all other projects in the matrix.

Step 12. Sort the similarity scores in descending order.

Step 13. Select the top number_of_recommend projects with the highest similarity scores, excluding the input project itself.

Step 14. Extract the indices of the recommended projects.

Step 15. Return a dictionary containing the recommended project IDs and their corresponding similarity scores.

Formula for cosine similarity:

$$\text{Cosine Similarity}(\mathrm{A}, \mathrm{B}) = \frac{\mathrm{A} \cdot \mathrm{B}}{||\mathrm{A}|| \cdot ||\mathrm{B}||}$$

Working:

From the description of a project, English stop words are removed. Stop words are basically a set of commonly used words in any language, not just English and are the words that're not relevant for recommendation purpose. Then, vectors are calculated from vocabulary of words derived from the normalized description. Note that the description are non nulls. Once the vectors are derived, a cosine similarity matrix is generated, the matrix is always square with row and columns equal to the number of descriptions present. The cosine similarity between the vectors is calculated and filled in the earlier initialized matrix. The obtained similarity scored present in first row (also in first column but for easier manipulation we use first row) is mapped to respective projects. The mapping is sorted based on similarity scores from 1 to 0 where 1 being the most similar and 0 being the least. The most similar ones are presented to the template for showing it to the user. The similarity score is always visible to the user.

Chapter 4: Implementation and Testing

4.1 Implementation

4.1.1 Tools used

4.1.1.1 Python

The Python programming language is used to build this site as it is a dynamic, interpreted (bytecode-compiled) language used for web development (server-side), software development, mathematics, system scripting, and various other domains. There are no type declarations of variables, parameters, functions, or methods in source code. Through Python, the connection between the database systems can be established. Usually, rapid development or for production-ready software are developed using Python.

4.1.1.2 SQLite

SQLite is a default database structure of Django framework, written in the C programming language that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. It is not a standalone app; rather, it is a library that software developers embed in their apps. SQLite is the most widely deployed database engine, as it is used by several of the top web browsers, operating systems, mobile phones, and other embedded systems. It generally follows PostgreSQL syntax but does not enforce type checking by default.

4.1.1.3 Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. As it is our elective subject in our current semester and it takes care of much of the hassle of web development, we can focus on developing the project without needing to reinvent the wheel. The best part is that it’s free and open source. Also, we want to implement the theoretical learnings into real world project experience.

4.1.1.4 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

4.1.1.5 NLTK

Natural Language Toolkit (NLTK) is a Python library for working with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources, such as WordNet. NLTK includes a wide range of text processing libraries for tasks like tokenization, stemming, tagging, parsing, and semantic reasoning. It is a popular choice among researchers and developers for tasks related to natural language processing (NLP). This has been used to download to download stopwords and tokenize words from description of user created projects.

4.1.2 Implementation Details of Modules

```
def recommendation():
    df = pd.DataFrame(
        creator_Basic.objects.all()
        .order_by("-TargetLaunchDate")
        .values("id", "description")
    )
    df["description"] = df["description"].fillna("")
    vocabulary = {
        word: idx
        for idx, word in enumerate(set(" ".join(df["description"]).split()))
    }
    tfidf_vectors = [
        text_to_tfidf_vector(description, vocabulary)
        for description in df["description"]
    ]
    n_descriptions = len(df)
    cosine_sim_matrix = np.zeros((n_descriptions, n_descriptions))
    for i in range(n_descriptions):
        for j in range(i, n_descriptions):
            similarity = cosine_similarity(tfidf_vectors[i], tfidf_vectors[j])
            cosine_sim_matrix[i, j] = similarity
            cosine_sim_matrix[j, i] = similarity

    cosine_sim_df = pd.DataFrame(cosine_sim_matrix)
    mapping = pd.Series(df.index, index=df["id"])
    return cosine_sim_df, mapping, df
```

The above shown code snippet is the main recommendation engine of this system. The first line of code fetches id and description of all the objects present in database sorted in descending order by TargetLaunchDate and stores it in a pandas dataframe. A Pandas DataFrame is a two-dimensional, size-mutable, tabular data structure with labeled axes (rows and columns). If description of any project is null, the null place is filled with an empty string. A vocabulary of non-repeating words is created and mapped with an index.

Non repeat of words are ensured using the set data structure which only eliminates redundancy.

```
def normalize_document(text):
    wpt = nltk.WordPunctTokenizer()
    stop_words = nltk.corpus.stopwords.words("english")
    text = re.sub(r"[^a-zA-Z\s]", "", text.lower(), re.I | re.A)
    text = text.strip()
    tokens = wpt.tokenize(text)
    filtered_tokens = [token for token in tokens if token not in stop_words]
    text = " ".join(filtered_tokens)
    return text
```

With the above code snippet, non-alphabets and spaces are substituted with empty string and input parameter *text* is also converted to lowercase. Then leading or trailing spaces from *text* is removed. The resulting *text* is then tokenized. Tokenize basically means to break down large body of data into smaller ones. The large sentences that *text* was is converted to a list of words, and stop words are removed from tokens creating a filtered token. A sentence is put back together joining the words in filtered tokens with a space.

```
def text_to_tfidf_vector(text, vocabulary):
    words = normalize_document(text).split()
    vector = np.zeros(len(vocabulary))
    for word in words:
        if word in vocabulary:
            vector[vocabulary[word]] += 1
    return vector
```

Then the text is vectorized. In the above code snippet, a function called text_to_tfidf_vector is declared which takes input parameter text and vocabulary, here text is a string and vocabulary is a python dictionary. The input parameter text is normalized using the normalize_document function mentioned earlier and split at each space assigning a list of words present in text to the variable word. Another variable called vector is initialized which is an array of zeros of length of vocabulary i.e., the size of array is determined by number of elements in the dictionary *vocabulary*. The array is then modified after iterating though each word of input parameter text, if the word is present in the vocabulary, the vector array at position that's equal to value of key word present in dictionary vocabulary is incremented by 1. The function returns a numpy array called vector. Numpy is preferred over python list here for its performance.

```
def cosine_similarity(vector_a, vector_b):
    def calculate_norm(vector):
        return math.sqrt(sum(x**2 for x in vector))

    dot_product = np.dot(vector_a, vector_b)
    norm_of_vector_a = calculate_norm(vector_a)
    norm_of_vector_b = calculate_norm(vector_b)
    similarity = dot_product / (norm_of_vector_a * norm_of_vector_b)
    return similarity
```

The function in above code snippet is equivalent to the formula for the calculation of cosine similarity:

$$\text{Cosine Similarity}(\mathrm{A}, \mathrm{B}) = \frac{\mathrm{A} \cdot \mathrm{B}}{||\mathrm{A}|| \cdot ||\mathrm{B}||}$$

For numerator, the first line of the function cosine_similarity calculates the dot product between the two vectors.

The third and fourth lines calculate the norms of each vector. A norm, in the context of mathematics and linear algebra, is a mathematical concept used to measure the size or length of a vector. As dependency for cosine similarity, L2 norm also called Euclidean Norm is calculated. Norm is calculated using inner function norm_calculation which returns the square root of sum of squares of each element in the vector. Then, the similarity is equal to dot product of the two vectors divided by the dot product of norms of each vector.

