[Comic-Collection]

ON

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering
(Information Technology)

By

Under the guidance of

Dipti Karani

Department of Information Technology

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY, Chembur, Mumbai 400074

(An Autonomous Institute, Affiliated to University of Mumbai)

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

- - - - - - - - - - -
(**Signature)**
Aditi Taksale 58

Abstract

Abstracts contain most of the following kinds of information in brief form. The body of your paper will, of course, develop and explain these ideas much more fully. As you will see in the samples below, the proportion of your abstract that you devote to each kind of information—and the sequence of that information—will vary, depending on the nature and genre of the paper that you are summarizing in your abstract. And in some cases, some of this information is implied, rather than stated explicitly. The Publication Manual of the American Psychological Association, which is widely used in the social sciences, gives specific guidelines for what to include in the abstract for different kinds of papers—for empirical studies, literature reviews or meta-analyses, theoretical papers, methodological papers, and case studies.

Keywords-*literature, theoretical, methodological, include, Publication*

Contents

1 Introduction

CHAPTER: 1 INTRODUCTION

[Comic-Collection]

ON

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering
(Information Technology)

By

Aditi Taksale 58

Under the guidance of

Dipti Karani

Department of Information Technology

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY, Chembur, Mumbai 400074

(An Autonomous Institute, Affiliated to University of Mumbai)

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

- - - - - - - - - - -
(**Signature)**
Aditi Taksale 58

Abstract

Abstracts contain most of the following kinds of information in brief form. The body of your paper will, of course, develop and explain these ideas much more fully. As you will see in the samples below, the proportion of your abstract that you devote to each kind of information—and the sequence of that information—will vary, depending on the nature and genre of the paper that you are summarizing in your abstract. And in some cases, some of this information is implied, rather than stated explicitly. The Publication Manual of the American Psychological Association, which is widely used in the social sciences, gives specific guidelines for what to include in the abstract for different kinds of papers—for empirical studies, literature reviews or meta-analyses, theoretical papers, methodological papers, and case studies.

Keywords-*literature, theoretical, methodological, include, Publication*

Contents

5 Introduction

CHAPTER: 1 INTRODUCTION

The Idea Submission Platform is a full-stack web application developed to provide users with an organized and interactive space to submit, explore, and engage with innovative ideas. This project brings together modern web technologies— Flask for a lightweight and efficient backend, MongoDB for scalable and flexible data management, and HTML, CSS, JavaScript (with Bootstrap and AJAX) for a responsive and intuitive frontend experience.

The platform enables users to:

- Register and log in securely to manage their submitted ideas.
- Submit new ideas with details such as title, description, category tags, and more.
- Comment on and upvote others' ideas, fostering interaction and feedback.
- Browse, filter, and explore submitted ideas based on categories and popularity.
- Maintain a personalized user profile for managing all their idea-related activities.

The application follows a modular architecture, ensuring easy scalability and code maintainability. Flask APIs manage all essential CRUD operations, while MongoDB offers schema-less data storage to accommodate diverse idea entries. The user interface is designed to be clean, dynamic, and mobile-friendly, ensuring a consistent experience across devices.

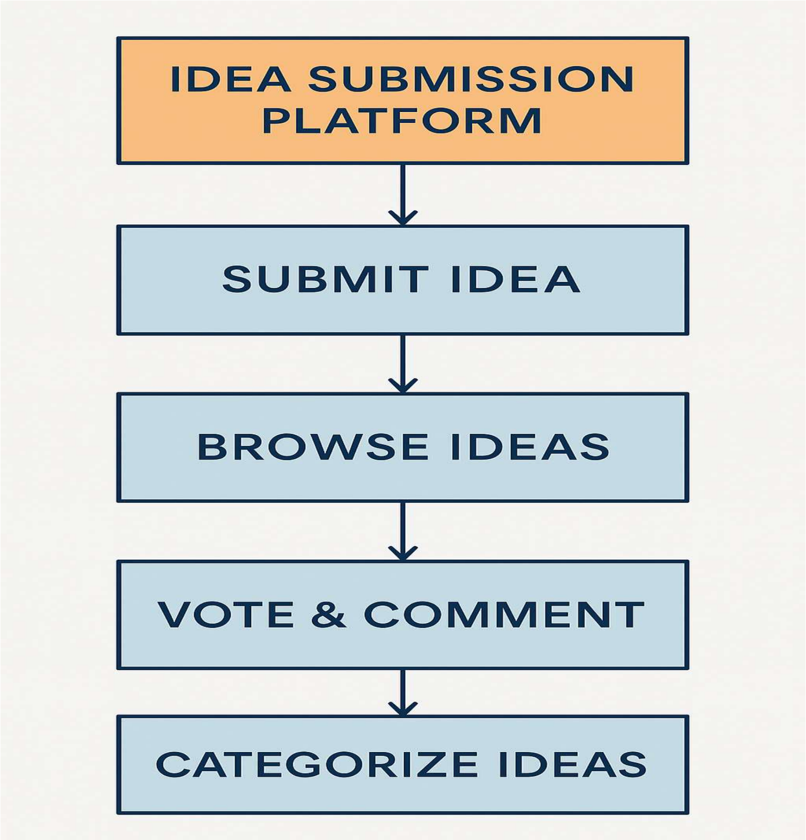
This project not only highlights seamless integration between backend and frontend technologies but also demonstrates how full-stack development can be utilized to build impactful, community-driven platforms focused on collaboration and innovation.

Chapter 2

Design and Implementation

.

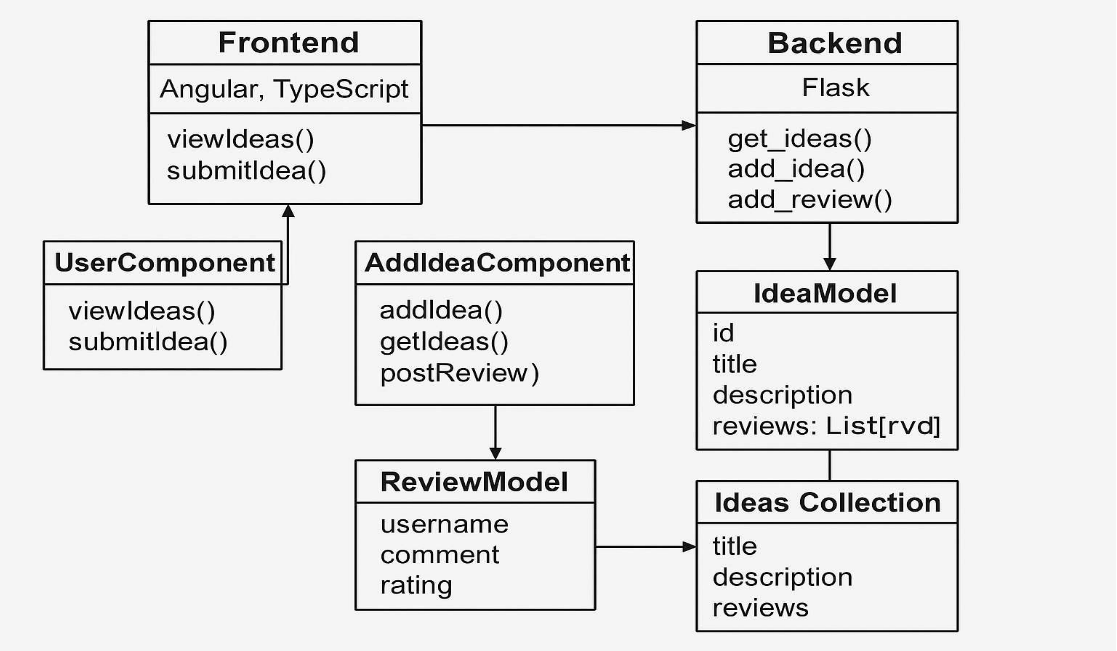
2.1 Block Diagram



This project follows a full-stack architecture using **Angular**, **Flask**, and **MongoDB**.

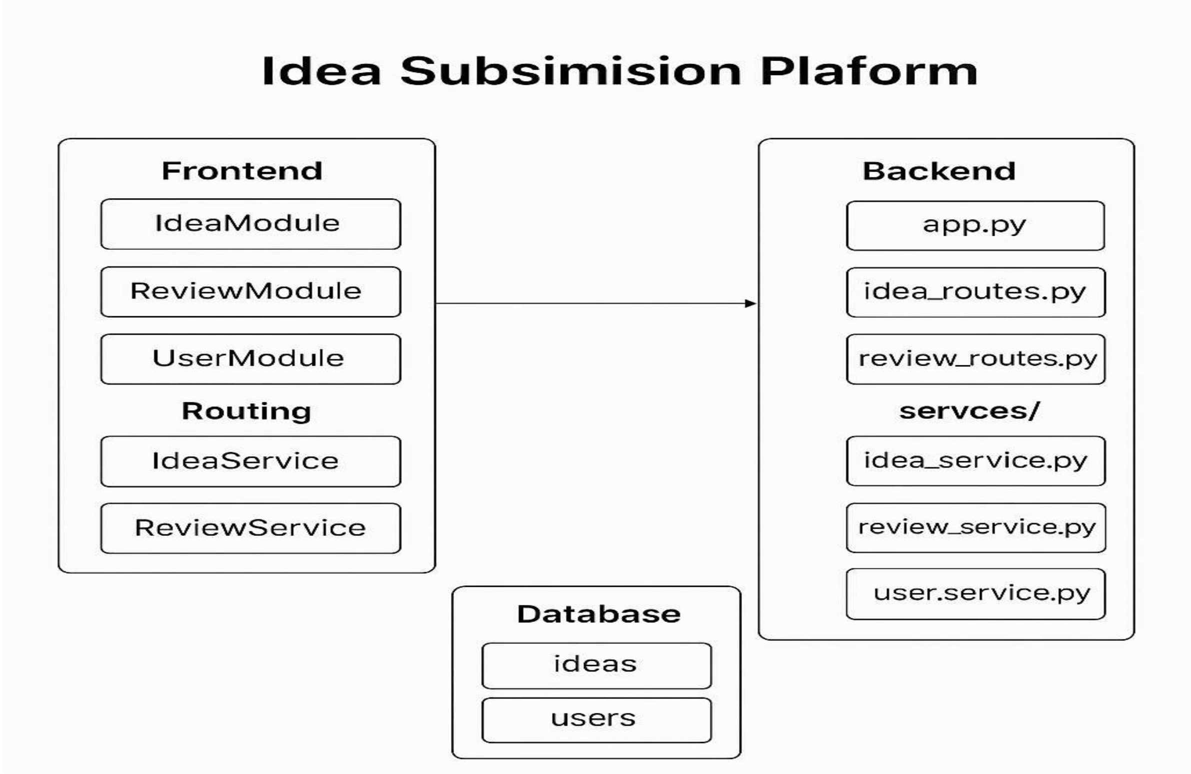
- **Angular** (Frontend): Handles user interactions like adding comics and posting reviews.
- **Flask** (Backend): Acts as an API server to manage data flow and perform **CRUD operations** (Create, Read, Update, Delete).
- **MongoDB** (Database): Stores comic details and user reviews in a flexible NoSQL format.

