

# Viral Biyawala

Surat, Gujarat, India | viralbiyawala@gmail.com | +(91) 8238178802

Portfolio: <https://viralbiyawala.pythonanywhere.com/>

GitHub: [github.com/viralbiyawala](https://github.com/viralbiyawala)

LinkedIn: [linkedin.com/in/viralbiyawala](https://www.linkedin.com/in/viralbiyawala)

## Index

---

1. Education
2. Experience
3. Projects
4. Skills
5. Certifications
6. Achievements

## Education

---

### Bachelor of Technology in Computer Science and Engineering

Institution: Indian Institute of Information Technology Surat

Duration: Dec 2021 - May 2025

CGPA: 9.33/10.0

### Higher Secondary Certificate (HSC) in Science PCM

Institution: Gujarat Secondary and Higher Secondary Education Board

Duration: Mar 2020 - Mar 2021

CGPA: 88.46%

### Secondary School Certificate (SSC) in Science

Institution: Gujarat Secondary and Higher Secondary Education Board

Duration: Mar 2018 - Mar 2019

CGPA: 89.00%

## Experience

---

### AI/ML Intern at Build Macro

Duration: Jul 2024 - Sep 2024

- Engineered a robust AI solution employing OCR and YOLO technologies to automatically identify windows, doors, and walls from floor plans
- Reduced the manual processing time from 1 hour to 5 minutes by automating object detection and area calculations
- Deployed the system on Google Cloud Platform with 85 accuracy in detecting architectural elements

## Projects

---

### This are the projects which i have done

1. Stanford LLM Tutor
2. Recipe Site Traffic Prediction

- 3. MindCare - Mental Health Awareness
- 4. LearnAI Studio
- 5. Attrition Analysis | Atlas-Labs
- 6. Life Saver - Blood Bank Management System
- 7. ViralShare
- 8. University Enrollment Prediction
- 9. Analysis of Android Apps
- 10. Portfolio
- 11. Data Science Projects

## Stanford LLM Tutor

Stanford LLM Tutor is an AI chat assistant designed for understanding Stanford University's Large Language Models lecture notes. It uses structured JSON data, FAISS vector databases for quick indexing, and advanced algorithms to provide accurate explanations based on Stanford's LLM content. Integrated with Hugging Face's pipeline, it delivers tailored responses to user queries.

Live L

GitHub: [https://github.com/ViralBiyawala/Stanford\\_LLM\\_Tutor](https://github.com/ViralBiyawala/Stanford_LLM_Tutor)

# Stanford LLM Tutor | AI Bot

### ## Project Overview

This repository contains the implementation of an AI bot built using a transformer model (`gpt2`) from Hugging Face. The chatbot leverages FAISS for vector database storage to efficiently match user queries with relevant data. The data used for training and response generation was scraped from the official Stanford LLM course.

### ## Features

- **Language Model**: Utilizes the `gpt2` model from Hugging Face.
- **Vector Database**: Implements FAISS to store and retrieve keys efficiently.
- **Data Sources**: Scraped from various lectures of the Stanford LLM course.
- **Content Types**: Handles various content types including paragraphs, tables, equations, links, ordered lists, and unordered lists.
- **Query Matching**: Matches user queries to the top 2 closest keys using FAISS and constructs a prompt with the retrieved data.

### ## How It Works

1. **Data Scraping**: The data is scraped from various lectures of the Stanford LLM course. The `h2`, `h3`, and `**` tags serve as keys, and the corresponding content is categorized into paragraphs, tables, links, equations, ordered lists, and unordered lists.**
2. **Vector Database (FAISS)**: The keys are stored in a FAISS vector database using L2 distance for efficient retrieval. When a user query is received, FAISS finds the top 2 closest matching keys based on vector similarity.
3. **Prompt Generation**: The chatbot constructs a structured prompt using the data retrieved from FAISS. This prompt includes paragraphs, tables, equations, links, ordered lists, and unordered lists as relevant to the matched

keys.

4. **Response Generation**: The constructed prompt is fed into the GPT-2 model to generate a coherent and relevant response to the user query.

### ## Data Schema

The data scraped from the Stanford LLM course lectures have the following schema:

```
...
key1:{
  {
    'paragraphs': [],
    'tables': [],
    'links': [],
    'equations': [],
    'ordered_lists': [],
    'unordered_lists': []
  } }
key2:{
  {
    'paragraphs': [],
    'tables': [],
    'links': [],
    'equations': [],
    'ordered_lists': [],
    'unordered_lists': []
  } }
...
```

Each key corresponds to ``h2``, ``h3``, or ``<strong>`` tags from the lecture pages. The data associated with each key includes paragraphs, tables, links, equations, ordered lists, and unordered lists if they exist.

### ## Example Usage

1. **User Query**: "What are Benefits and Harms?"
2. **FAISS Retrieval**: The query is matched to the top 2 closest keys in the vector database using L2 distance.
3. **Prompt Construction**:

```
...

# Create a structured prompt
prompt = f"Question: {query}\n\n"

# Add top 2 matched sections
```

```

prompt += f"Sections:**\n- {result_key1}\n- {result_key2}\n\n"

# Add content to the prompt
for result_key, result_content in [(result_key1, result_content1), (result_key2, result_content2)]:
    if result_content.get('paragraphs'):
        prompt += "**Paragraphs:**\n" + "\n".join(result_content['paragraphs']) + "\n\n"
    if result_content.get('ordered_lists'):
        prompt += "**Ordered Lists:**\n" + "\n".join(["\n".join(ol) for ol in result_content['ordered_lists']]) + "\n\n"
    if result_content.get('unordered_lists'):
        prompt += "**Unordered Lists:**\n" + "\n".join(["\n".join(ul) for ul in result_content['unordered_lists']]) +
"\n\n"
    if result_content.get('tables'):
        prompt += "**Tables:**\n" + "\n".join(["\n".join(table) for table in result_content['tables']]) + "\n\n"
    if result_content.get('links'):
        prompt += "**Links:**\n" + "\n".join(result_content['links']) + "\n\n"
    if result_content.get('equations'):
        prompt += "**Equations:**\n" + "\n".join(result_content['equations']) + "\n\n"

# Add a closing statement
prompt += "Answer is : "

# Define max_length
max_length = min(len(prompt) + 100, 750)

# Generate response
response = generator(prompt[:750], max_length=max_length, num_return_sequences=1, truncation=True,
pad_token_id=50256)
...

```

4. **Generated Response**: The GPT-2 model uses the prompt to generate a detailed response.

## ## Example Screenshot

![Example Screenshot](example\_screenshotv2.png)

This screenshot shows an example interaction where the chatbot responds to a user query about the basics of LLMs.

## ## Training Details

- **Platform**: Kaggle
- **Hardware**: CPU

The model was trained on Kaggle using CPU resources.

## ## Contributing

Contributions are welcome! Please open an issue or submit a pull request for any improvements or new features.

## ## License

This project is licensed under the MIT License. See the [LICENSE](LICENSE) file for details.