Back to the main function, a two-dimensional square array of zeros of size of dataframe is initialized. Then, iterating through each dimension of the two-dimensional array, the cosine similarity of the two elements of the represented matrix, is calculated, where vector A is the *i* element of the matrix and vector B is the *j* element of the matrix. The calculated similarity is assigned to two the matrix at index (i, j) and (j, i). This makes the first row and column of matrix equal at same index i.e., first element of row equals to first element of column and so on. The function then creates a dataframe of the similarity matrix and maps each index of the newly created dataframe to the original dataframe which contains all the projects. The function returns a tuple of cosine similarity matrix, mapped dataframe and original dataframe.

```
def recommendation():
    df = pd.DataFrame(
        creator_Basic.objects.all()
        .order_by("-TargetLaunchDate")
        .values("id", "description")
    )
    df["description"] = df["description"].fillna("")
    vocabulary = {
        word: idx
        for idx, word in enumerate(set(" ".join(df["description"]).split()))
    }
    tfidf_vectors = [
        text_to_tfidf_vector(description, vocabulary)
        for description in df["description"]
    ]
    n_descriptions = len(df)
    cosine_sim_matrix = np.zeros((n_descriptions, n_descriptions))
    for i in range(n_descriptions):
        for j in range(i, n_descriptions):
            similarity = cosine_similarity(tfidf_vectors[i], tfidf_vectors[j])
            cosine_sim_matrix[i, j] = similarity
            cosine_sim_matrix[j, i] = similarity

    cosine_sim_df = pd.DataFrame(cosine_sim_matrix)
    mapping = pd.Series(df.index, index=df["id"])
    return cosine_sim_df, mapping, df
```

```
def recommend(input_id, number_of_recommend=3):
    (
        similarity_matrix,
        mapping,
        projects,
    ) = custom_recommendation_engine. recommendation()
    project_index = mapping[input_id]
    similarity_score = list(enumerate(similarity_matrix[project_index]))
    similarity_score = sorted(
        similarity_score, key=lambda x: x[1], reverse=True
    )
    similarity_score = similarity_score[1 : number_of_recommend + 1]
    project_indices = [i[0] for i in similarity_score]
    return {
        "recommendation_list": projects["id"].iloc[project_indices],
        "similarity_score": similarity_score,
    }
```

The above snippet is the driver of the recommendation engine. It takes in input_id which is the project id to recommend based of and number_of_recommended which tells how many recommendations to send to the view. At first, three variables, similarity_matrix, mapping and projects are initialized which takes in values returned by the recommendation function mentioned earlier. This mapping is a Series object. The values for this Series came from the existing index of the DataFrame df, and the index labels for this Series is derived from the id column of the DataFrame. In other words, it creates a mapping where the "id" values in the DataFrame become the index labels in the Series, and the corresponding DataFrame row numbers (indices) become the values in the Series. From the series, the index of tuple where the project is in, is fetched and stored in project_index. Then, the row of data at index project_index is enumerated and cast to a list eventually being stored at variable similarity score which is then sorted based on the element at index 1 in descending order. The list is then sliced such that the first element is removed and the end value is equals to number_of_recommended + 1. The first element is removed as the project will have 100% similarity with itself. After sorting, the project indices are stored in a list where it'll be returned by this function as a dictionary after indexing the id column of the DataFrame projects using the project_indices variable which is an iterable of indices that corresponds to the recommended projects.

4.2 Testing

4.2.1 Unit Testing of basic modules

Table 4.1 - Test Case for Signup

Test Case	Result
Signup with invalid email	Shows error message saying email is invalid.
Signup with valid email	Says sign up successful and shows message asking to activate.
Signup with taken email	Shows message saying email is already taken.

Table 4.2 - Test Cases for Login

Test Case	Result
Login attempt with unregistered account	Error message shown
Login attempt with registered user whose account isn't activated	Error message shown
Login attempt with registered user whose account is activated	Successful login and redirection to home page

Table 4.3 - Test Case for Creating Project

Test Case	Result
Creation with valid details known to admin	Post Created
Creation with project with missing details	Tells that details are missing
Creation with missing date field	Logs an error in console while browser says something went wrong

Creation with missing image	Server throws an exception

4.2.2 System Testing

The following things have been validated from system testing:

- The server starts without any error on port 8000.
- The templates used for UI are loaded successfully.
- Media files are being stored in /media directory as intended.

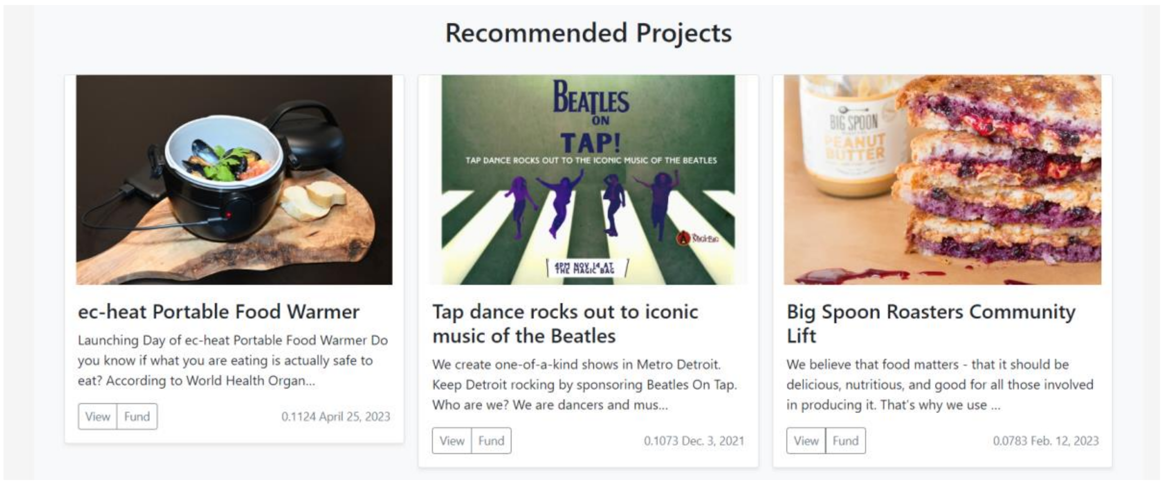
However, the payment gateway is not working as expected as it was misjudged during requirements analysis. PayPal API integration was made but the account used for it was banned and there's no more means to test this system.

4.2.2.1 Testing the Recommendation System from User Perspective

Test Case:

Firstly, a project is created with unique description that doesn't match with any projects created till date. The description of this project is set as "Education is not just a fundamental right; it is the cornerstone of progress and development. However, many children and youth worldwide are denied access to quality education, often due to economic disparities, discrimination, or political instability. One cause that has been championed by countless individuals and organizations is the empowerment of youth through education. This cause aims to break down barriers, provide opportunities, and equip young people with the knowledge and skills they need to create a brighter future for themselves and their communities."