2.2 UML Diagram



1. Frontend (HTML/CSS/JavaScript + AJAX) • **UserComponent** • viewIdeas() – Display all submitted ideas • submitIdea() – Submit a new idea with details • **ProfileComponent** • viewUserIdeas() – Show ideas submitted by the • **IdeaService (AJAX-based)** • getIdeas() – Fetch all ideas from the backend • postIdea() – Send a new idea to the backend • postComment() – Submit a comment on an idea • voteIdea() – Upvote an idea ▪	**2. Backend (Flask)** • **IdeaModel** • id • title • description • category • author (username/email) • comments: List[CommentModel] • votes: int • **CommentModel** • username • text • timestamp ▪
3. Database (MongoDB) • **Ideas Collection** • title • description • category • author • votes • comments: [{ username, text, timestamp }] ○	

2.3 URL Diagram



- Frontend (Angular + TypeScript)
  - Components let users view, add comics, and submit reviews.
  - ComicService and ReviewService handle API calls.
- Backend (Flask)
  - comic_routes.py and review_routes.py define endpoints.
  - comic_service.py and review_service.py handle DB logic.
  - MongoDB stores comics and their reviews.
- Database (MongoDB)
  - One collection: comics, each with a list of embedded reviews.
- Flow:
  Angular UI → Flask API → MongoDB → Flask Response → Angular UI

2.4 Hardware Requirements

1. CPU: Quad-core processor or higher
2. RAM: 16 GB or higher Storage: SSD with at least 500 GB
3. Network: High-speed internet connection
4. User Devices: Any modern computer

2.5 Software Requirements

Languages: Angular, TypeScript, Flask, Mongodb

1. Frontend Development

Frameworks and Libraries:

- **Angular** – for building dynamic and component-based user interfaces
- **TypeScript** – for writing structured and scalable frontend logic
- **CSS** – for custom styling and layout design
- **Flask** – used as a lightweight backend API server (also interacts with frontend in some parts)

Tools:

- **Visual Studio Code** – code editor for development
- **Node.js** – for managing Angular dependencies and build processes

---

2. Backend Development

Database Management System:

- **MongoDB** – a NoSQL database used to store comic data and user reviews in a flexible document-based format
- **Backend Framework**:
- **Flask** – handles RESTful API creation, manages data flow, and performs CRUD operations with MongoDB

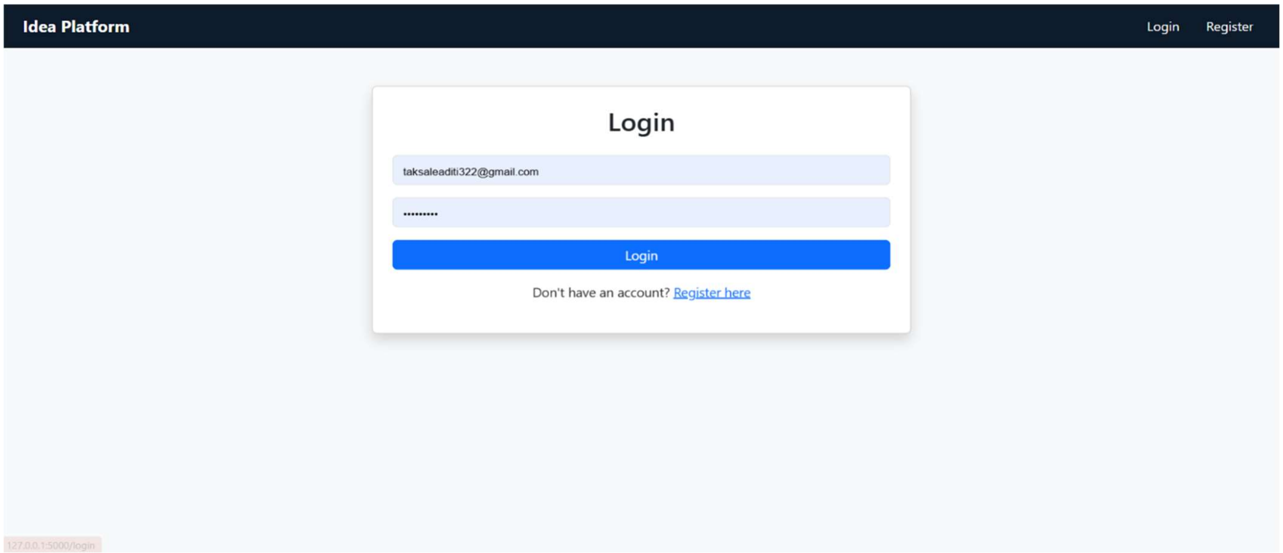
3. Development and Hosting-

Hosting Platform:- Netlify

Chapter 3: Results

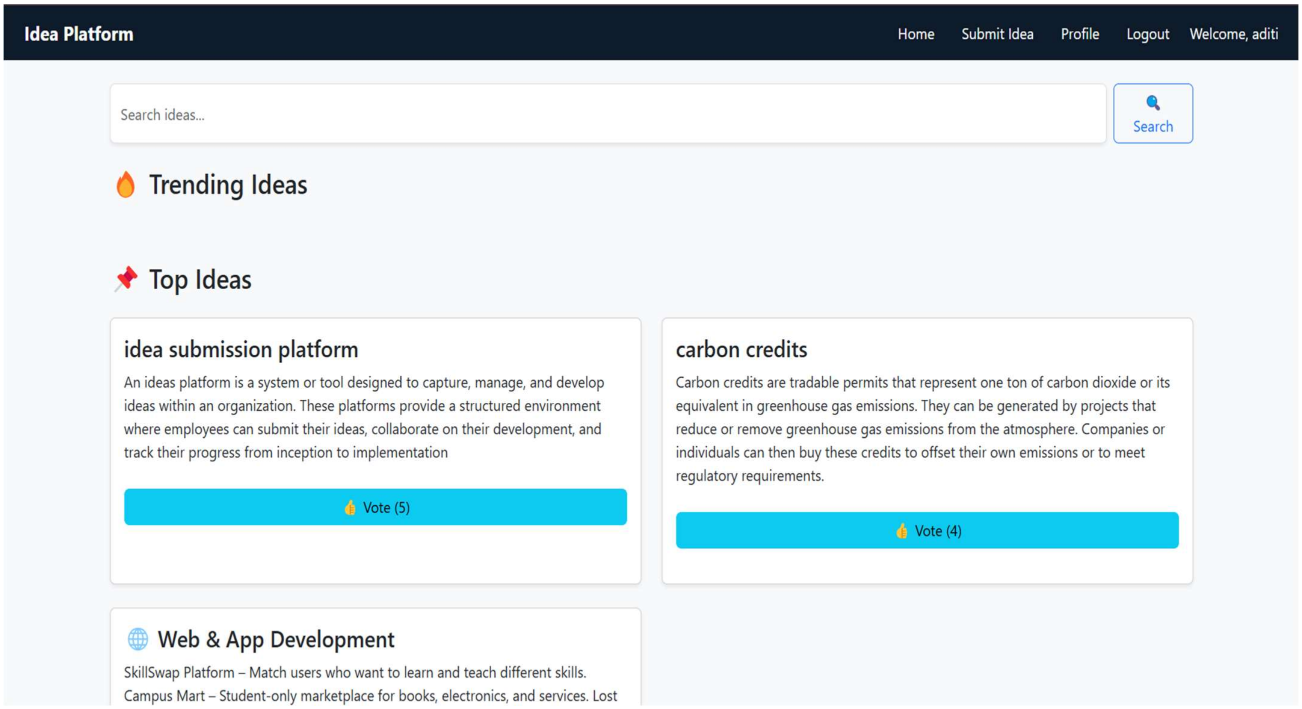
3.1 Results of Implementation:

LoginPage:



This is the login page where user can login using Google or email address

HOME Page:



idea submission platform

An ideas platform is a system or tool designed to capture, manage, and develop ideas within an organization. These platforms provide a structured environment where employees can submit their ideas, collaborate on their development, and track their progress from inception to implementation