## ## Acknowledgements

- [Hugging Face](https://huggingface.co/)
- [FAISS](https://github.com/facebookresearch/faiss)
- [Stanford LLM Course](https://stanford-cs324.github.io/winter2022/)

Live Link: <https://www.kaggle.com/code/viralbiyawala/llm-bot-use>

GitHub: [https://github.com/ViralBiyawala/Stanford\\_LLM\\_Tutor](https://github.com/ViralBiyawala/Stanford_LLM_Tutor)

# Stanford LLM Tutor | AI Bot

## ## Project Overview

This repository contains the implementation of an AI bot built using a transformer model (`gpt2`) from Hugging Face. The chatbot leverages FAISS for vector database storage to efficiently match user queries with relevant data. The data used for training and response generation was scraped from the official Stanford LLM course.

## ## Features

- **Language Model**: Utilizes the `gpt2` model from Hugging Face.
- **Vector Database**: Implements FAISS to store and retrieve keys efficiently.
- **Data Sources**: Scraped from various lectures of the Stanford LLM course.
- **Content Types**: Handles various content types including paragraphs, tables, equations, links, ordered lists, and unordered lists.
- **Query Matching**: Matches user queries to the top 2 closest keys using FAISS and constructs a prompt with the retrieved data.

## ## How It Works

- Data Scraping**: The data is scraped from various lectures of the Stanford LLM course. The `h2`, `h3`, and `**` tags serve as keys, and the corresponding content is categorized into paragraphs, tables, links, equations, ordered lists, and unordered lists.**
- Vector Database (FAISS)**: The keys are stored in a FAISS vector database using L2 distance for efficient retrieval. When a user query is received, FAISS finds the top 2 closest matching keys based on vector similarity.
- Prompt Generation**: The chatbot constructs a structured prompt using the data retrieved from FAISS. This prompt includes paragraphs, tables, equations, links, ordered lists, and unordered lists as relevant to the matched

keys.

4. **Response Generation**: The constructed prompt is fed into the GPT-2 model to generate a coherent and relevant response to the user query.

### ## Data Schema

The data scraped from the Stanford LLM course lectures have the following schema:

```
...
key1:{
  {
    'paragraphs': [],
    'tables': [],
    'links': [],
    'equations': [],
    'ordered_lists': [],
    'unordered_lists': []
  } }
key2:{
  {
    'paragraphs': [],
    'tables': [],
    'links': [],
    'equations': [],
    'ordered_lists': [],
    'unordered_lists': []
  } }
...
```

Each key corresponds to ``h2``, ``h3``, or ``<strong>`` tags from the lecture pages. The data associated with each key includes paragraphs, tables, links, equations, ordered lists, and unordered lists if they exist.

### ## Example Usage

1. **User Query**: "What are Benefits and Harms?"
2. **FAISS Retrieval**: The query is matched to the top 2 closest keys in the vector database using L2 distance.
3. **Prompt Construction**:

```
...

# Create a structured prompt
prompt = f"Question: {query}\n\n"

# Add top 2 matched sections
```

```

prompt += f"Sections:**\n- {result_key1}\n- {result_key2}\n\n"

# Add content to the prompt
for result_key, result_content in [(result_key1, result_content1), (result_key2, result_content2)]:
    if result_content.get('paragraphs'):
        prompt += "**Paragraphs:**\n" + "\n".join(result_content['paragraphs']) + "\n\n"
    if result_content.get('ordered_lists'):
        prompt += "**Ordered Lists:**\n" + "\n".join(["\n".join(ol) for ol in result_content['ordered_lists']]) + "\n\n"
    if result_content.get('unordered_lists'):
        prompt += "**Unordered Lists:**\n" + "\n".join(["\n".join(ul) for ul in result_content['unordered_lists']]) +
"\n\n"
    if result_content.get('tables'):
        prompt += "**Tables:**\n" + "\n".join(["\n".join(table) for table in result_content['tables']]) + "\n\n"
    if result_content.get('links'):
        prompt += "**Links:**\n" + "\n".join(result_content['links']) + "\n\n"
    if result_content.get('equations'):
        prompt += "**Equations:**\n" + "\n".join(result_content['equations']) + "\n\n"

# Add a closing statement
prompt += "Answer is : "

# Define max_length
max_length = min(len(prompt) + 100, 750)

# Generate response
response = generator(prompt[:750], max_length=max_length, num_return_sequences=1, truncation=True,
pad_token_id=50256)
...

```

4. **Generated Response**: The GPT-2 model uses the prompt to generate a detailed response.

## ## Example Screenshot

![Example Screenshot](example\_screenshotv2.png)

This screenshot shows an example interaction where the chatbot responds to a user query about the basics of LLMs.

## ## Training Details

- **Platform**: Kaggle
- **Hardware**: CPU

The model was trained on Kaggle using CPU resources.

## ## Contributing

Contributions are welcome! Please open an issue or submit a pull request for any improvements or new features.

## License

This project is licensed under the MIT License. See the [LICENSE](LICENSE) file for details.

## Acknowledgements

- [Hugging Face](https://huggingface.co/)
- [FAISS](https://github.com/facebookresearch/faiss)
- [Stanford LLM Course](https://stanford-cs324.github.io/winter2022/)

**Recipe Site Traffic Prediction**

By meticulously scrutinizing data patterns, I achieved an outstanding 83% accuracy rate. Employing a blend of intricate feature selection, precise hyperparameter adjustments, and advanced ensemble voting strategies, I significantly amplified model efficacy. The result? A striking 40% average surge in website traffic, fueled by our ability to deliver spot-on daily recipe forecasts.

Live L

Live Link: <https://app.datacamp.com/workspace/w/058082b0-6db9-4b7e-a510-faa8de89f91d>

**MindCare - Mental Health Awareness**

I led the backend development and seamlessly integrated Machine Learning, merging React (Vite framework), Django, and MongoDB into a cohesive system for MindCare. This innovative amalgamation forms the backbone for swiftly identifying concerning posts, utilizing email mechanisms to notify users, and introducing a novel quiz-based yoga recommender for enhancing holistic well-being, all augmented by REST API architecture for seamless data transfer.

Live L

GitHub: <https://github.com/ViralBiyawala/MindCare>

# MindCare

MindCare is a project dedicated to raising awareness about mental health issues and providing support through personalized recommendations and an interactive platform.

## Table of Contents

- [Description](#description)
- [About](#about)
- [Team Members](#team-members)
- [Features](#features)
- [Glimpse of Website](#glimpse-of-website)
- [Usage](#usage)

## Description

MindCare is designed to educate users about mental health conditions such as stress, depression, and anxiety. It



offers personalized recommendations based on quizzes and provides a platform for users to interact and seek support. The project utilizes React with the Vite framework for the frontend, Django for the backend, and MongoDB as the database. Additionally, it integrates machine learning algorithms to analyze user content, identify potentially concerning posts, and intervene when necessary. This approach ensures that users receive timely support and resources tailored to their specific needs.

## ## About

MindCare is a web application developed to promote mental well-being and provide resources for individuals facing mental health challenges. It offers educational content, personalized recommendations, and a supportive community forum to facilitate discussion and support. The platform aims to create a safe space where users can openly share their experiences, access valuable information, and receive encouragement from peers and professionals. By leveraging technology, MindCare strives to bridge the gap between those seeking help and the resources available to them, fostering a more inclusive and supportive environment for mental health care.