When the recommended projects are viewed currently, the top ones are project 59, 37, 61 and 57 with similarity score of 0.1124, 0.1073, 0.0783, 0.0714 respectively. Judging based on similarity scores the projects are not very similar.


Recommended Projects

ec-heat Portable Food Warmer
Launching Day of ec-heat Portable Food Warmer Do you know if what you are eating is actually safe to eat? According to World Health Organ...

0.1124 April 25, 2023

Tap dance rocks out to iconic music of the Beatles
We create one-of-a-kind shows in Metro Detroit. Keep Detroit rocking by sponsoring Beatles On Tap. Who are we? We are dancers and mus...

0.1073 Dec. 3, 2021

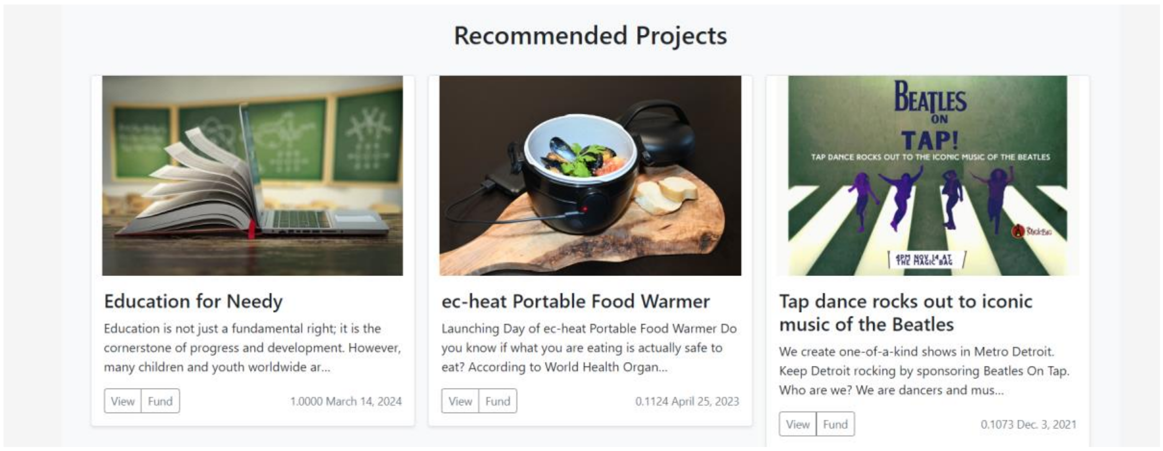
Big Spoon Roasters Community Lift
We believe that food matters - that it should be delicious, nutritious, and good for all those involved in producing it. That's why we use ...

0.0783 Feb. 12, 2023

Figure 4.1 - Recommendations for mentioned project

Now, a new project is added with description "Education is not just a fundamental right; it is the cornerstone of progress and development. However, many children and youth worldwide are denied access to quality education, often due to economic disparities, discrimination, or political instability. One cause that has been championed by countless individuals and organizations is the empowerment of youth through education. This cause aims to break down barriers, provide opportunities, and equip young people with the knowledge and skills they need to create a brighter future for themselves and their communities."

The description is same as the previous one, if not same, very similar. The previously created project should appear as our first recommendation with a very high similarity score.


Recommended Projects

Education for Needy
Education is not just a fundamental right; it is the cornerstone of progress and development. However, many children and youth worldwide ar...

1.0000 March 14, 2024

ec-heat Portable Food Warmer
Launching Day of ec-heat Portable Food Warmer Do you know if what you are eating is actually safe to eat? According to World Health Organ...

0.1124 April 25, 2023

Tap dance rocks out to iconic music of the Beatles
We create one-of-a-kind shows in Metro Detroit. Keep Detroit rocking by sponsoring Beatles On Tap. Who are we? We are dancers and mus...

0.1073 Dec. 3, 2021

Figure 4.2 - Recommendations for created project

As observed, two of the recommendations are same as before, i.e., project 59 and 37 with similarity score of 0.1124 and 0.1073. As expected, the project that was created earlier about education is the top recommendation with similarity score of 1.

Chapter 5: Conclusion and Future Recommendations

5.1 Conclusion

The Spotlight is now at the initial phase with its beta version having most of the basic functionalities discussed before. All the modules have been working after integrating and are ready for the demo. As the features add up the level of complexity has been increasing as well. However, it is not complete with the ideas we have put through and might need more improvisation in the coming days as well. This makes us think about the future extensions that we are going to implement on this website.

Some of the features of Spotlight are mentioned below.

- Authentication system including user registration, user activation through Welcome email, and user password reset
- Implementation of the MVT architecture.
- Users can create their own project along with its description and the project files.
- User and Non-user can see the posted project in the home page of our website
- Can toggle between Recently added.
- Project or View all Projects Progress Meter to Easily Track Campaign Progress

5.2 Further Work

- To improve user interface
- Add extra project information
- Modify user profile
- Responsive design that can adapt to user's device screen size
- Register and sign in using the social accounts
- Search Engine
- Better and usable payment gateway
- Add validation to amount

Chapter 6: References

[1] B. Boehm, "Spiral Development: Experience, Principles, and Refinements," Software Engineering Institute. CMU/SEI-2000-SR-008., July 2000.

[2] "KickStarter," [Online]. Available: https://www.kickstarter.com/. [Accessed 2023].

[3] "IndieGoGo," [Online]. Available: https://www.indiegogo.com/. [Accessed 2023].

[4] "Patreon," [Online]. Available: https://www.patreon.com/. [Accessed 2023].

[5] "GoFundMe," [Online]. Available: https://www.gofundme.com/. [Accessed 2023].

[6] "CrowdCube," [Online]. Available: https://www.crowdcube.com/. [Accessed 2023].

[7] E. Pong, "Floship," 23 September 2016. [Online]. Available: https://www.floship.com/blog/_7-potential-problems-with-crowdfunding/. [Accessed 2023].

[8] T. Bouhsine, "Research Gate," April 2020. [Online]. Available: https://www.researchgate.net/publication/343988849_Design_And_Full_Stack_Development_Of_A_Crowdfunding_Platform_Sahem. [Accessed 2023].

[9] "Vilmate," [Online]. Available: https://vilmate.com/blog/how-to-build-a-crowdfunding-platform-like-kickstarter/.

[10] D. B. and K. Z., "RubyGarage," 26 August 2020. [Online]. Available: https://rubygarage.org/blog/how-to-start-a-crowdfunding-website. [Accessed 2023].

[11] Floship, "Medium," [Online]. Available: https://medium.com/@floship/problems-with-crowdfunding-7-hazards-to-watch-out-for-83ecea54144b. [Accessed 2023].

[12] S. R. Nepali, "Journal of Business," [Online]. Available: https://journalofbusiness.org/index.php/GJMBR/article/download/3411/3312. [Accessed 2023].

[13] J. Gera and H. Kaur, "Science Direct," [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959518301152. [Accessed 2023].