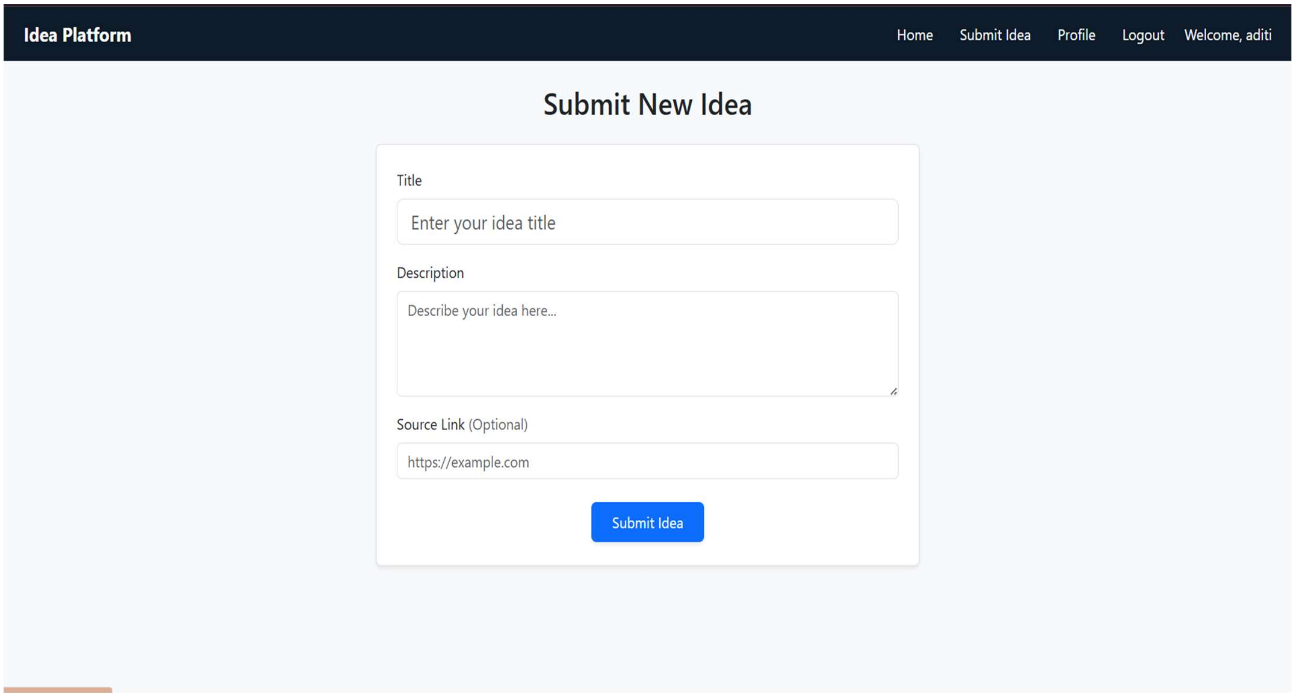
carbon credits

Carbon credits are tradable permits that represent one ton of carbon dioxide or its equivalent in greenhouse gas emissions. They can be generated by projects that reduce or remove greenhouse gas emissions from the atmosphere. Companies or individuals can then buy these credits to offset their own emissions or to meet regulatory requirements.

🌐 Web & App Development

SkillSwap Platform – Match users who want to learn and teach different skills. Campus Mart – Student-only marketplace for books, electronics, and services. Lost

TO SUBMIT IDEA



Submit New Idea

Title

Enter your idea title

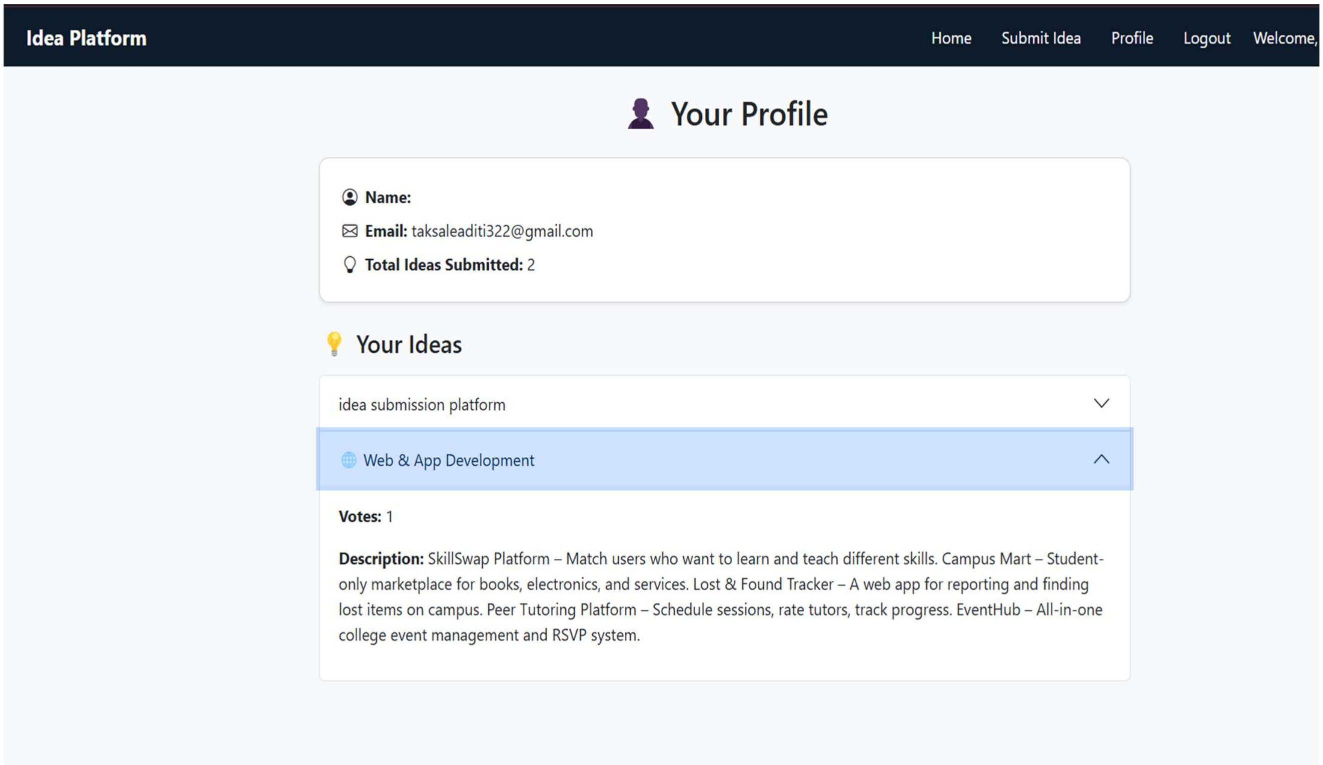
Description

Describe your idea here...

Source Link (Optional)

https://example.com

Profile page



Your Profile

Name:

Email: taksaleaditi322@gmail.com

Total Ideas Submitted: 2

Your Ideas

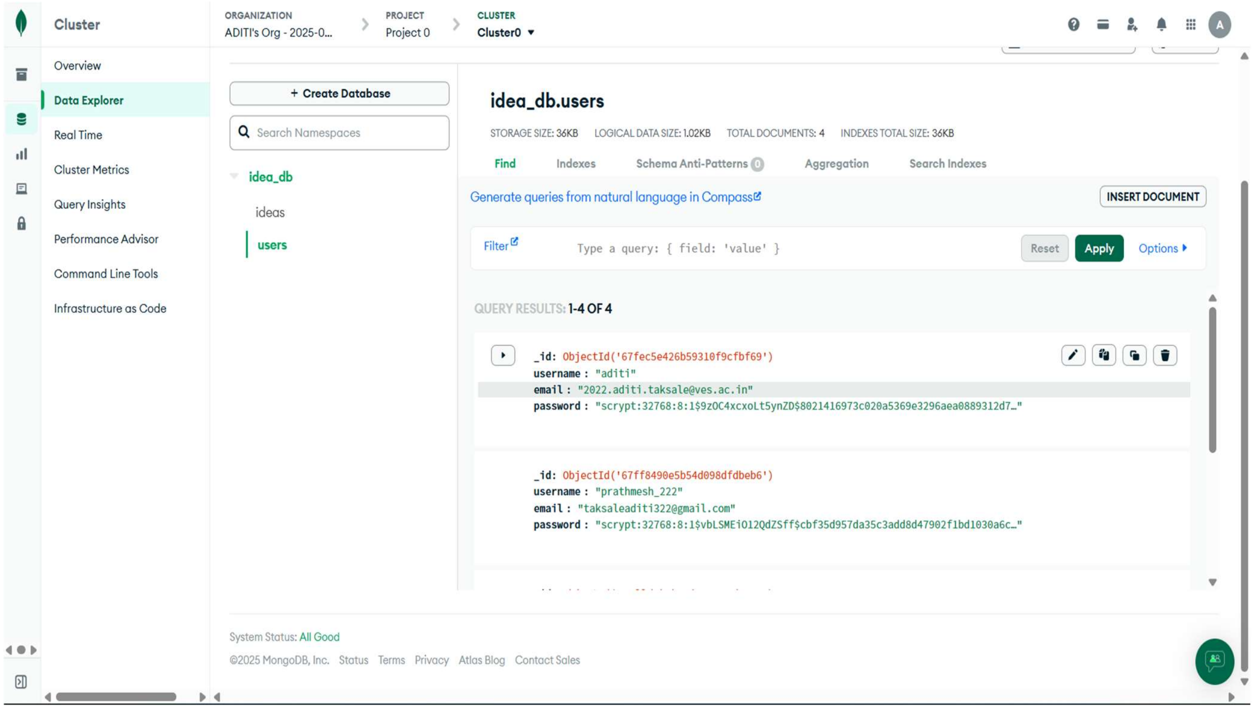
idea submission platform

Web & App Development

Votes: 1

Description: SkillSwap Platform – Match users who want to learn and teach different skills. Campus Mart – Student-only marketplace for books, electronics, and services. Lost & Found Tracker – A web app for reporting and finding lost items on campus. Peer Tutoring Platform – Schedule sessions, rate tutors, track progress. EventHub – All-in-one college event management and RSVP system.