## ## Team Members

- [Shivam Sikotra (UI21CS55)](<https://github.com/ShivamSikotra11/>)
- [Shyam Patel (UI21CS57)](<https://github.com/shyam2024>)
- [Viral Biyawala (UI21CS66)](<https://github.com/ViralBiyawala>)

## ## Features

- **Quizzes**: Assess mental health status through interactive quizzes designed by mental health professionals. These quizzes help identify symptoms of stress, depression, and anxiety, and provide a baseline for personalized recommendations.
- **Personalized Recommendations**: Based on quiz results, users receive tailored suggestions such as yoga practices, meditation techniques, and lifestyle changes that can help improve their mental health. These recommendations are designed to be practical and easy to implement in daily life.
- **Educational Content**: Access a wealth of information on various mental health topics, including the symptoms, causes, and treatments of stress, depression, and anxiety. The educational content is curated to be engaging and informative, helping users gain a better understanding of their mental health.
- **Interactive Platform**: Engage with a community of users who are also focused on improving their mental health. The platform supports real-time interactions, allowing users to share experiences, offer support, and build connections with others who understand their struggles.
- **Forum**: A dedicated space for users to discuss their mental health journeys, ask questions, and seek advice from peers and mental health professionals. The forum is moderated to ensure a safe and respectful environment for all participants.
- **Machine Learning**: Utilizes advanced machine learning algorithms to monitor user posts and detect signs of distress or harmful behavior. When such content is identified, the system can alert moderators or provide users with resources and support options to address their needs.
- **Accessibility**: Designed with inclusivity in mind, MindCare includes features that ensure accessibility for users with disabilities.
- **Responsive Design**: Optimized for seamless viewing and interaction across all devices, including desktops, tablets, and smartphones. This ensures that users can access MindCare's resources and community support wherever they are.

## ## Glimpse of Website

### ### Blogs

![Blogs](images/blog.jpg)

### ### Quiz

![Quiz](images/quiz.jpg)

### ### Interaction Page

![Interaction Page](images/interact.jpg)

### ### Graphs on Profile Page

![Graphs on Profile Page](images/profile.jpg)

### ## Usage

### Live Link : <https://mindcare-sable.vercel.app/>

### ### Steps :

1. **Registration and Login**: Create a new account or log in to access your existing account.
2. **Home Page Tour**: Follow the guided tour to familiarize yourself with the website's features.
3. **Taking the Quiz**: Complete the mental health quiz to receive personalized recommendations.
4. **Viewing Suggestions**: Explore general insights and solutions for stress, depression, and anxiety.
5. **Interacting with the Community**: Post, read, like, and comment on user posts to engage with the community.
6. **Receiving Alerts**: The system monitors posts for concerning content and alerts appropriate parties.
7. **Profile Management**: View and update your profile information and track your quiz history.
8. **Logging Out**: Securely log out of your account when finished using the website.

### ## Installation

Clone the repository using `git clone https://github.com/ShivamSikotra11/MHA.git`.

### ### Backend

1. Navigate to the `backend` directory.`
2. Install dependencies using `pip install -r requirements.txt`.`
3. Start the Django server using `python manage.py runserver`.`

### ### Frontend

1. Navigate to the `frontend` directory.`
2. Install dependencies using `npm install`.`
3. Start the Vite development server using `npm run dev`.`

Live Link: <https://mindcare-sable.vercel.app/>

GitHub: <https://github.com/ViralBiyawala/MindCare>

# MindCare

MindCare is a project dedicated to raising awareness about mental health issues and providing support through

personalized recommendations and an interactive platform.

## ## Table of Contents

- [Description](#description)
- [About](#about)
- [Team Members](#team-members)
- [Features](#features)
- [Glimpse of Website](#glimpse-of-website)
- [Usage](#usage)

## ## Description

MindCare is designed to educate users about mental health conditions such as stress, depression, and anxiety. It offers personalized recommendations based on quizzes and provides a platform for users to interact and seek support. The project utilizes React with the Vite framework for the frontend, Django for the backend, and MongoDB as the database. Additionally, it integrates machine learning algorithms to analyze user content, identify potentially concerning posts, and intervene when necessary. This approach ensures that users receive timely support and resources tailored to their specific needs.

## ## About

MindCare is a web application developed to promote mental well-being and provide resources for individuals facing mental health challenges. It offers educational content, personalized recommendations, and a supportive community forum to facilitate discussion and support. The platform aims to create a safe space where users can openly share their experiences, access valuable information, and receive encouragement from peers and professionals. By leveraging technology, MindCare strives to bridge the gap between those seeking help and the resources available to them, fostering a more inclusive and supportive environment for mental health care.

## ## Team Members

- [Shivam Sikotra (UI21CS55)](<https://github.com/ShivamSikotra11/>)
- [Shyam Patel (UI21CS57)](<https://github.com/shyam2024>)
- [Viral Biyawala (UI21CS66)](<https://github.com/ViralBiyawala>)

## ## Features

- **Quizzes**: Assess mental health status through interactive quizzes designed by mental health professionals. These quizzes help identify symptoms of stress, depression, and anxiety, and provide a baseline for personalized recommendations.
- **Personalized Recommendations**: Based on quiz results, users receive tailored suggestions such as yoga practices, meditation techniques, and lifestyle changes that can help improve their mental health. These recommendations are designed to be practical and easy to implement in daily life.
- **Educational Content**: Access a wealth of information on various mental health topics, including the symptoms, causes, and treatments of stress, depression, and anxiety. The educational content is curated to be engaging and informative, helping users gain a better understanding of their mental health.
- **Interactive Platform**: Engage with a community of users who are also focused on improving their mental health. The platform supports real-time interactions, allowing users to share experiences, offer support, and build connections with others who understand their struggles.
- **Forum**: A dedicated space for users to discuss their mental health journeys, ask questions, and seek advice

from peers and mental health professionals. The forum is moderated to ensure a safe and respectful environment for all participants.

- **Machine Learning**: Utilizes advanced machine learning algorithms to monitor user posts and detect signs of distress or harmful behavior. When such content is identified, the system can alert moderators or provide users with resources and support options to address their needs.
- **Accessibility**: Designed with inclusivity in mind, MindCare includes features that ensure accessibility for users with disabilities.
- **Responsive Design**: Optimized for seamless viewing and interaction across all devices, including desktops, tablets, and smartphones. This ensures that users can access MindCare's resources and community support wherever they are.

## ## Glimpse of Website

### ### Blogs

![Blogs](images/blog.jpg)

### ### Quiz

![Quiz](images/quiz.jpg)

### ### Interaction Page

![Interaction Page](images/interact.jpg)

### ### Graphs on Profile Page

![Graphs on Profile Page](images/profile.jpg)

## ## Usage

### Live Link : <https://mindcare-sable.vercel.app/>

### ### Steps :

1. **Registration and Login**: Create a new account or log in to access your existing account.
2. **Home Page Tour**: Follow the guided tour to familiarize yourself with the website's features.
3. **Taking the Quiz**: Complete the mental health quiz to receive personalized recommendations.
4. **Viewing Suggestions**: Explore general insights and solutions for stress, depression, and anxiety.
5. **Interacting with the Community**: Post, read, like, and comment on user posts to engage with the community.
6. **Receiving Alerts**: The system monitors posts for concerning content and alerts appropriate parties.
7. **Profile Management**: View and update your profile information and track your quiz history.
8. **Logging Out**: Securely log out of your account when finished using the website.

## ## Installation

Clone the repository using `git clone https://github.com/ShivamSikotra11/MHA.git`.