[14] "Hiteri," [Online]. Available: https://hiteri.org/.

Chapter 7: Appendix

7.1 Source code for Recommendation System

```
import re

import math

import numpy as np

import pandas as pd

from creator.models import creator_Basic

import nltk




def normalize_document(text):

    wpt = nltk.WordPunctTokenizer()

    stop_words = nltk.corpus.stopwords.words("english")

    text = re.sub(r"[^a-zA-Z\s]", "", text.lower(), re.I | re.A)

    text = text.strip()

    tokens = wpt.tokenize(text)

    filtered_tokens = [token for token in tokens if token not in stop_words]

    text = " ".join(filtered_tokens)

    return text




def text_to_tfidf_vector(text, vocabulary):

    words = normalize_document(text).split()

    vector = np.zeros(len(vocabulary))

    for word in words:

        if word in vocabulary:

            vector[vocabulary[word]] += 1
    return vector



def cosine_similarity(vector_a, vector_b):
    def calculate_norm(vector):
        return math.sqrt(sum(x**2 for x in vector))

    dot_product = np.dot(vector_a, vector_b)
    norm_of_vector_a = calculate_norm(vector_a)
    norm_of_vector_b = calculate_norm(vector_b)
    similarity = dot_product / (norm_of_vector_a * norm_of_vector_b)
    return similarity



def recommendation():
    df = pd.DataFrame(
        creator_Basic.objects.all()
        .order_by("-TargetLaunchDate")
        .values("id", "description")
    )
    df["description"] = df["description"].fillna("")
    vocabulary = {
        word: idx
        for         idx,         word         in         enumerate(set("
".join(df["description"]).split()))
    }
    tfidf_vectors = [
        text_to_tfidf_vector(description, vocabulary)
        for description in df["description"]
    ]
    n_descriptions = len(df)
    cosine_sim_matrix = np.zeros((n_descriptions, n_descriptions))
    for i in range(n_descriptions):
        for j in range(i, n_descriptions):
            similarity        =        cosine_similarity(tfidf_vectors[i],
tfidf_vectors[j])
            cosine_sim_matrix[i, j] = similarity
            cosine_sim_matrix[j, i] = similarity

    cosine_sim_df = pd.DataFrame(cosine_sim_matrix)
    mapping = pd.Series(df.index, index=df["id"])
    return cosine_sim_df, mapping, df

from utils import custom_recommendation_engine
def recommend(input_id, number_of_recommend=3):
    (
        similarity_matrix,
        mapping,
        projects,
    ) = custom_recommendation_engine.recommendation()
    project_index = mapping[input_id]
    similarity_score = list(enumerate(similarity_matrix[project_index]))
    similarity_score = sorted(
        similarity_score, key=lambda x: x[1], reverse=True
    )
    similarity_score = similarity_score[1 : number_of_recommend + 1]
    project_indices = [i[0] for i in similarity_score]
    return {
        "recommendation_list": projects["id"].iloc[project_indices],
        "similarity_score": similarity_score,
    }
```

7.2 Screenshots

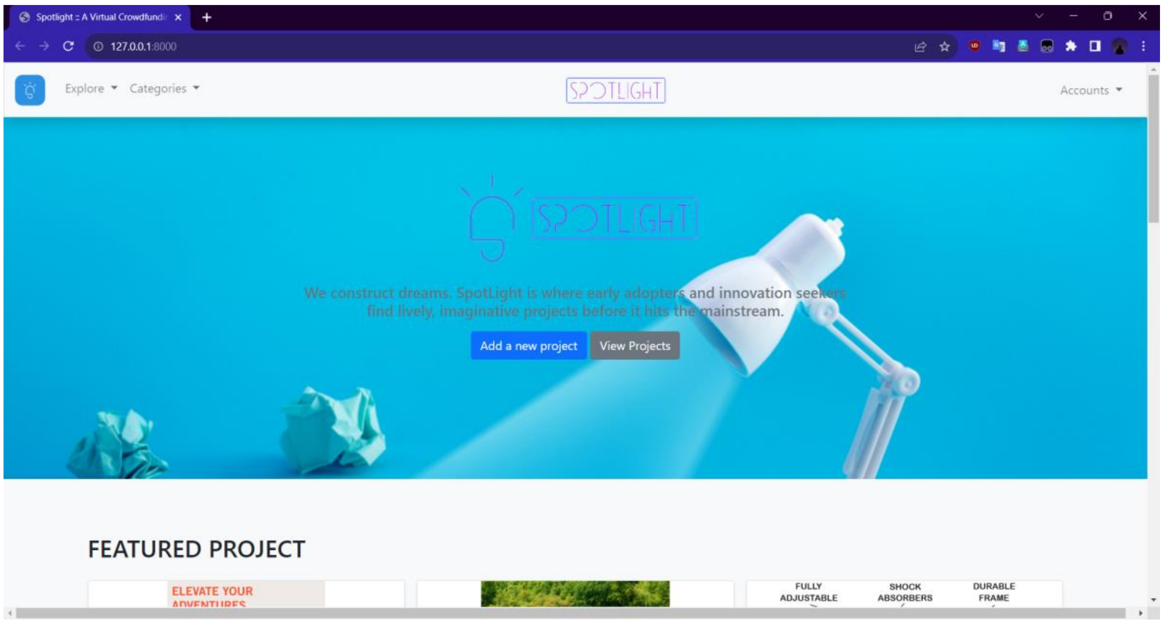

Spotlight :: A Virtual Crowdfundi
127.0.0.1:8000
Explore
Categories
SPOTLIGHT
Accounts
We construct dreams. SpotLight is where early adopters and innovation seekers find lively, imaginative projects before it hits the mainstream.
Add a new project
View Projects
FEATURED PROJECT
ELEVATE YOUR ADVENTURES
FULLY ADJUSTABLE
SHOCK ABSORBERS
DURABLE FRAME