In Backend(Using Mongodb):



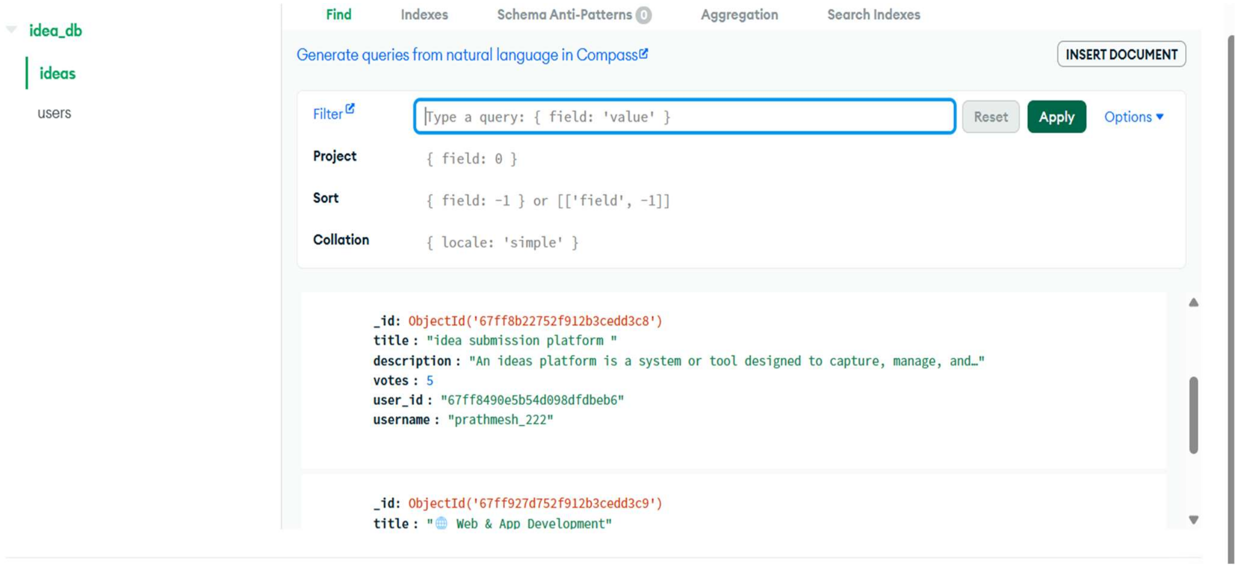
idea_db.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 1.02KB TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-4 OF 4

```
_id: ObjectId('67fec5e426b59310f9cfbf69')
username : "aditi"
email : "2022.aditi.taksale@ves.ac.in"
password : "scrypt:32768:8:1$9zOC4xcxoLt5ynZD$8021416973c020a5369e3296aea0889312d7…"

_id: ObjectId('67ff8490e5b54d098dfdbeb6')
username : "prathmesh_222"
email : "taksaleaditi322@gmail.com"
password : "scrypt:32768:8:1$vbLSMEiO12QdZSff$cbf35d957da35c3add8d47902f1bd1030a6c…"
```



```
_id: ObjectId('67ff8b22752f912b3cedd3c8')
title : "idea submission platform "
description : "An ideas platform is a system or tool designed to capture, manage, and…"
votes : 5
user_id : "67ff8490e5b54d098dfdbeb6"
username : "prathmesh_222"

_id: ObjectId('67ff927d752f912b3cedd3c9')
title : "🌐 Web & App Development"
```

Chapter 4 Conclusion

The Idea Submission Platform effectively integrates modern web technologies to provide an engaging and functional space for users to share, explore, and interact with innovative ideas. By utilizing Flask for backend development, MongoDB for flexible and scalable data storage, and a responsive frontend built with HTML, CSS, JavaScript, and AJAX, the platform offers a smooth and intuitive user experience.

This project showcases the capabilities of full-stack development by combining modular backend logic, dynamic client-side rendering, and efficient API communication. It emphasizes the importance of interactivity, collaboration, and accessibility in platforms designed to foster creativity and innovation. Additionally, it highlights the value of clean UI design, seamless data handling, and scalability in building modern web-based applications.

4.1 Reference:

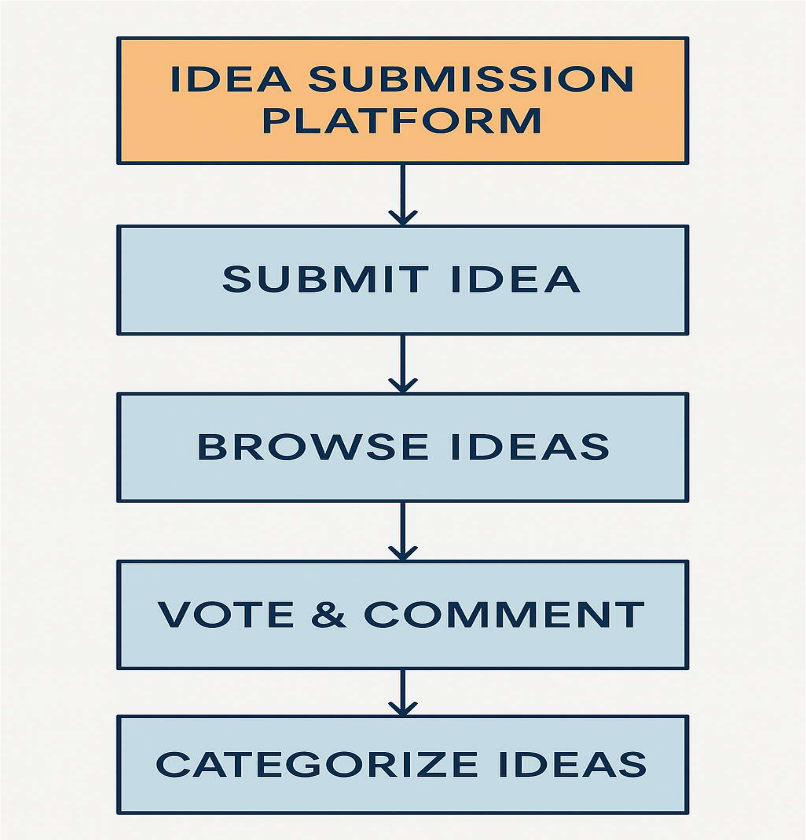
[1] Angular: https://youtu.be/0LhBvp8qpro?si=NBads_TQ6T_wyoew
[2] Flask: https://youtu.be/oA8brF3w5XQ?si=sx1v6m9ZdxEIumzK
[3] Deploy: https://youtu.be/9srnyNC1e_o?si=2aRIo9OPAfKJyVpw
[4] Mongodb: https://youtu.be/J6mDkcqU_ZE?si=8v9Oka3fFse4UUUU

Chapter 2

Design and Implementation

.

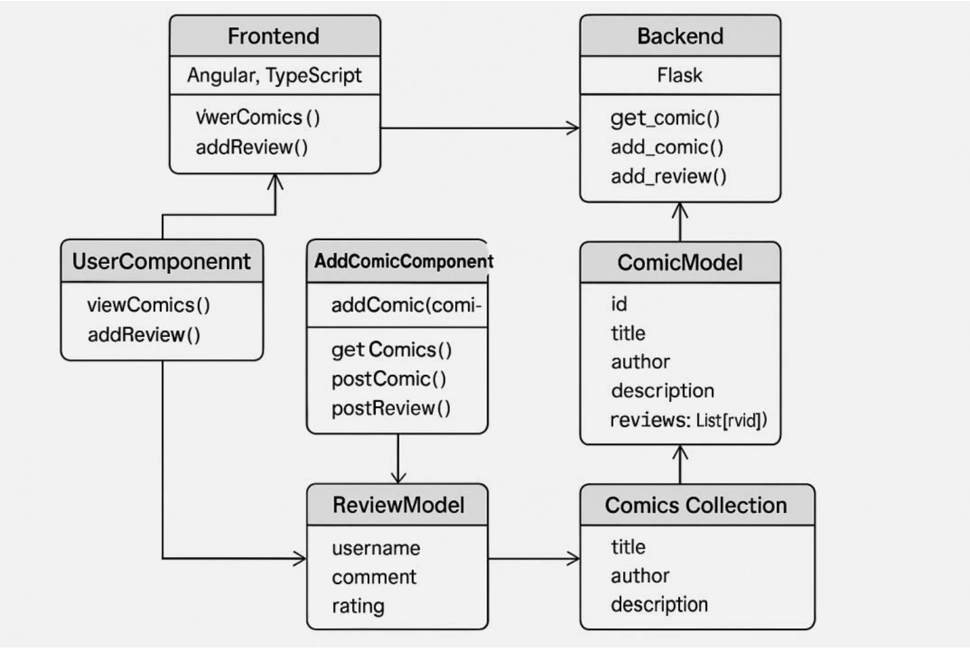
2.6 Block Diagram



This project follows a full-stack architecture using **Angular**, **Flask**, and **MongoDB**.

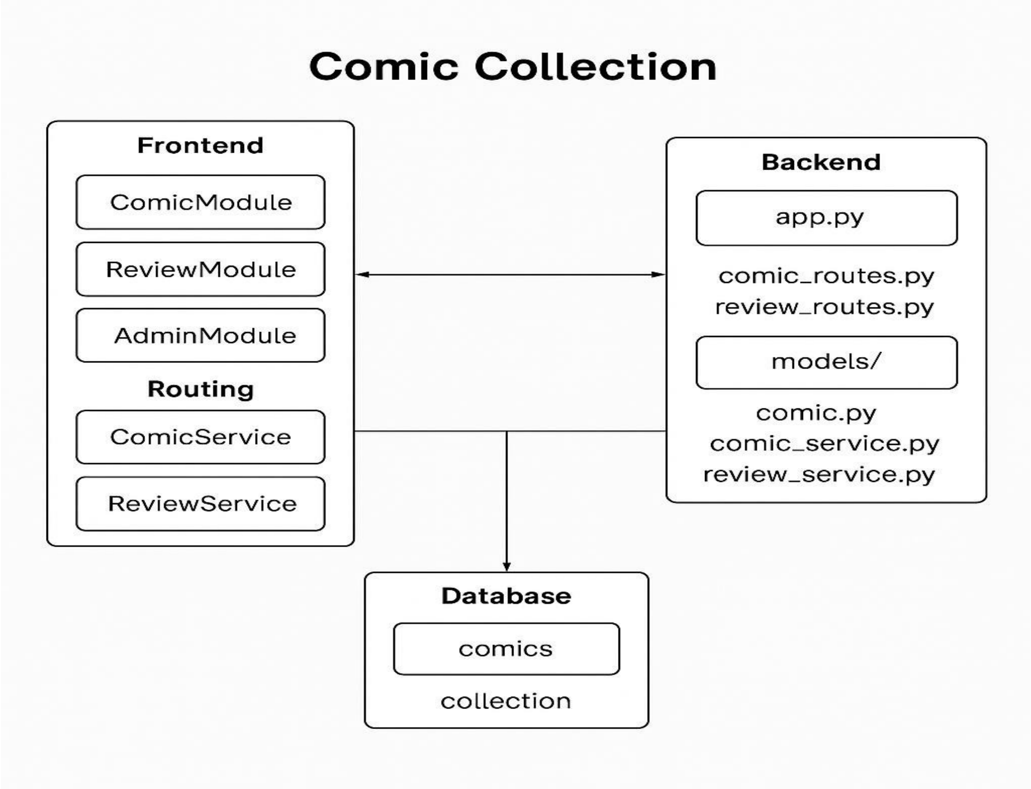
- **Angular** (Frontend): Handles user interactions like adding comics and posting reviews.
- **Flask** (Backend): Acts as an API server to manage data flow and perform **CRUD operations** (Create, Read, Update, Delete).
- **MongoDB** (Database): Stores comic details and user reviews in a flexible NoSQL format.