### ### Backend

1. Navigate to the `backend` directory.`
2. Install dependencies using `pip install -r requirements.txt`.`
3. Start the Django server using `python manage.py runserver`.`

### ### Frontend

1. Navigate to the `frontend` directory.
2. Install dependencies using `npm install`.
3. Start the Vite development server using `npm run dev`.

## LearnAI Studio

LearnAI Studio is a versatile platform designed to provide users with structured, modular courses by leveraging LLMs and generative AI. We integrated advanced technologies, including the Gemini API and Pinecone, to enhance vector and semantic search capabilities, achieving a 40% improvement in accuracy. Redis caching was also employed to optimize data retrieval, resulting in a 35% performance boost and reducing response times from 2095.72ms to 1362.22ms. LearnAI Studio supports a diverse range of courses, including technical, non-technical, and interdisciplinary subjects, ensuring a comprehensive and adaptive learning experience for all users.

Live L

GitHub: <https://github.com/jitanshuraut/Learn-AI-Studio>

# Version 2 in Development

## Features will added:

- Personalized Structure: Upload images of course outlines, like college course syllabus, for a customized experience.
- RouteLLM: Uses a multimodal approach, applying different models for specific tasks—for example, OpenAI for math topic .
- Advancements will be made to improve the database.

# \*\*Learn AI Studio\*\*

![image (1)](<https://github.com/user-attachments/assets/29779487-ebfc-4c41-87f7-3e7a96d8a1a6>)

Welcome to **Learn AI Studio** – an innovative platform designed to revolutionize personalized learning experiences using the power of generative AI.

## [Try the Demo](<https://learn-ai-studio.vercel.app/>)

### ## How It Works

1. **Create Account**
  - Sign up and receive 5 free credits
2. **Input a Prompt**
  - Enter a topic or subject you want to explore
3. **AI Course Generation**
  - Our AI processes your prompt
  - Generates a structured course outline
  - Creates content modules and interactive elements

### ## Tech Stack

Category	Technologies
Frontend & Full Stack	Next.js
Backend	Prisma ORM, Redis
Authentication	Nuth Auth
Storage	Pinecone (vector storage), PostgreSQL
Course Generation	Gemini API (LLM)

## ## Getting Started

### 1. Clone the repository:

```
```bash
git clone https://github.com/jitanshuraut/Learn-AI-Studio.git
cd your-repo/
```
```

### 2. Install dependencies:

```
```bash
bun install
```
```

### 3. Set up the environment variables by creating a `.env` file and configuring the required variables.

### 4. Run the application:

```
```bash
bun dev
```
```

### 5. Open your browser and navigate to:

```
```url
http://localhost:3000
```
```

Follow the instructions on the page to complete the setup.

## ## How to contribute

To contribute via pull request, follow these steps:

1. Create an issue describing the feature you want to work on (Bugs, themes, plugins, new features or webhooks).
2. Write your code, tests and documentation, and format them with ``black``
3. Create a pull request describing your changes

## ## Have an idea? Notice a bug? Need help?

I'd love to hear your feedback! Feel free to log an issue on our [GitHub issues page](https://github.com/jitanshuraut/Learn-AI-Studio.git). If your question is more personal, [Jitanshu Raut on LinkedIn](https://www.linkedin.com/in/jitanshu-raut-438b76246/) is always open as well.

Live Link: <https://learn-ai-studio.vercel.app/>  
GitHub: <https://github.com/jitanshuraut/Learn-AI-Studio>

# Version 2 in Development

## Features will added:

- Personalized Structure: Upload images of course outlines, like college course syllabus, for a customized experience.
- RouteLLM: Uses a multimodal approach, applying different models for specific tasks—for example, OpenAI for math topic .
- Advancements will be made to improve the database.

# **Learn AI Studio**

!image (1)](<https://github.com/user-attachments/assets/29779487-ebfc-4c41-87f7-3e7a96d8a1a6>)

Welcome to **Learn AI Studio** – an innovative platform designed to revolutionize personalized learning experiences using the power of generative AI.

## [Try the Demo](https://learn-ai-studio.vercel.app/)

## How It Works

1. **Create Account**
  - Sign up and receive 5 free credits
2. **Input a Prompt**
  - Enter a topic or subject you want to explore
3. **AI Course Generation**
  - Our AI processes your prompt
  - Generates a structured course outline
  - Creates content modules and interactive elements

## Tech Stack

| Category | Technologies |  
|-----|-----|  
| Frontend & Full Stack | Next.js |  
| Backend | Prisma ORM, Redis |  
| Authentication | Nuth Auth |  
| Storage | Pinecone (vector storage), PostgreSQL |  
| Course Generation | Gemini API (LLM) |

## ## Getting Started

### 1. Clone the repository:

```
```bash
git clone https://github.com/jitanshuraud/Learn-AI-Studio.git
cd your-repo/
```
```

### 2. Install dependencies:

```
```bash
bun install
```
```

### 3. Set up the environment variables by creating a `.env` file and configuring the required variables.`

### 4. Run the application:

```
```bash
bun dev
```
```

### 5. Open your browser and navigate to:

```
```url
http://localhost:3000
```
```

Follow the instructions on the page to complete the setup.

## ## How to contribute

To contribute via pull request, follow these steps:

1. Create an issue describing the feature you want to work on (Bugs, themes, plugins, new features or webhooks).
2. Write your code, tests and documentation, and format them with `black``
3. Create a pull request describing your changes

## ## Have an idea? Notice a bug? Need help?

I'd love to hear your feedback! Feel free to log an issue on our [GitHub issues page](https://github.com/jitanshuraud/Learn-AI-Studio.git). If your question is more personal, [Jitanshu Raut on LinkedIn](https://www.linkedin.com/in/jitanshu-raut-438b76246/) is always open as well.

## Attrition Analysis | Atlas-Labs

This Power BI dashboard analyzes employee attrition, visualizing key metrics such as overall attrition rate, hiring



trends, and active employees by department and job role. It includes demographic insights, performance tracking, and detailed attrition analysis by various factors, helping stakeholders manage and reduce employee turnover effectively.

GitHu

## # ATLAS Labs Employee Attrition Dashboard | Power BI

### ## Overview

This Power BI dashboard analyzes employee attrition, visualizing key metrics such as overall attrition rate, hiring trends, and active employees by department and job role. It includes demographic insights, performance tracking, and detailed attrition analysis by various factors, helping stakeholders manage and reduce employee turnover effectively.

### ## Features

- **Attrition Rate Overview**: Displays the overall attrition rate and hiring trends.
- **Employee Demographics**: Visualizes employee distribution by age, gender, marital status, and ethnicity.
- **Performance Tracker**: Tracks individual employee performance metrics over the years.
  - WorkLife Balance
  - Job Satisfaction
  - Environment Satisfaction
  - Relationship Satisfaction
- **Detailed Attrition Analysis**: Examines attrition rates by department, job role, hire date, travel frequency, overtime, and tenure.

### ## Screenshots

Here are some screenshots of the dashboard:

#### ### Overview Page

![Overview](pictures/overview.png)

#### ### Demographics Page

![Demographics](pictures/demographics.png)

#### ### Model and Relationship view

![Model](pictures/model.jpg)

### ## Full PDF Report

For a detailed view, you can download the full PDF report [ATLAS\_ALBS.pdf](ATLAS\_LABS.pdf).