Figure 7.1 - Landing Page

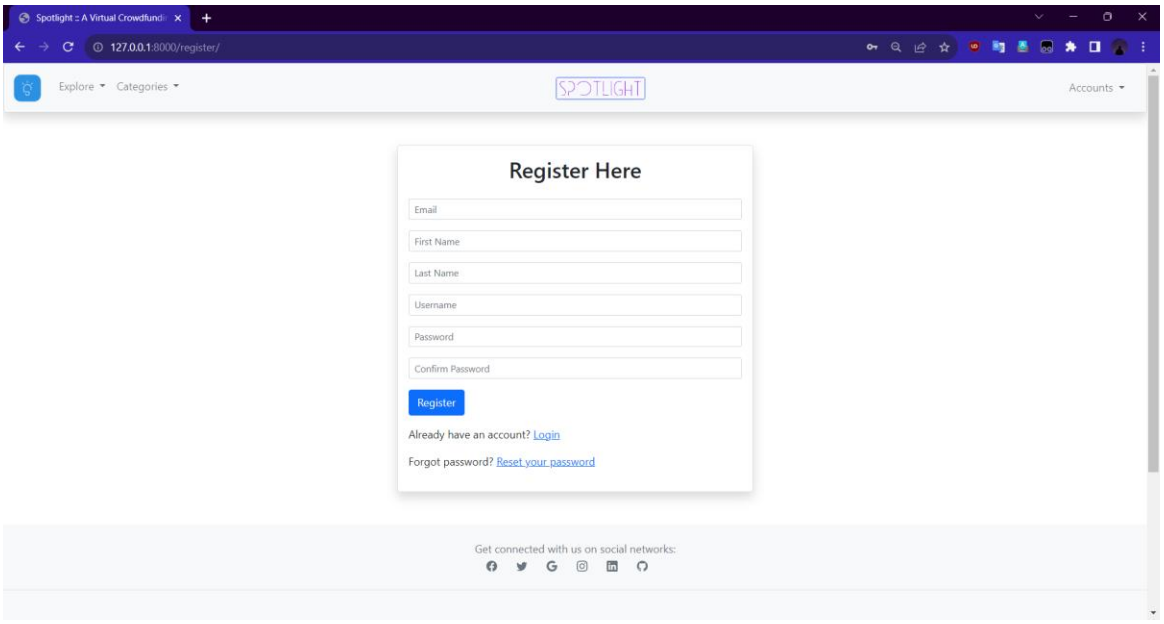
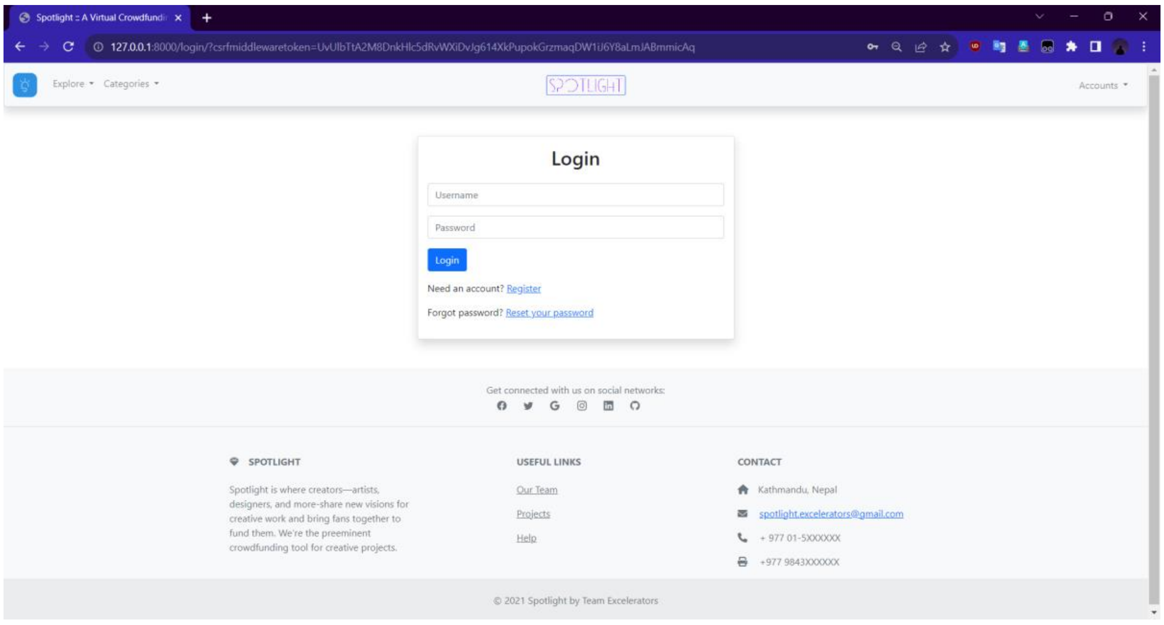

Spotlight :: A Virtual Crowdfundi
127.0.0.1:8000/register/
Explore
Categories
SPOTLIGHT
Accounts
Register Here
Email
First Name
Last Name
Username
Password
Confirm Password
Register
Already have an account? Login
Forgot password? Reset your password
Get connected with us on social networks:

Figure 7.2 - Register Page


Spotlight :: A Virtual Crowdfundi
127.0.0.1:8000/login/?csrfmiddlewaretoken=UvUlbTtA2M8DnkHlc5dRvWXiDvJg614XkPupokGrzmaqDW1iJ6Y8aLmJABmmicAq
Explore
Categories
SPOTLIGHT
Accounts
Login
Username
Password
Login

Need an account? Register

Forgot password? Reset your password

Get connected with us on social networks:

SPOTLIGHT

Spotlight is where creators—artists, designers, and more-share new visions for creative work and bring fans together to fund them. We're the preeminent crowdfunding tool for creative projects.

USEFUL LINKS

Our Team

Projects

Help

CONTACT

Kathmandu, Nepal

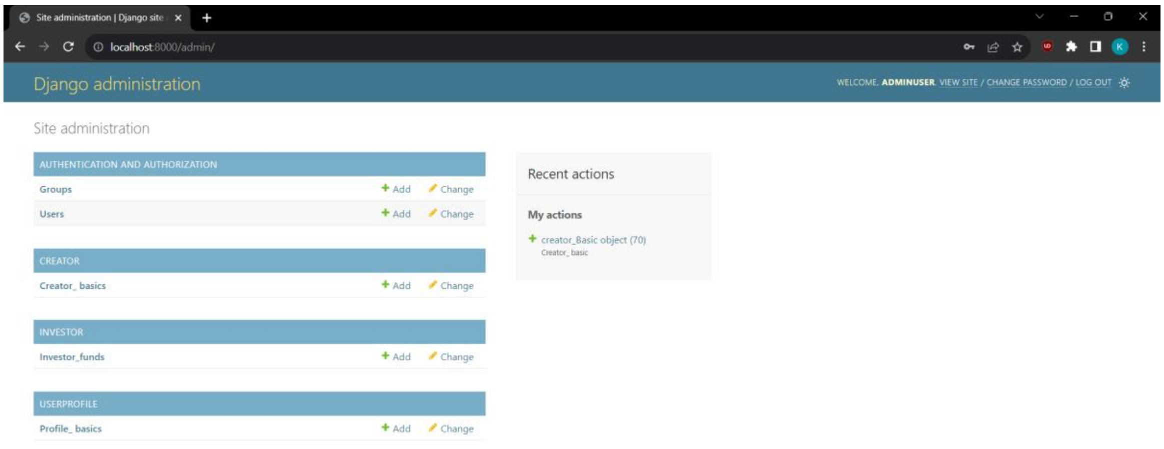
spotlight.excelerators@gmail.com

+ 977 01-5XXXXXX

+977 9843XXXXXX

© 2021 Spotlight by Team Excelerators

Figure 7.3 - Login Page


Site administration | Django site
localhost:8000/admin/
Django administration
WELCOME, ADMINUSER. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