2.7 UML Diagram



2. Frontend (Angular - TypeScript) • **UserComponent** ○ ▪ viewComics() ○ ▪ addReview() • **AddComicComponent** ○ ▪ addComic(comicData) • **ComicService** ○ ▪ getComics() ○ ▪ postComic() ○ ▪ postReview()	**3. Backend (Flask)** • **ComicModel** ○ ▪ id ○ ▪ title ○ ▪ author ○ ▪ description ○ ▪ reviews: List[ReviewModel] • **ReviewModel** ○ ▪ username ○ ▪ comment ○ ▪ rating
4. Database (MongoDB) • **Comics Collection** ○ title ○ author ○ description ○ reviews [{ username, comment, rating }]	

2.8 URL Diagram



- Frontend (Angular + TypeScript)
    - Components let users view, add comics, and submit reviews.
    - ComicService and ReviewService handle API calls.
- Backend (Flask)
    - comic_routes.py and review_routes.py define endpoints.
    - comic_service.py and review_service.py handle DB logic.
    - MongoDB stores comics and their reviews.
- Database (MongoDB)
    - One collection: comics, each with a list of embedded reviews.
- Flow:
  Angular UI → Flask API → MongoDB → Flask Response → Angular UI

2.9 Hardware Requirements

1. CPU: Quad-core processor or higher
2. RAM: 16 GB or higher Storage: SSD with at least 500 GB
3. Network: High-speed internet connection
4. User Devices: Any modern computer

2.10 Software Requirements

Languages: Angular, TypeScript, Flask, Mongodb

1. Frontend Development
Frameworks and Libraries:

- **Angular** – for building dynamic and component-based user interfaces
- **TypeScript** – for writing structured and scalable frontend logic
- **CSS** – for custom styling and layout design
- **Flask** – used as a lightweight backend API server (also interacts with frontend in some parts)

Tools:

- **Visual Studio Code** – code editor for development
- **Node.js** – for managing Angular dependencies and build processes

---

2. Backend Development
Database Management System:

- **MongoDB** – a NoSQL database used to store comic data and user reviews in a flexible document-based format
- **Backend Framework**:
- **Flask** – handles RESTful API creation, manages data flow, and performs CRUD operations with MongoDB

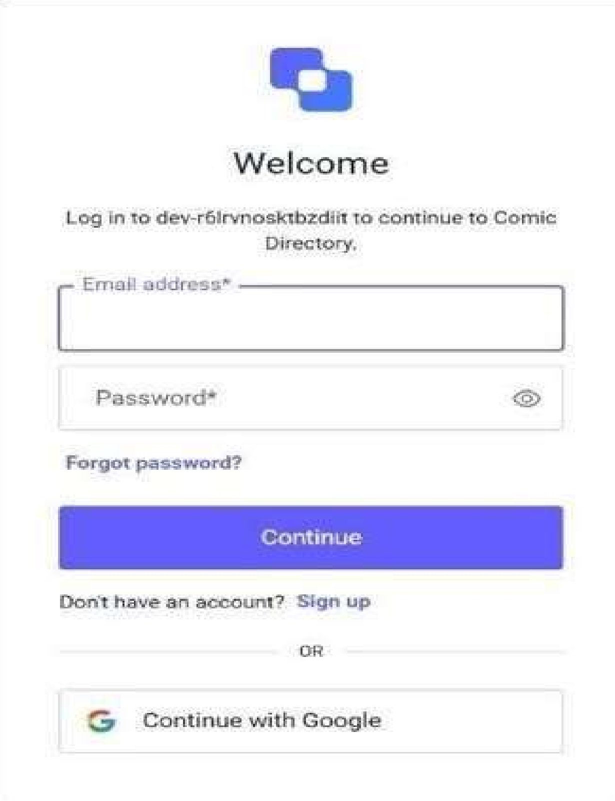
3. Development and Hosting-

Hosting Platform:- Netlify

Chapter 3: Results

3.1 Results of Implementation:

LoginPage:



Welcome

Log in to dev-r6lrvnosktbzdiit to continue to Comic Directory.

Email address*

Password*

Forgot password?

Continue

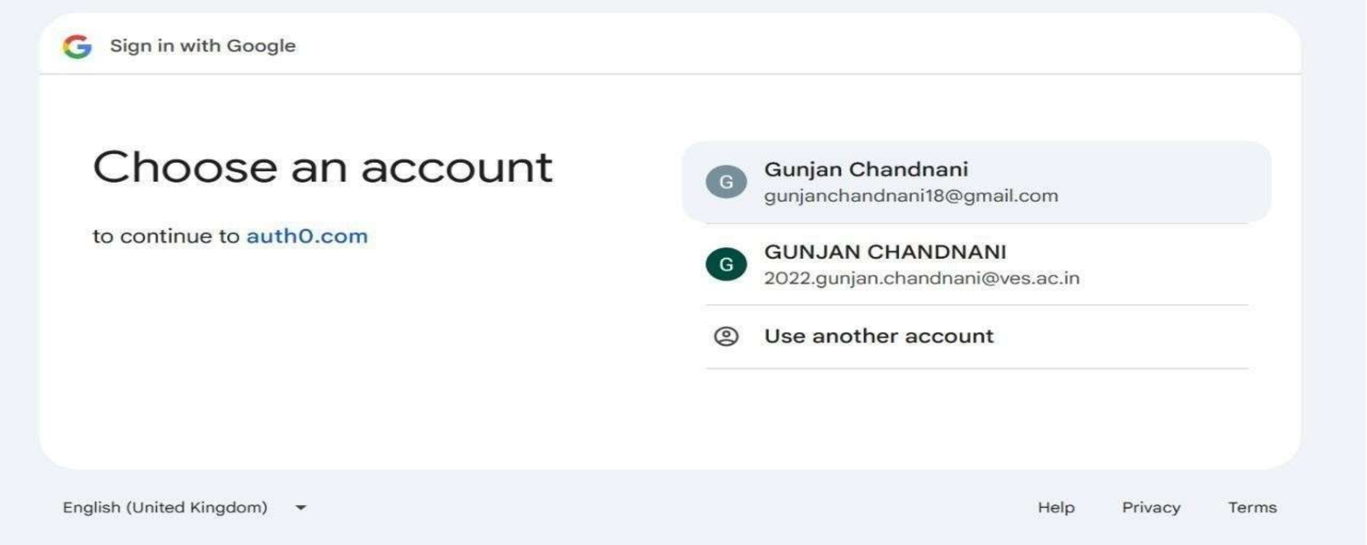
Don't have an account? Sign up

OR

Continue with Google

This is the login page where user can login using Google or email address

When User Continue using Google:



Sign in with Google

Choose an account

to continue to auth0.com

Gunjan Chandnani
gunjanchandnani18@gmail.com

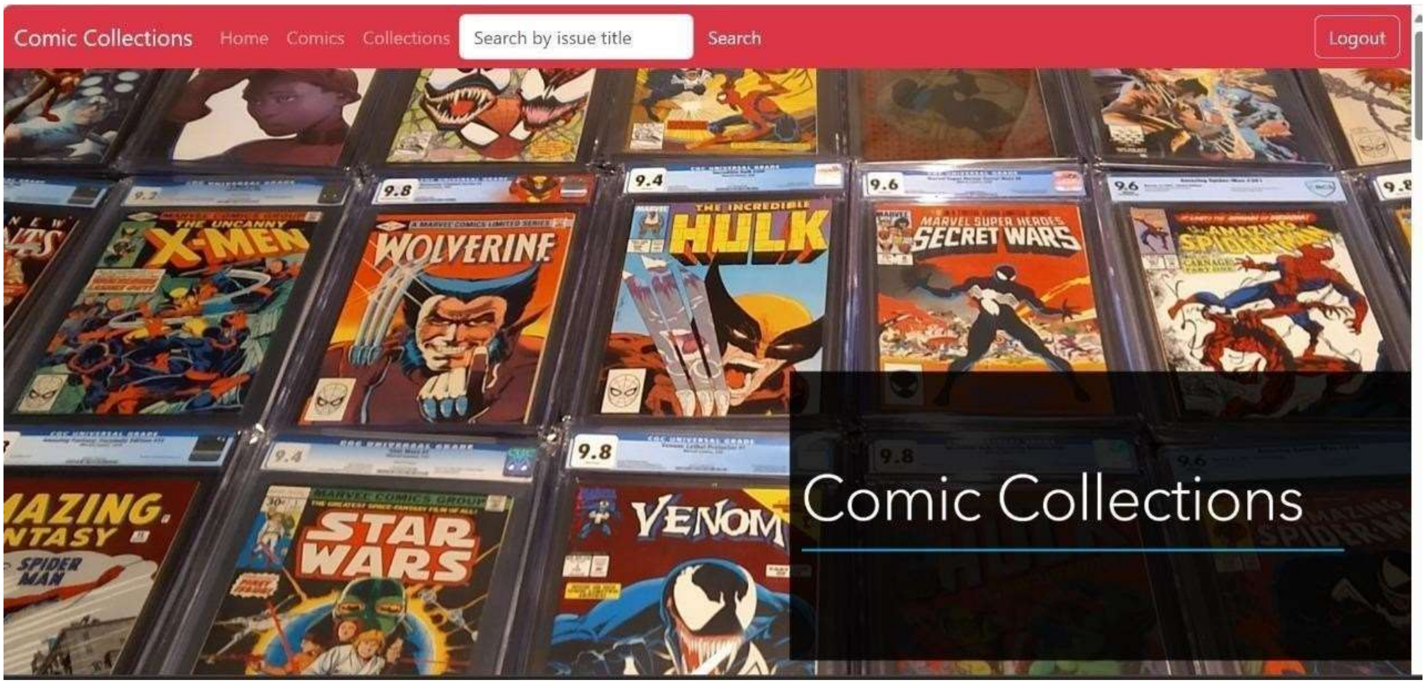
GUNJAN CHANDNANI
2022.gunjan.chandnani@ves.ac.in