### ## How to Use

1. **Open Power BI Report**: Download and open the .pbix file in Power BI Desktop.
2. **Navigate the Pages**: The dashboard consists of four main pages: Overview, Demographics, Performance Tracker, and Attrition.
3. **Interactive Filters**: Use the interactive filters to drill down into specific segments of data.
4. **Sharing**: Publish the report to Power BI Service for real-time updates and collaborative access.

## ## Data Sources

The raw data used for analysis is [here](case-study-atlas-labs.zip).

The dashboard is powered by data from the following sources:

- Employee records including hire dates, department, job roles, and performance ratings.
- Demographic information such as age, gender, marital status, and ethnicity.
- Performance evaluation data.

## ## Installation

1. Download the .pbix file [ATLAS\_ALBS.pbix](ATLAS\_LABS.pbix).
2. Open Power BI Desktop.
3. Load the .pbix file to view the dashboard.

## ## Acknowledgments

- [\*\*ATLAS LABS\*\*](https://www.atlaslab.in/) for providing the dataset.

## ## Contribution

Feel free to contribute by submitting issues or pull requests. For major changes, please open an issue first to discuss what you would like to change.

GitHub: <https://github.com/ViralBiyawala/ATLAS-PowerBI>

# ATLAS Labs Employee Attrition Dashboard | Power BI

## ## Overview

This Power BI dashboard analyzes employee attrition, visualizing key metrics such as overall attrition rate, hiring trends, and active employees by department and job role. It includes demographic insights, performance tracking, and detailed attrition analysis by various factors, helping stakeholders manage and reduce employee turnover effectively.

## ## Features

- **Attrition Rate Overview**: Displays the overall attrition rate and hiring trends.
- **Employee Demographics**: Visualizes employee distribution by age, gender, marital status, and ethnicity.
- **Performance Tracker**: Tracks individual employee performance metrics over the years.
  - WorkLife Balance
  - Job Satisfaction
  - Environment Satisfaction
  - Relationship Satisfaction
- **Detailed Attrition Analysis**: Examines attrition rates by department, job role, hire date, travel frequency, overtime, and tenure.

## ## Screenshots

Here are some screenshots of the dashboard:

### ### Overview Page

![Overview](pictures/overview.png)

### ### Demographics Page

![Demographics](pictures/demographics.png)

### ### Model and Relationship view

![Model](pictures/model.jpg)

### ## Full PDF Report

For a detailed view, you can download the full PDF report [ATLAS\_ALBS.pdf](ATLAS\_LABS.pdf).

### ## How to Use

1. **\*\*Open Power BI Report\*\***: Download and open the .pbix file in Power BI Desktop.
2. **\*\*Navigate the Pages\*\***: The dashboard consists of four main pages: Overview, Demographics, Performance Tracker, and Attrition.
3. **\*\*Interactive Filters\*\***: Use the interactive filters to drill down into specific segments of data.
4. **\*\*Sharing\*\***: Publish the report to Power BI Service for real-time updates and collaborative access.

### ## Data Sources

The raw data used for analysis is [here](case-study-atlas-labs.zip).

The dashboard is powered by data from the following sources:

- Employee records including hire dates, department, job roles, and performance ratings.
- Demographic information such as age, gender, marital status, and ethnicity.
- Performance evaluation data.

### ## Installation

1. Download the .pbix file [ATLAS\_ALBS.pbix](ATLAS\_LABS.pbix).
2. Open Power BI Desktop.
3. Load the .pbix file to view the dashboard.

### ## Acknowledgments

- **\*\*ATLAS LABS\*\***[(https://www.atlaslab.in/)] for providing the dataset.

### ## Contribution

Feel free to contribute by submitting issues or pull requests. For major changes, please open an issue first to discuss what you would like to change.

## Life Saver - Blood Bank Management System

I developed a robust backend for the LifeSaver using Flask and SQLite. It included a multi-tiered login system, OTP verification, and efficient blood distribution across cities. It also incorporated visualization tools and an admin interface for seamless process management. The goal was to redefine blood donation processes, showcasing our full

stack web development expertise.

Live L

GitHub: <https://github.com/ViralBiyawala/BBMS>

## # Blood Bank Management System

The Blood Bank Management System is a comprehensive web-based platform designed to revolutionize the management of blood donations, requests, and inventory. This project aims to address the challenges and inefficiencies of existing blood bank management systems, such as outdated interfaces, manual processes, and lack of real-time coordination. By leveraging modern web technologies, we've created a dynamic solution that ensures seamless interaction and collaboration among users, ultimately enhancing the availability and accessibility of life-saving blood products.

### ## Features

**\*\*Tailored Login System:\*\*** A sophisticated three-type of login system caters to donors, hospitals, and administrators, providing each user group with personalized functionalities and intuitive interfaces.

**\*\*Enhanced Security Measures:\*\*** Robust registration and authentication mechanisms using OTP verification ensure secure access for all users, safeguarding sensitive information and maintaining data integrity.

**\*\*Efficient Donor Management:\*\*** Donors can effortlessly schedule appointments, access their donation history, and update profile details, streamlining the blood donation process.

**\*\*Streamlined Hospital Management:\*\*** Hospitals facilitate seamless blood request submissions, efficient recipient record management, and effortless blood group searches, enhancing patient care delivery.

**\*\*Effective Blood Inventory Management:\*\*** The system efficiently tracks blood donations, manages inventory levels, and facilitates timely distribution to meet the demands of healthcare facilities.

**\*\*Proactive Notification System:\*\*** Dynamic notification system delivers instant alerts to donors, hospitals, and administrators, keeping them informed about blood requests, donations, and inventory updates in real-time.

**\*\*Administrative Oversight:\*\*** The admin panel grants administrators comprehensive control over user management, system monitoring, and report generation, empowering administrators with advanced analytics and reporting tools to monitor system performance, analyze trends, and derive actionable insights for informed decision-making.

### ## User Actions

#### ### Donor

**\*\*Register and Profile Creation:\*\*** Seamlessly register as a donor and craft a comprehensive profile, including personal details, blood type, and donation history.

**\*\*Appointment Scheduling:\*\*** Effortlessly schedule blood donation appointments and receive timely reminders and notifications regarding your donation status.

**\*\*Donation Tracking:\*\*** Easily monitor your donation history and access insightful statistics to track your impactful contributions.

**\*\*Donation Certification:\*\*** Upon successful donation, receive a prestigious Donation Certificate as a token of appreciation

### ### Hospital

**\*\*Registration and Profile Establishment:\*\*** Swiftly register as a hospital entity and establish a robust profile, furnishing essential hospital and blood bank particulars.

**\*\*Blood Donation Requests:\*\*** Efficiently solicit blood donations from donors and stay updated in real-time with dynamic updates on donation statuses and availability.

**\*\*Patient Management:\*\*** Streamline patient management operations by seamlessly adding and tracking patient information within the dedicated hospital section.

### ### Administrator

**\*\*User Management:\*\*** Seamlessly manage the accounts of donors, hospitals, and fellow administrators, exercising granular control over permissions and access levels.

**\*\*System Oversight and Analytics:\*\*** Diligently monitor system performance, leveraging advanced analytics to generate comprehensive reports on donation history, inventory levels, and user engagement metrics.