CREATOR		
Creator_ basics	+ Add	Change

INVESTOR		
Investor_funds	+ Add	Change

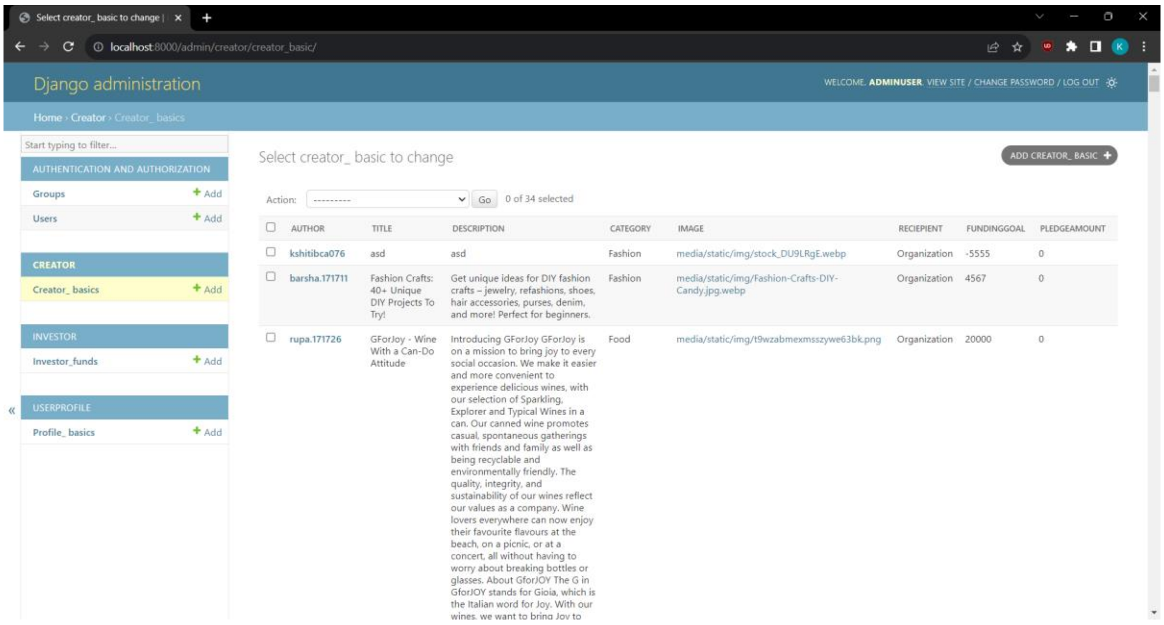
USERPROFILE		
Profile_ basics	+ Add	Change

Recent actions

My actions

+ creator_Basic object (70)
Creator_ basic

Figure 7.4 - Admin Page

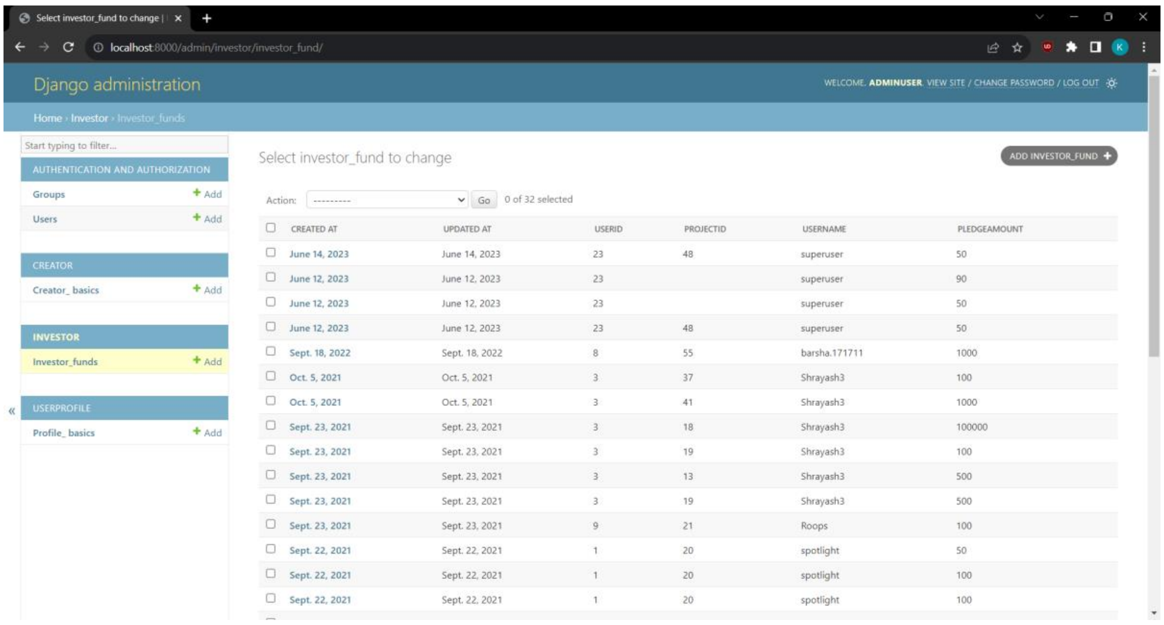

Select creator_ basic to change |
localhost:8000/admin/creator/creator_basic/
Django administration
WELCOME, ADMINUSER. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Creator › Creator_ basics
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
CREATOR
Creator_ basics + Add
INVESTOR
Investor_funds + Add
USERPROFILE
Profile_ basics + Add
ADD CREATOR_ BASIC

Select creator_ basic to change

Action: --------- Go 0 of 34 selected

AUTHOR	TITLE	DESCRIPTION	CATEGORY	IMAGE	RECIEPIENT	FUNDINGGOAL	PLEDGEAMOUNT
kshitibca076	asd	asd	Fashion	media/static/img/stock_DU9LRgE.webp	Organization	-5555	0
barsha.171711	Fashion Crafts: 40+ Unique DIY Projects To Try!	Get unique ideas for DIY fashion crafts – jewelry, refashions, shoes, hair accessories, purses, denim, and more! Perfect for beginners.	Fashion	media/static/img/Fashion-Crafts-DIY-Candy.jpg.webp	Organization	4567	0
rupa.171726	GForJoy - Wine With a Can-Do Attitude	Introducing GForJoy GForJoy is on a mission to bring joy to every social occasion. We make it easier and more convenient to experience delicious wines, with our selection of Sparkling, Explorer and Typical Wines in a can. Our canned wine promotes casual, spontaneous gatherings with friends and family as well as being recyclable and environmentally friendly. The quality, integrity, and sustainability of our wines reflect our values as a company. Wine lovers everywhere can now enjoy their favourite flavours at the beach, on a picnic, or at a concert, all without having to worry about breaking bottles or glasses. About GforJOY The G in GforJOY stands for Gioia, which is the Italian word for Joy. With our wines, we want to bring Joy to	Food	media/static/img/t9wzabmexmsszywe63bk.png	Organization	20000	0