Use another account

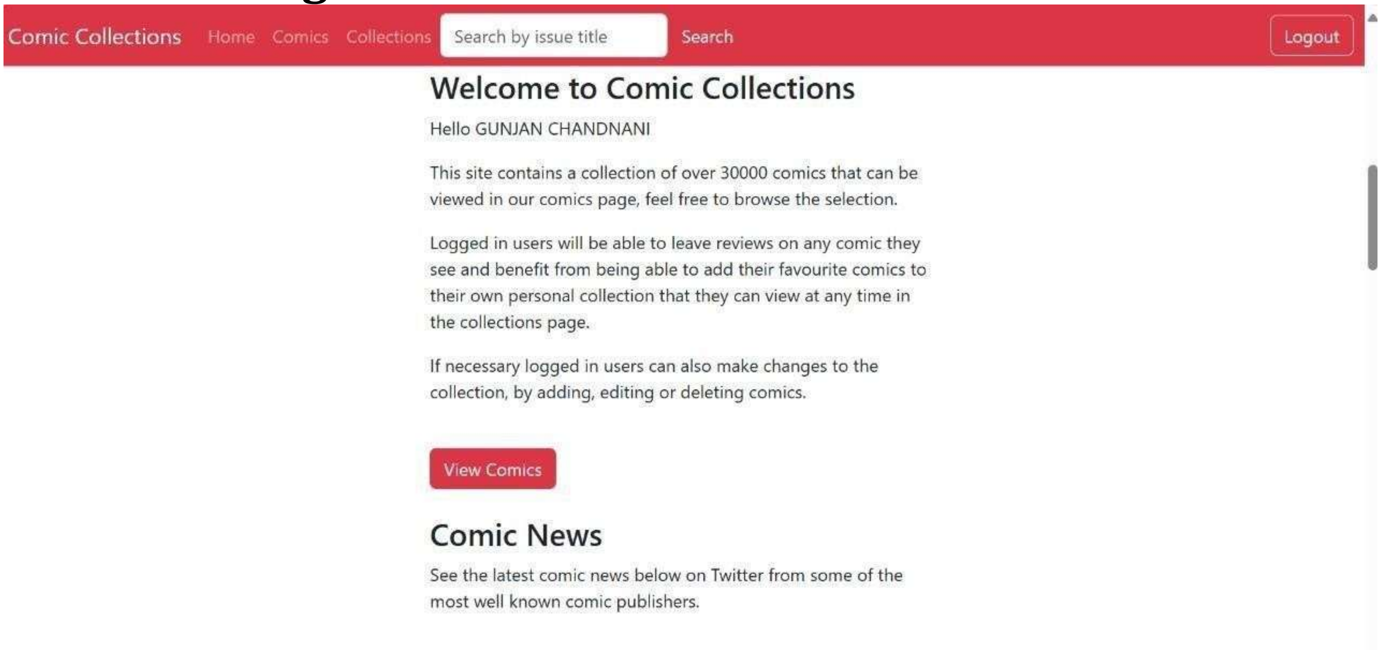
English (United Kingdom)

Help Privacy Terms

Home Page:



Collection Page:



Welcome to Comic Collections

Hello GUNJAN CHANDNANI

This site contains a collection of over 30000 comics that can be viewed in our comics page, feel free to browse the selection.

Logged in users will be able to leave reviews on any comic they see and benefit from being able to add their favourite comics to their own personal collection that they can view at any time in the collections page.

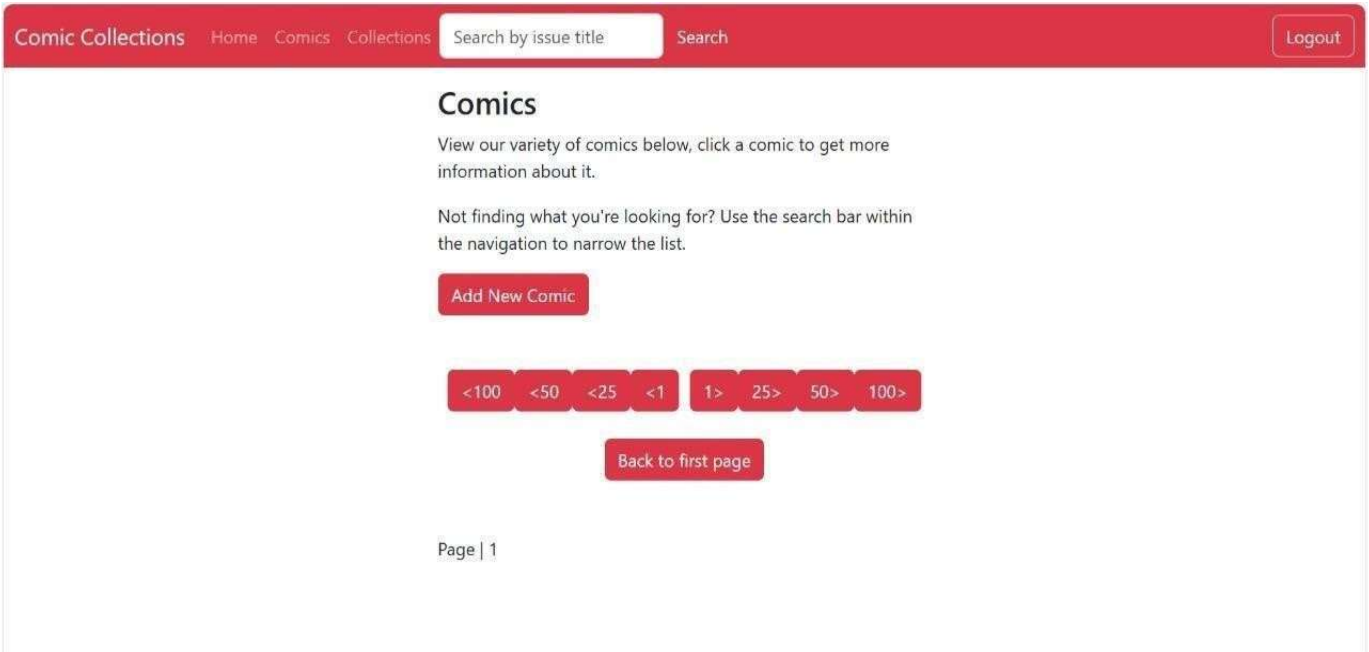
If necessary logged in users can also make changes to the collection, by adding, editing or deleting comics.

View Comics

Comic News

See the latest comic news below on Twitter from some of the most well known comic publishers.

Comic Page:



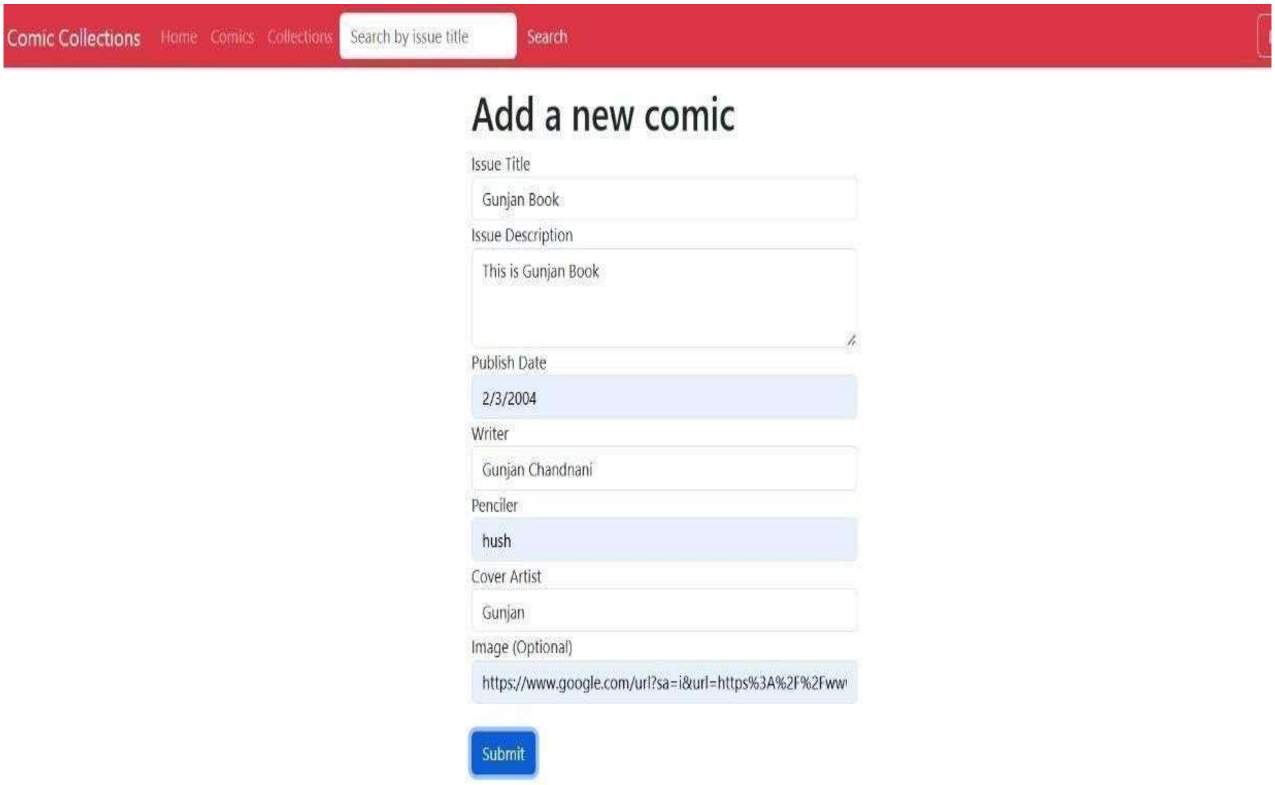
Comics

View our variety of comics below, click a comic to get more information about it.

Not finding what you're looking for? Use the search bar within the navigation to narrow the list.