**\*\*Customization and Configuration:\*\*** Tailor the system to organizational needs by configuring system settings and fine-tuning the user interface and functionalities to optimize user experience and operational efficiency.

## ## Technologies

**\*\*Backend Framework:\*\*** Flask powers the backend, offering a lightweight and flexible architecture for web development.

**\*\*Database System:\*\*** SQLite ensures reliability and scalability for data storage and retrieval.

**\*\*Frontend Technologies:\*\*** HTML, CSS, and JavaScript come together to create a rich and interactive user interface.

**\*\*Frontend Framework:\*\*** Bootstrap enhances the frontend with a responsive and customizable design system.

**\*\*Data Visualization:\*\*** Chart.js adds depth to the user experience with powerful and flexible charts and graphs.

## ## Installation

1. Clone the repository: `git clone https://github.com/username/blood-bank-management-system.git`

2. Install the dependencies: `pip install -r requirements.txt`
3. Run the application: `python app.py`

## ## Usage

1. Website Link: <https://lifesaver-uxlr.onrender.com/>
2. Navigate to the registration page and create an account as a donor, hospital, or blood bank.
3. Log in with your credentials.
4. Explore the respective functionalities based on your user role.
5. Schedule appointments, request blood, manage records, and more.

## ## Screenshots

### ### Login Page

![Login Page](login.jpg)

### ### OTP Verification Page

![OTP Verification Page](otp.jpg)

### ### Home Page

![Home Page](home.jpg)

### ### Appointment Page

![Appointment Page](appointment.jpg)

### ### Admin Dashboard

![Admin Dashboard](admin\_dashboard.jpg)

## ## License

This project is licensed under the Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International (CC BY-NC-ND 4.0) license. You can use this project for learning purposes, but you cannot modify or distribute it without permission. See the LICENSE.md file for details.

## ## Acknowledgments

- [Flask Documentation](<https://flask.palletsprojects.com/en/2.1.x/>)
- [SQLite Documentation](<https://www.sqlite.org/docs.html>)
- [HTML Tutorial](<https://www.w3schools.com/html/>)
- [CSS Tutorial](<https://www.w3schools.com/css/>)
- [JavaScript Tutorial](<https://www.w3schools.com/js/>)
- [Bootstrap Documentation](<https://getbootstrap.com/docs/5.1/getting-started/introduction/>)
- [Chart.js Documentation](<https://www.chartjs.org/docs/latest/>)

Live Link: <https://lifesaver.pythonanywhere.com/>

GitHub: <https://github.com/ViralBiyawala/BBMS>

## # Blood Bank Management System

The Blood Bank Management System is a comprehensive web-based platform designed to revolutionize the management of blood donations, requests, and inventory. This project aims to address the challenges and inefficiencies of existing blood bank management systems, such as outdated interfaces, manual processes, and lack of real-time coordination. By leveraging modern web technologies, we've created a dynamic solution that ensures seamless interaction and collaboration among users, ultimately enhancing the availability and accessibility of life-saving blood products.

### ## Features

**\*\*Tailored Login System:\*\*** A sophisticated three-type of login system caters to donors, hospitals, and administrators, providing each user group with personalized functionalities and intuitive interfaces.

**\*\*Enhanced Security Measures:\*\*** Robust registration and authentication mechanisms using OTP verification ensure secure access for all users, safeguarding sensitive information and maintaining data integrity.

**\*\*Efficient Donor Management:\*\*** Donors can effortlessly schedule appointments, access their donation history, and update profile details, streamlining the blood donation process.

**\*\*Streamlined Hospital Management:\*\*** Hospitals facilitate seamless blood request submissions, efficient recipient record management, and effortless blood group searches, enhancing patient care delivery.

**\*\*Effective Blood Inventory Management:\*\*** The system efficiently tracks blood donations, manages inventory levels, and facilitates timely distribution to meet the demands of healthcare facilities.

**\*\*Proactive Notification System:\*\*** Dynamic notification system delivers instant alerts to donors, hospitals, and administrators, keeping them informed about blood requests, donations, and inventory updates in real-time.

**\*\*Administrative Oversight:\*\*** The admin panel grants administrators comprehensive control over user management, system monitoring, and report generation, empowering administrators with advanced analytics and reporting tools to monitor system performance, analyze trends, and derive actionable insights for informed decision-making.

### ## User Actions

#### ### Donor

**\*\*Register and Profile Creation:\*\*** Seamlessly register as a donor and craft a comprehensive profile, including personal details, blood type, and donation history.

**\*\*Appointment Scheduling:\*\*** Effortlessly schedule blood donation appointments and receive timely reminders and notifications regarding your donation status.

**\*\*Donation Tracking:\*\*** Easily monitor your donation history and access insightful statistics to track your impactful

contributions.

**\*\*Donation Certification:\*\*** Upon successful donation, receive a prestigious Donation Certificate as a token of appreciation

### ### Hospital

**\*\*Registration and Profile Establishment:\*\*** Swiftly register as a hospital entity and establish a robust profile, furnishing essential hospital and blood bank particulars.

**\*\*Blood Donation Requests:\*\*** Efficiently solicit blood donations from donors and stay updated in real-time with dynamic updates on donation statuses and availability.

**\*\*Patient Management:\*\*** Streamline patient management operations by seamlessly adding and tracking patient information within the dedicated hospital section.

### ### Administrator

**\*\*User Management:\*\*** Seamlessly manage the accounts of donors, hospitals, and fellow administrators, exercising granular control over permissions and access levels.

**\*\*System Oversight and Analytics:\*\*** Diligently monitor system performance, leveraging advanced analytics to generate comprehensive reports on donation history, inventory levels, and user engagement metrics.

**\*\*Customization and Configuration:\*\*** Tailor the system to organizational needs by configuring system settings and fine-tuning the user interface and functionalities to optimize user experience and operational efficiency.

## ## Technologies

**\*\*Backend Framework:\*\*** Flask powers the backend, offering a lightweight and flexible architecture for web development.

**\*\*Database System:\*\*** SQLite ensures reliability and scalability for data storage and retrieval.

**\*\*Frontend Technologies:\*\*** HTML, CSS, and JavaScript come together to create a rich and interactive user interface.

**\*\*Frontend Framework:\*\*** Bootstrap enhances the frontend with a responsive and customizable design system.

**\*\*Data Visualization:\*\*** Chart.js adds depth to the user experience with powerful and flexible charts and graphs.

## ## Installation

1. Clone the repository: ``git clone https://github.com/username/blood-bank-management-system.git``
2. Install the dependencies: ``pip install -r requirements.txt``
3. Run the application: ``python app.py``



## ## Usage

1. Website Link: <https://lifesaver-uxlr.onrender.com/>
2. Navigate to the registration page and create an account as a donor, hospital, or blood bank.
3. Log in with your credentials.
4. Explore the respective functionalities based on your user role.
5. Schedule appointments, request blood, manage records, and more.

## ## Screenshots

### ### Login Page

![Login Page](login.jpg)

### ### OTP Verification Page

![OTP Verification Page](otp.jpg)

### ### Home Page

![Home Page](home.jpg)

### ### Appointment Page

![Appointment Page](appointment.jpg)

### ### Admin Dashboard

![Admin Dashboard](admin\_dashboard.jpg)

## ## License

This project is licensed under the Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International (CC BY-NC-ND 4.0) license. You can use this project for learning purposes, but you cannot modify or distribute it without permission. See the LICENSE.md file for details.