Figure 7.5 - Projects Page

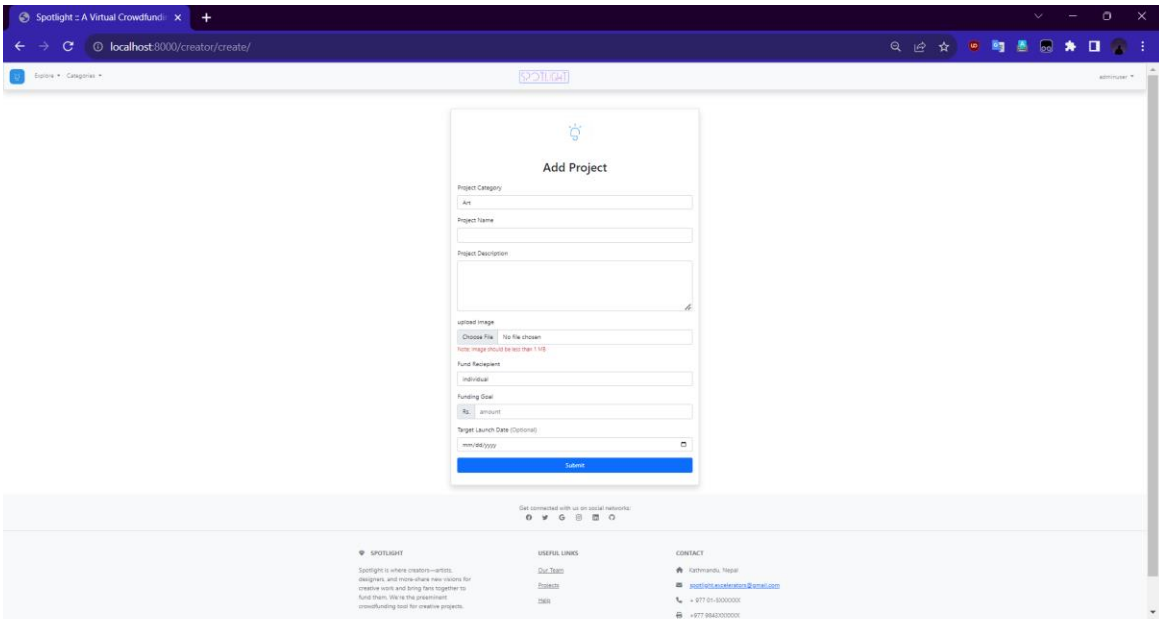

Select investor_fund to change |
localhost:8000/admin/investor/investor_fund/
Django administration
WELCOME, ADMINUSER. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Investor › Investor_funds
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
CREATOR
Creator_ basics + Add
INVESTOR
Investor_funds + Add
USERPROFILE
Profile_ basics + Add
ADD INVESTOR_FUND

Select investor_fund to change

Action: --------- Go 0 of 32 selected

CREATED AT	UPDATED AT	USERID	PROJECTID	USERNAME	PLEDGEAMOUNT
June 14, 2023	June 14, 2023	23	48	superuser	50
June 12, 2023	June 12, 2023	23		superuser	90
June 12, 2023	June 12, 2023	23		superuser	50
June 12, 2023	June 12, 2023	23	48	superuser	50
Sept. 18, 2022	Sept. 18, 2022	8	55	barsha.171711	1000
Oct. 5, 2021	Oct. 5, 2021	3	37	Shrayash3	100
Oct. 5, 2021	Oct. 5, 2021	3	41	Shrayash3	1000
Sept. 23, 2021	Sept. 23, 2021	3	18	Shrayash3	100000
Sept. 23, 2021	Sept. 23, 2021	3	19	Shrayash3	100
Sept. 23, 2021	Sept. 23, 2021	3	13	Shrayash3	500
Sept. 23, 2021	Sept. 23, 2021	3	19	Shrayash3	500
Sept. 23, 2021	Sept. 23, 2021	9	21	Roops	100
Sept. 22, 2021	Sept. 22, 2021	1	20	spotlight	50
Sept. 22, 2021	Sept. 22, 2021	1	20	spotlight	100
Sept. 22, 2021	Sept. 22, 2021	1	20	spotlight	100

Figure 7.6 - Funds page


Spotlight :: A Virtual Crowdfundi
localhost:8000/creator/create/
Explore
Categories
SPOTLIGHT
adminuser

Add Project

Project Category
Art

Project Name

Project Description

upload image
Choose File No file chosen
Note: image should be less than 1 MB

Fund Reciepient
Individual

Funding Goal
Rs. amount

Target Launch Date (Optional)
mm/dd/yyyy

Submit

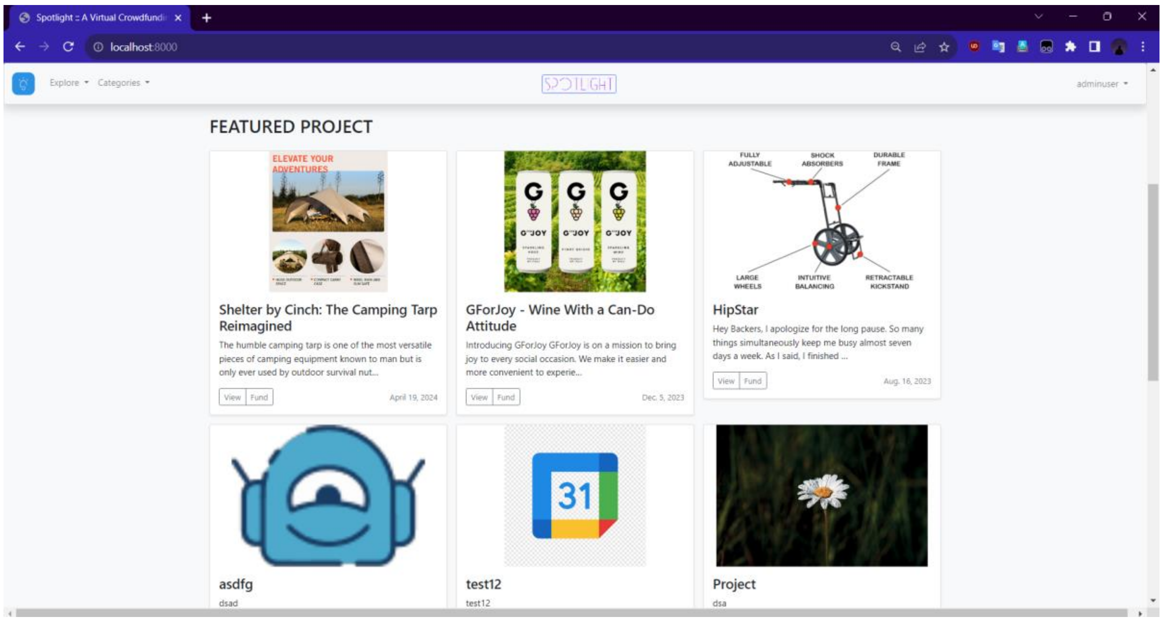
Get connected with us on social networks:

SPOTLIGHT
Spotlight is where creators—artists, designers, and more-share new visions for creative work and bring fans together to fund them. We're the preeminent crowdfunding tool for creative projects.