Page | 1

When User Click on Add New Comic:



Add a new comic

Issue Title: Gunjan Book

Issue Description: This is Gunjan Book

Publish Date: 2/3/2004

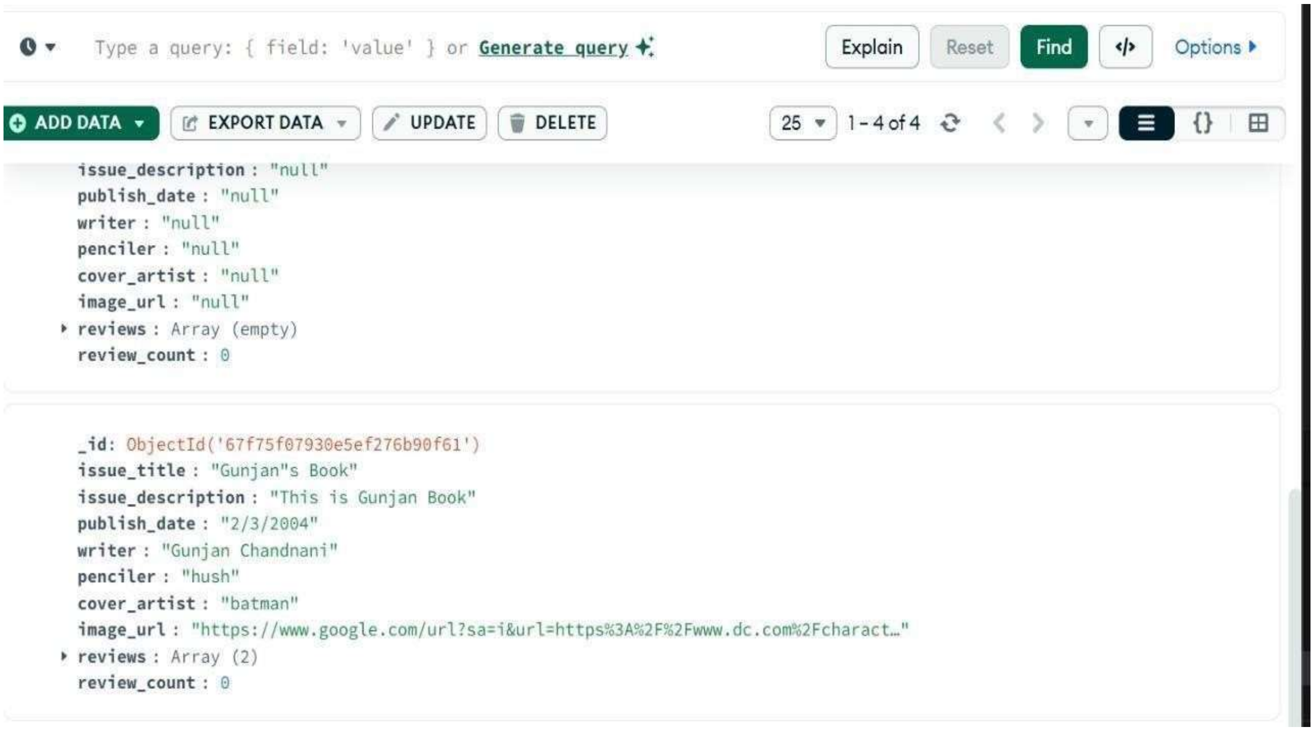
Writer: Gunjan Chandnani

Penciler: hush

Cover Artist: Gunjan

Image (Optional): https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww

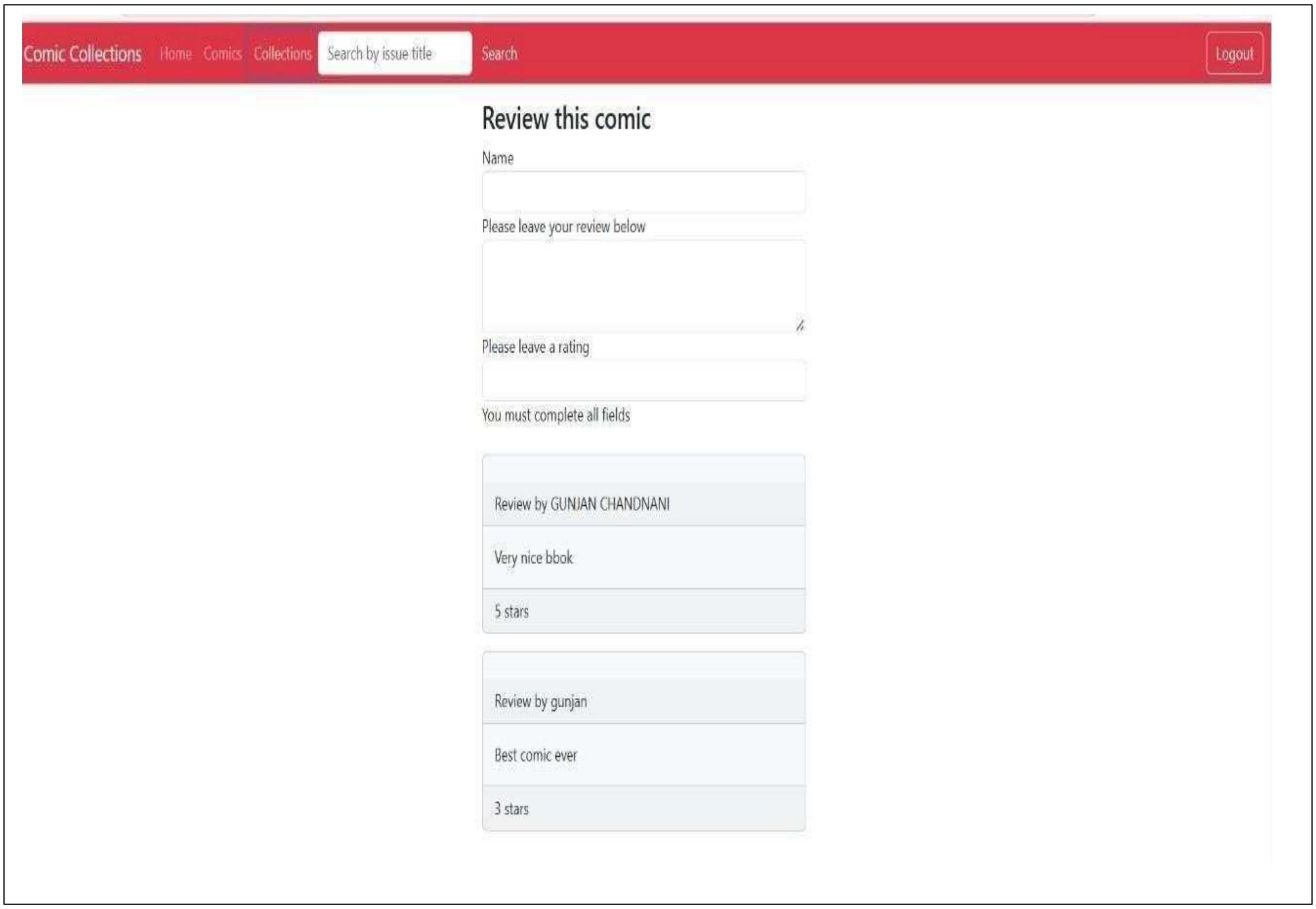
In Backend(Using Mongodb):



```
    issue_description : "null"
    publish_date : "null"
    writer : "null"
    penciler : "null"
    cover_artist : "null"
    image_url : "null"
  ▸ reviews : Array (empty)
    review_count : 0

    _id: ObjectId('67f75f07930e5ef276b90f61')
    issue_title : "Gunjan"s Book"
    issue_description : "This is Gunjan Book"
    publish_date : "2/3/2004"
    writer : "Gunjan Chandnani"
    penciler : "hush"
    cover_artist : "batman"
    image_url : "https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.dc.com%2Fcharact…"
  ▸ reviews : Array (2)
    review_count : 0
```

Review Page:



Review this comic

Name

Please leave your review below

Please leave a rating

You must complete all fields

Review by GUNJAN CHANDNANI

Very nice bbok

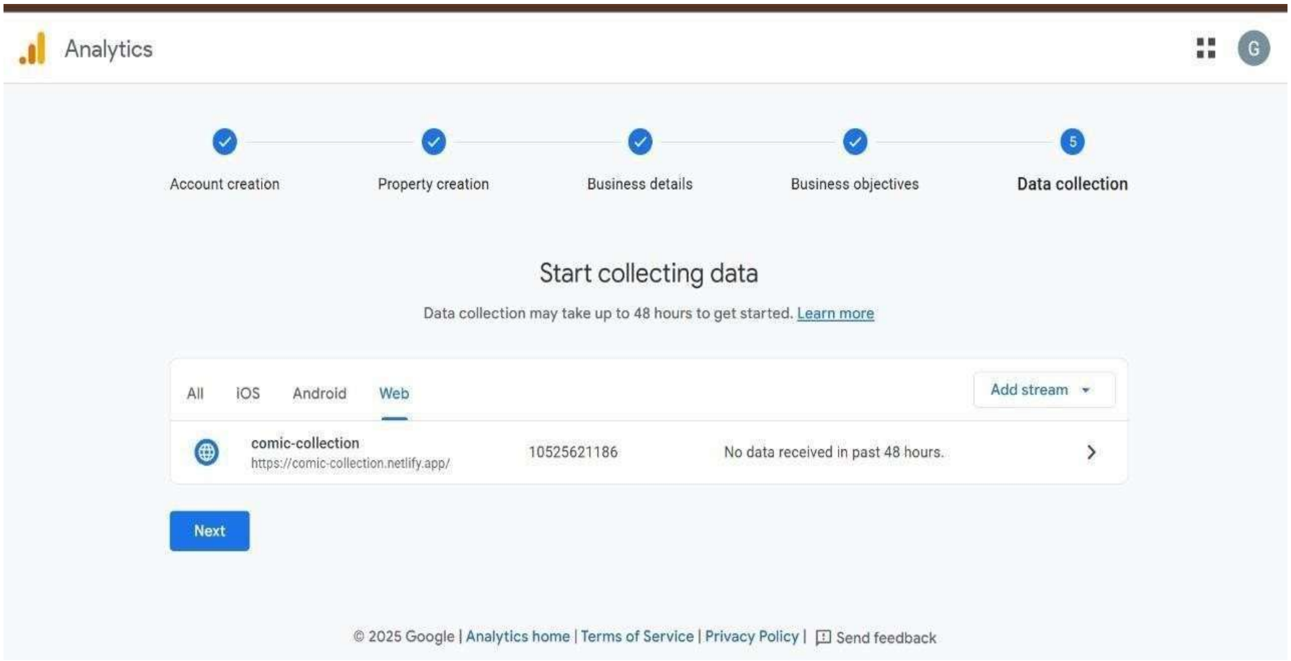
5 stars

Review by gunjan

Best comic ever

3 stars

3.2 Google Ananlysis

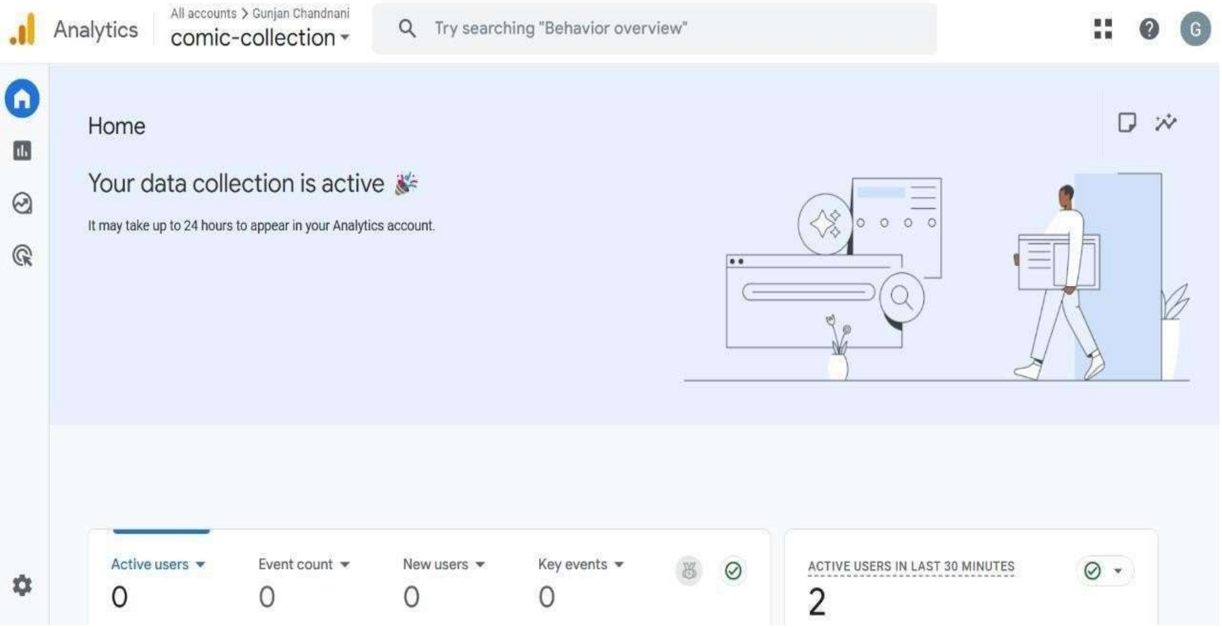


Start collecting data

Data collection may take up to 48 hours to get started. Learn more

comic-collection https://comic-collection.netlify.app/	10525621186	No data received in past 48 hours.

© 2025 Google | Analytics home | Terms of Service | Privacy Policy | Send feedback


Analytics
All accounts > Gunjan Chandnani
comic-collection
Try searching "Behavior overview"
G

Home

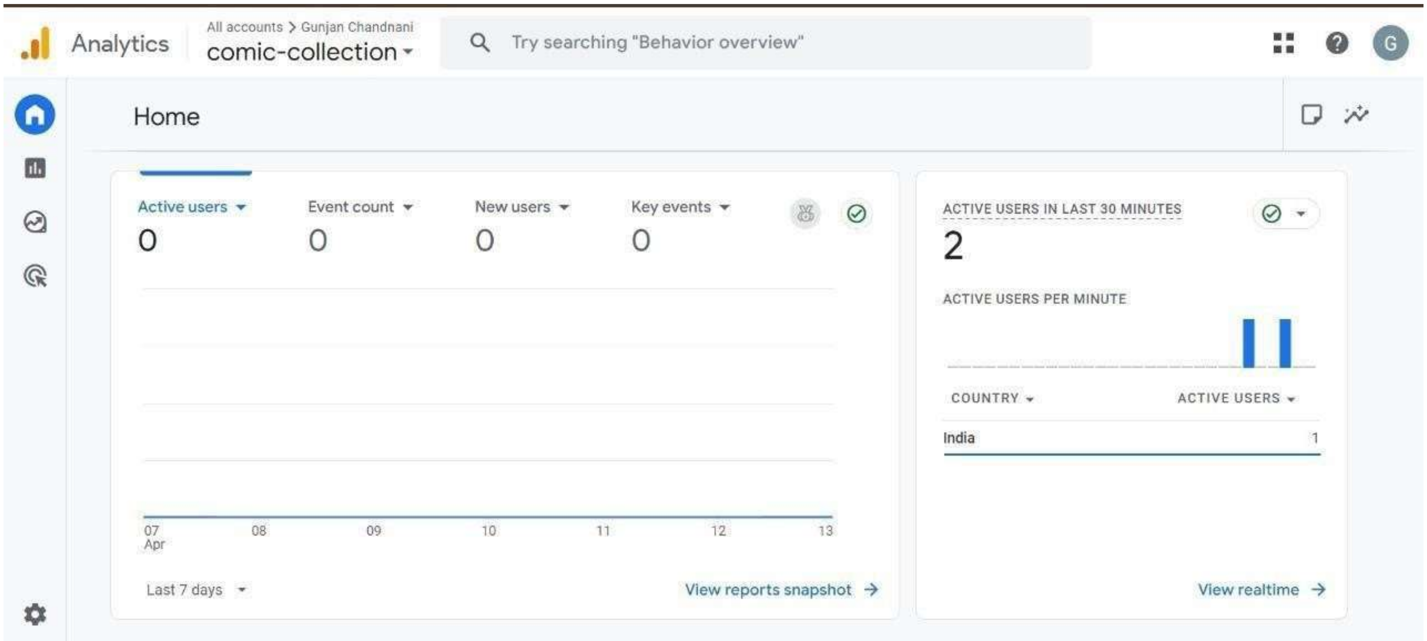
Your data collection is active 🎉

It may take up to 24 hours to appear in your Analytics account.

Active users	Event count	New users	Key events
0	0	0	0

ACTIVE USERS IN LAST 30 MINUTES

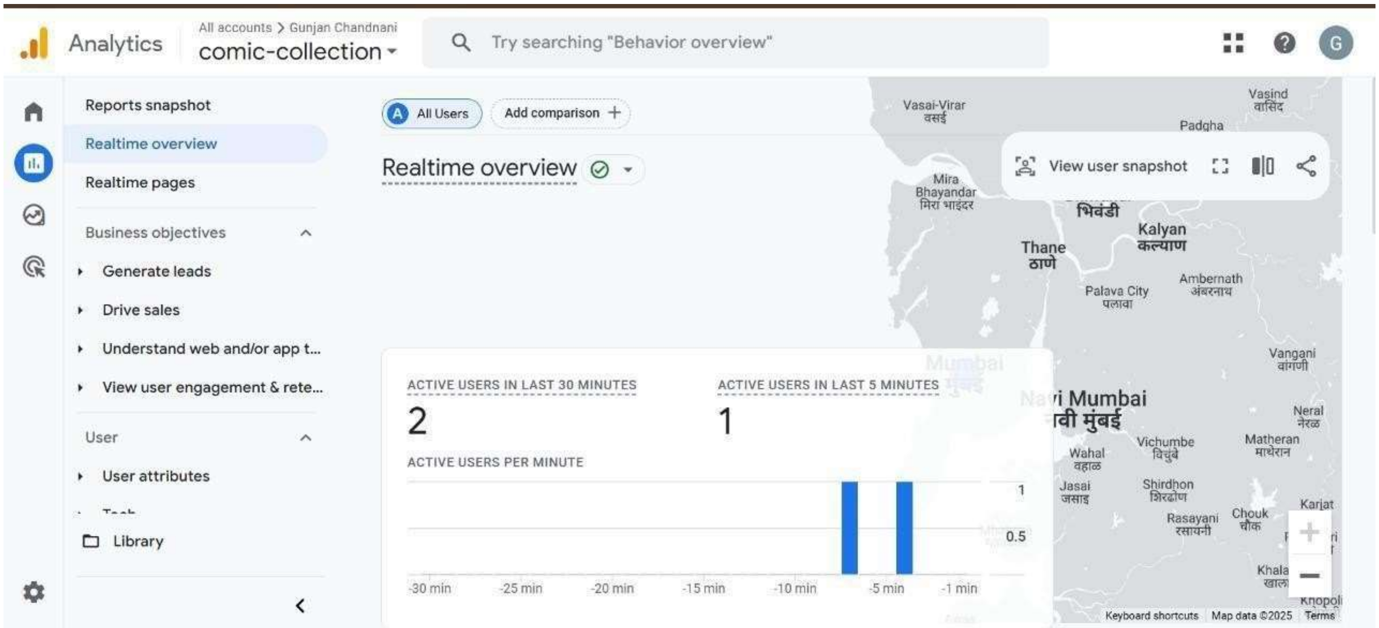
2



Active users	Event count	New users	Key events
0	0	0	0

ACTIVE USERS IN LAST 30 MINUTES
2

COUNTRY	ACTIVE USERS
India	1



Realtime overview

ACTIVE USERS IN LAST 30 MINUTES
2

ACTIVE USERS IN LAST 5 MINUTES
1

Chapter 4

Conclusion

4.2 Conclusion

The Comic Collection and Review Platform successfully integrates modern web technologies to deliver a seamless user experience for comic enthusiasts. By combining **Angular**, **TypeScript**, **CSS**, and **Flask** on the frontend with **MongoDB** on the backend, the platform enables users to effortlessly manage their comic collections, write reviews, and revisit their favorite titles.

This project demonstrates the power of full-stack development using a component-based frontend, efficient RESTful APIs, and a flexible NoSQL database. It also highlights the importance of user-centered design, data management, and scalability in modern web applications.

4.3 Reference:

[5] Angular: https://youtu.be/0LhBvp8qpro?si=NBads_TQ6T_wyoew
[6] Flask: https://youtu.be/oA8brF3w5XQ?si=sx1v6m9ZdxEIumzK
[7] Deploy: https://youtu.be/9srnyNC1e_o?si=2aRIo9OPAfKJyVpw
[8] Mongodb: https://youtu.be/J6mDkcqU_ZE?si=8v9Oka3fFse4UUUU