## ## Acknowledgments

- [Flask Documentation](<https://flask.palletsprojects.com/en/2.1.x/>)
- [SQLite Documentation](<https://www.sqlite.org/docs.html>)
- [HTML Tutorial](<https://www.w3schools.com/html/>)
- [CSS Tutorial](<https://www.w3schools.com/css/>)
- [JavaScript Tutorial](<https://www.w3schools.com/js/>)
- [Bootstrap Documentation](<https://getbootstrap.com/docs/5.1/getting-started/introduction/>)
- [Chart.js Documentation](<https://www.chartjs.org/docs/latest/>)

## ViralShare

ViralShare is a peer-to-peer file sharing application designed to streamline the exchange of files among users on a network. It offers a lightweight, secure, and user-friendly platform for setting up servers, connecting to peers, and

sharing files effortlessly. With built-in password protection and intuitive interface, ViralShare ensures seamless file sharing experiences while maintaining data security and user privacy.

GitHu

# Welcome to ViralShare!

ViralShare is a lightweight peer-to-peer file sharing application designed to facilitate easy and secure sharing of files among users on a network. With ViralShare, you can quickly set up your own file sharing server, connect to other peers, share your files, and download files shared by others.

## ## Getting Started

To start using ViralShare, follow these simple steps:

### ### 1. Create the Executable

- Make sure you have Python installed on your system. You can download it from the official Python website.
- Open a terminal or command prompt.
- Download the `ViralShare.py` and `build\_executable.py` scripts to your local machine.
- Run the following command to create the executable:  
``python build\_executable.py``
- This will install executable file `ViralShare.exe` in `dict` folder, and copy and paste it to your desired location.

### ### 2. Installing and Running ViralShare

- Once the download is complete, locate the downloaded ViralShare.exe file on your computer.
- Double-click on the ViralShare.exe file to launch the application.
- A terminal window will also open along with file sharing interface, displaying all information and error logging that happens live while running application.
- Two logging files, `access.log` and `error.log`, will be created at the same level as the application file.

### ### 3. Setting Up Your Server

- Upon launching ViralShare, you will be prompted to set up your server. Enter a port number and a password for your server. This will allow other users to connect to your server and share files.
- Click on the "Start Server" button to initiate your server.

### ### 4. Connecting to Other Peers

- To connect to other peers, you'll need their server information, including their host address, port number, and password.
- Enter the host address, port number, and password of the peer you want to connect to in the provided fields.

- Click on the "Connect to Peer" button to establish a connection with the peer.

### ### 5. Sharing Your Files

- ViralShare allows you to share files from a specific folder on your computer.
- All files that need to be shared must be in the folder `share` on the same level as the application file, which will be automatically build when you start the server.
- Click on the "List Files" button to view the files available for sharing from your designated share folder.

### ### 6. Downloading Files from Peers

- To download files shared by other connected peers, click on the "List All Files" button.
- This will display a list of files shared by other peers connected to your server.
- Enter the filename you want to download in the download box and hit the download button.
- The downloaded file will be stored in the `download` folder on the same level as the application file, which will be automatically build when you start the server.

## ## Additional Features

- **Password Protection:** ViralShare ensures that only authorized users can access and use the application. The application cannot be modified or distributed without explicit permission from the owner.
- **License:** Users are granted the right to use ViralShare for personal use only. Any modifications or distributions require prior approval from the owner.
- **Contributors:** This application was developed by:
  - [Biyawala Viral Deven](https://github.com/ViralBiyawala)
  - [Shivam Sikotra](https://github.com/ShivamSikotra11)
  - [Patel Shyam](https://github.com/shyam2024)

## For contributions, please contact [Biyawala Viral Deven](https://github.com/ViralBiyawala).

That's it! You're now ready to use ViralShare to share and download files securely with other users on your network. If you encounter any issues or have feedback, feel free to reach out to the owner for support.

GitHub: <https://github.com/ViralBiyawala/ViralShare>

# Welcome to ViralShare!

ViralShare is a lightweight peer-to-peer file sharing application designed to facilitate easy and secure sharing of files among users on a network. With ViralShare, you can quickly set up your own file sharing server, connect to other peers, share your files, and download files shared by others.

## ## Getting Started

To start using ViralShare, follow these simple steps:

### ### 1. Create the Executable

- Make sure you have Python installed on your system. You can download it from the official Python website.
- Open a terminal or command prompt.
- Download the `ViralShare.py` and `build\_executable.py` scripts to your local machine.
- Run the following command to create the executable:  
``python build\_executable.py``
- This will install executable file `ViralShare.exe` in `dict` folder, and copy and paste it to your desired location.

### ### 2. Installing and Running ViralShare

- Once the download is complete, locate the downloaded ViralShare.exe file on your computer.
- Double-click on the ViralShare.exe file to launch the application.
- A terminal window will also open along with file sharing interface, displaying all information and error logging that happens live while running application.
- Two logging files, `access.log` and `error.log`, will be created at the same level as the application file.

### ### 3. Setting Up Your Server

- Upon launching ViralShare, you will be prompted to set up your server. Enter a port number and a password for your server. This will allow other users to connect to your server and share files.
- Click on the "Start Server" button to initiate your server.

### ### 4. Connecting to Other Peers

- To connect to other peers, you'll need their server information, including their host address, port number, and password.
- Enter the host address, port number, and password of the peer you want to connect to in the provided fields.
- Click on the "Connect to Peer" button to establish a connection with the peer.

### ### 5. Sharing Your Files

- ViralShare allows you to share files from a specific folder on your computer.
- All files that need to be shared must be in the folder `share` on the same level as the application file, which will be automatically build when you start the server.
- Click on the "List Files" button to view the files available for sharing from your designated share folder.

### ### 6. Downloading Files from Peers

- To download files shared by other connected peers, click on the "List All Files" button.
- This will display a list of files shared by other peers connected to your server.

- Enter the filename you want to download in the download box and hit the download button.
- The downloaded file will be stored in the `download` folder on the same level as the application file, which will be automatically build when you start the server.

### ## Additional Features

- **Password Protection:** ViralShare ensures that only authorized users can access and use the application. The application cannot be modified or distributed without explicit permission from the owner.
- **License:** Users are granted the right to use ViralShare for personal use only. Any modifications or distributions require prior approval from the owner.
- **Contributors:** This application was developed by:
  - [Biyawala Viral Deven](https://github.com/ViralBiyawala)
  - [Shivam Sikotra](https://github.com/ShivamSikotra11)
  - [Patel Shyam](https://github.com/shyam2024)

## For contributions, please contact [Biyawala Viral Deven](https://github.com/ViralBiyawala).

That's it! You're now ready to use ViralShare to share and download files securely with other users on your network. If you encounter any issues or have feedback, feel free to reach out to the owner for support.

## University Enrollment Prediction

Employing Scikit-learn's robust linear regression algorithm, I meticulously crafted a predictive model renowned for its accuracy in forecasting student enrollments. Impressively, the model achieved a strikingly low RMSE of 0.3, underscoring its exceptional precision. Through thorough analysis of extensive historical enrollment data, I demonstrated the practical application of advanced machine learning techniques in real-world scenarios.