USEFUL LINKS
Our Team
Projects
Help

CONTACT
Kathmandu, Nepal
spotlight.excelerators@gmail.com
+ 977 01-5XXXXXX
+977 9843XXXXXX

Figure 7.7- Add Project page



FEATURED PROJECT

Shelter by Cinch: The Camping Tarp Reimagined
The humble camping tarp is one of the most versatile pieces of camping equipment known to man but is only ever used by outdoor survival nut...
View | Fund April 19, 2024

GForJoy - Wine With a Can-Do Attitude
Introducing GForJoy GForJoy is on a mission to bring joy to every social occasion. We make it easier and more convenient to experie...
View | Fund Dec. 5, 2023

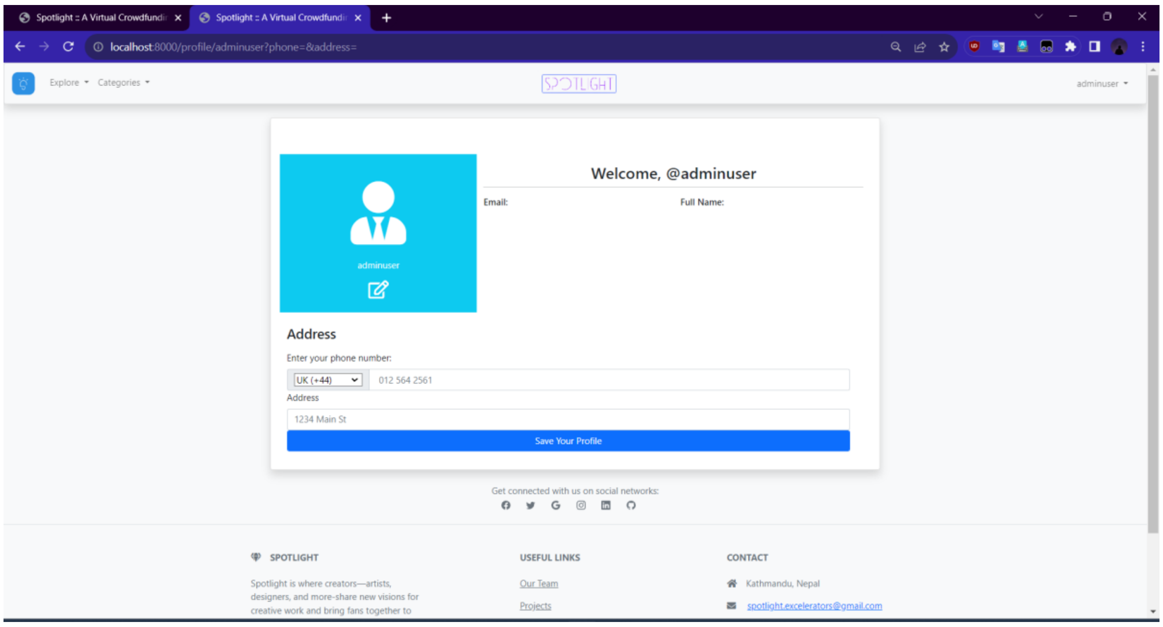
HipStar
Hey Backers, I apologize for the long pause. So many things simultaneously keep me busy almost seven days a week. As I said, I finished ...
View | Fund Aug. 16, 2023

asdfg
dsad

test12
test12

Project
dsa

Figure 7.8 - Project View page


Spotlight :: A Virtual Crowdfundi
localhost:8000/profile/adminuser?phone=&address=
Explore
Categories
SPOTLIGHT
adminuser

Welcome, @adminuser

Email:

Full Name:

adminuser

Address

Enter your phone number:

UK (+44) | 012 564 2561

Address

1234 Main St

Save Your Profile

Get connected with us on social networks:

SPOTLIGHT

Spotlight is where creators—artists, designers, and more-share new visions for creative work and bring fans together to

USEFUL LINKS

Our Team

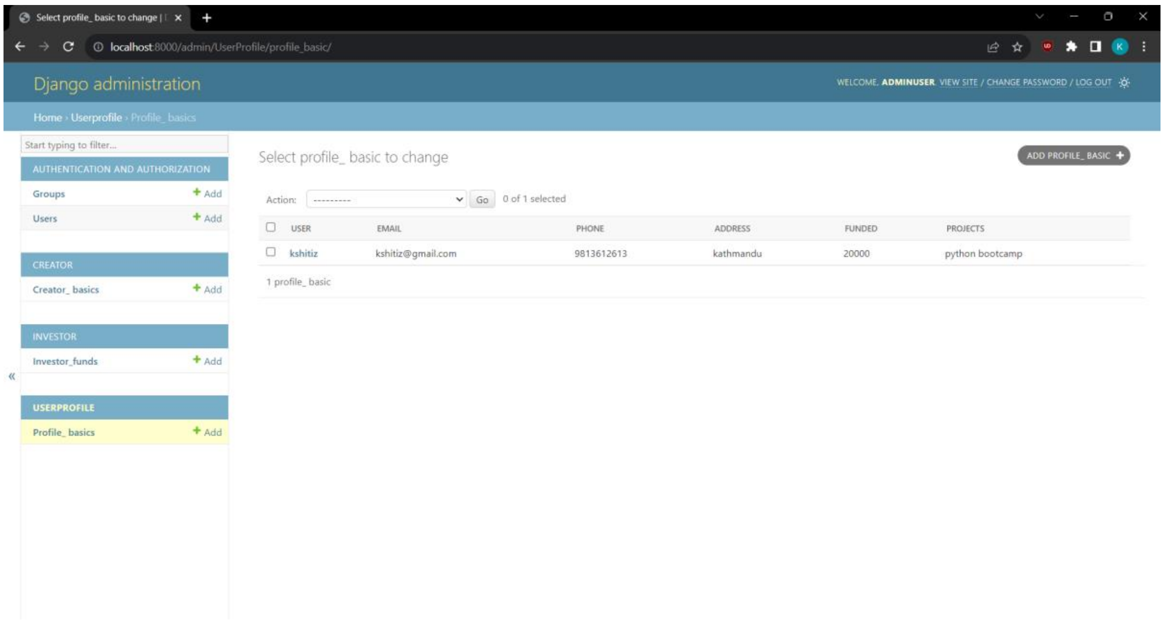
Projects

CONTACT

Kathmandu, Nepal

spotlight.excelerators@gmail.com

Figure 7.9 - Profile view page


Select profile_ basic to change | I
localhost:8000/admin/UserProfile/profile_basic/
Django administration
WELCOME, ADMINUSER. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Userprofile › Profile_ basics
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Users
Add
CREATOR
Creator_ basics
Add
INVESTOR
Investor_funds
Add
USERPROFILE
Profile_ basics
Add
ADD PROFILE_ BASIC

Select profile_ basic to change

Action: --------- Go 0 of 1 selected

USER	EMAIL	PHONE	ADDRESS	FUNDED	PROJECTS
kshitiz	kshitiz@gmail.com	9813612613	kathmandu	20000	python bootcamp

1 profile_ basic

Figure 7.10 - Funder profile page