Live L

Live Link: <https://www.datacamp.com/datalab/w/8e67ad0d-b572-4c54-9a28-2819dd8e87fc>

## Analysis of Android Apps

In the project, I delved into sentiment analysis, meticulously examining around 10,000 apps sourced from the Google Play Store. Leveraging powerful tools like Plotly, Seaborn, and Pandas, I intricately decoded user reviews to unearth profound insights into their sentiments and preferences. This comprehensive analysis paved the way for informed decision-making and strategic planning in app development.

GitHu

GitHub: [https://github.com/ViralBiyawala/DS\\_ML\\_Projects/tree/main/The\\_Android\\_App\\_Market\\_on\\_Google\\_Play](https://github.com/ViralBiyawala/DS_ML_Projects/tree/main/The_Android_App_Market_on_Google_Play)

## Portfolio

My portfolio showcases a diverse range of projects, including AI chatbots, predictive models, and web applications, all meticulously crafted to demonstrate my technical prowess and creativity. By leveraging cutting-edge technologies and innovative approaches, I have developed a robust portfolio that reflects my passion for AI, machine learning, and software development.

Live L

GitHub: <https://github.com/ViralBiyawala/Portfolio>

# Portfolio

Live Link: <https://viralbiyawala.pythonanywhere.com/>

GitHub: <https://github.com/ViralBiyawala/Portfolio>

# Portfolio

Viral Biyawala Portfolio

## Data Science Projects

My data science projects encompass a wide array of topics, including predictive modeling, data analysis, and machine learning applications. By leveraging Python, SQL, and advanced data science libraries, I have developed a diverse portfolio that showcases my expertise in data wrangling, visualization, and predictive analytics. These projects highlight my ability to derive actionable insights from complex datasets and deliver innovative solutions to real-world problems.

GitHu

# DS\_ML\_Projects

> Projects for Learning for The Data Science , Data Analysis , Machine Learning

# Data

> All the Data files are uploaded in the datasets folder of Each Individual Project Folders

## Data Source and Usage

> The data used in this project was obtained from [Kaggle](<https://www.kaggle.com/>) under their terms and conditions. Please note the following regarding the data:

>

> - The dataset is subject to copyright and intellectual property laws.

> - The data is intended for non-commercial and learning purposes only.

> - Any usage beyond the permitted purposes may require explicit permission from the data provider.

> - Attribution to the original data source is required. Please refer to the appropriate acknowledgments section for details.

>

> Please ensure that you review and comply with Kaggle's terms and conditions and any additional licensing or usage restrictions specified by the data provider.

>

> For more information about the dataset and how to access it, please refer to the [Kaggle Terms page](<https://www.kaggle.com/terms>).

## License

> This project is licensed under the [MIT License](LICENSE). Feel free to modify and adapt the code according to your needs.

GitHub: [https://github.com/ViralBiyawala/DS\\_ML\\_Projects](https://github.com/ViralBiyawala/DS_ML_Projects)

# DS\_ML\_Projects

> Projects for Learning for The Data Science , Data Analysis , Machine Learning

# Data

> All the Data files are uploaded in the datasets folder of Each Individual Project Folders

## Data Source and Usage

> The data used in this project was obtained from [Kaggle](<https://www.kaggle.com/>) under their terms and

conditions. Please note the following regarding the data:

>

> - The dataset is subject to copyright and intellectual property laws.

> - The data is intended for non-commercial and learning purposes only.

> - Any usage beyond the permitted purposes may require explicit permission from the data provider.

> - Attribution to the original data source is required. Please refer to the appropriate acknowledgments section for details.

>

> Please ensure that you review and comply with Kaggle's terms and conditions and any additional licensing or usage restrictions specified by the data provider.

>

> For more information about the dataset and how to access it, please refer to the [Kaggle Terms page](<https://www.kaggle.com/terms>).

## License

> This project is licensed under the [MIT License](LICENSE). Feel free to modify and adapt the code according to your needs.

## Skills

---

### languages

Python • SQL • C++ • YAML • JavaScript

### technologies

Microsoft Power BI • Google Cloud Platform • Tableau • Git • Django • Flask • PyTorch • PostgreSQL

### machine\_learning

Deep Learning • Computer Vision • Natural Language Processing (NLP) • MLOps • Statistics • Testing

### soft\_skills

Communication • Teamwork • Leadership • Research • Group Discussion • Problem-Solving • Time Management

## Certifications

---

**1. Data Science Professional Certificate**

**2. Associate Data Analyst Certification**

**3. Introduction to Statistics for Data Science**

**4. SQL Associate Certification**

**5. Data Engineer Associate Certification**

**6. Introduction to Web Development**

**7. Introduction to the Internet of Things and Embedded Systems**

**8. National Intellectual Property Awareness Mission**

### Data Science Professional Certificate

Institution: DataCamp

Duration: July 2023 - July 2025

Credential ID: DS0022970321476

Certificate Link: <https://www.datacamp.com/certificate/DS0022970321476>

### Associate Data Analyst Certification

Institution: DataCamp  
Duration: June 2024 - June 2026  
Credential ID: DAA0014613937359  
Certificate Link: <https://www.datacamp.com/certificate/DAA0014613937359>

## **Introduction to Statistics for Data Science**

Institution: Coursera - Stanford University  
Duration: July 2023  
Credential ID: YR3SX5W7HYDJ  
Certificate Link: <https://www.coursera.org/account/accomplishments/certificate/YR3SX5W7HYDJ>

## **SQL Associate Certification**

Institution: DataCamp  
Duration: July 2024 - July 2026  
Credential ID: SQA0014151700900  
Certificate Link: <https://www.datacamp.com/certificate/SQA0014151700900>

## **Data Engineer Associate Certification**

Institution: DataCamp  
Duration: June 2023 - June 2025  
Credential ID: DEA0017111704609  
Certificate Link: <https://www.datacamp.com/certificate/DEA0017111704609>

## **Introduction to Web Development**

Institution: Coursera - University of California, Davis  
Duration: April 2022  
Credential ID: 5DU9YH99ESNZ  
Certificate Link: <https://www.coursera.org/account/accomplishments/certificate/5DU9YH99ESNZ>

## **Introduction to the Internet of Things and Embedded Systems**

Institution: Coursera - University of California, Irvine  
Duration: March 2022  
Credential ID: RXB2NMHEVCET  
Certificate Link: <https://www.coursera.org/account/accomplishments/certificate/RXB2NMHEVCET>

## **National Intellectual Property Awareness Mission**

Institution: Government of India  
Duration: September 2022  
Credential ID:  
Certificate Link: [https://iprsearch.ipindia.gov.in/RQStatus/IPR\\_EVENT.aspx?FileName=QmF0Y2g0L01yLIZJUkFMMTI5MTY5MC5](https://iprsearch.ipindia.gov.in/RQStatus/IPR_EVENT.aspx?FileName=QmF0Y2g0L01yLIZJUkFMMTI5MTY5MC5)

## **Achievements**

---

Ranked 300th out of 975 participants in Kaggle BirdClef 2024, with a ROC value of 62.62%

Maintained 500+ daily problem streak on LeetCode and achieved a 3\* rating on CodeChef with a max rating of 1622

Earned 350,000+ XP, completed 7 projects, and finished 14 tracks on DataCamp with top performance