1. Introduction

1.1.Project description

- **Imagecap.ai is a powerful flutter application with powerfull state management system GetX with attractive and advanced levele designed concept to generate captions for your images effortlessly with an intuitive interface and seamless functionality.**
- **This app allows user to upload their image and receive creative captions generate using cutting-edge AI algorithms.**
- **Whether you're a social media enthusiast, a content creator , or simple looking to add some flair to your photos.**
- **Explore its features, unleash your creativity , and watch as your images come to life with engaging captions!**
- **In this application we no need to any admin panel just because of all the data (captions) are generated automatically through the python API.**
- **User can download the app from Official Website**

❖ User Panel :-

- **It is mandatory to signup and login user , Authentication feature implemented by API (Application Programming Interface) which is created in PHP.**
- **User can change him/her password whenever him/her forgot the password.**
- **User upload their pictures only supported JPEG , JPG , PNG and WEBP files.**
- **User can read creative captions of their pictures and also allow to copy the captions into the clipboard**
- **User can not able to tack the screen short of the captions or screen.**
- **User no need to logout in this app user will automatically logout when the session is over.**

1.2 Project Profile :-

Imagecap.ai	
Project Title :	Imagecap.ai
Fronted :	Flutter
Backend :	PHP and Python
Run :	Any Android Mobile Device & IOS
Platform :	Android Studio
Documentation Tool :	Microsoft Word
Internal Guide	Ms. Aarti Jariwala

2. Environment Description

2.1.Hardware and Software Requirements

- **Hardware Requirement**

  1. **Development Machine :**

     - **Processor : Intel Core i5 or higher (or equivalent) .**
     - **RAM : 8 GB minimum (16 GB or more recommended for better performance)**
     - **Storage: At least 10 GB of free storage (for Flutter SDK, dependencies, Python, and data storage) .**
     - **Graphics : Integrated GPU (Dedicated GPU is recommended for machine learning tasks but not necessary for general development) .**

  2. **Mobile Device (for testing) :**

     - **Operating System : Android (version 8.0 or higher) / iOS (version 11.0 or higher) .**
     - **RAM : 2 GB or more .**
     - **Storage: Minimum 500 MB free space for app installation .**
     - **Processor : ARM-based processors for smooth app operation .**

- **Software Requirements :**

  1. **Operating System :**

     - **Windows 10 / macOS / Linux (Development machine)**

2. **Development Environment :**

- **Flutter SDK : Version 3.0 or higher (for building the app) .**
- **Dart SDK : Included with Flutter SDK .**
- **Android Studio or VS Code : Preferred Integrated Development Environment (IDE) for Flutter development .**
  - ✓ **Android Studio Version : Arctic Fox or newer .**
  - ✓ **VS Code : Latest version with Flutter and Dart extensions .**

3. **Mobile Platform SDKs :**

- **Android SDK: (For Android development and testing) .**
- **Xcode : (For iOS development, required on macOS) .**

4. **Backend Services :**

- **PHP Server : To handle API request :**
  - ✓ **PHP Version : 7.4 or higher .**
  - ✓ **Apache : For local development of PHP APIs .**
  - ✓ **MySQL Database : For storing user information, images, and other metadata .**
- **Python Backend: For image caption generation :**
  - ✓ **Python Version : 3.7 or higher .**
  - ✓ **Flask Framework : For API creation and handling HTTP requests .**
  - ✓ **Torch & Transformers Library : Required for deep learning-based image caption generation .**

5. **Mobile Build Requirements :**

- **Android Emulator / iOS Simulator : For testing the app in various configurations .**
- **Flutter Doctor : Ensure that all Flutter and environment dependencies are properly installed .**

6. **Other Dependencies :**

- **Image Picker Library (Flutter) : For selecting images from the gallery or capturing them via the camera .**
- **HTTP Package (Flutter) : For making network requests to the PHP and Python backends .**
- **GetX Package (Flutter) : For state management and navigation .**
- **Awesome Dialog (Flutter) : For enhanced UI dialogs (used for error messages) .**
- **PIL (Python) : Python Imaging Library for processing images .**
- **Torch & Transformers (Python) : To implement image captioning using the Vision Transformer .**

7. **Other Tools :**

- **Postman : To test the API endpoints during development .**
- **Ngrok (optional) : For exposing your local development server to the internet for testing purposes .**

2.2.Technologies Used

1. **Flutter Technology**

   I. **Cross-Platform Development** :

   - **Flutter allows you to develop the Application for both Android and iOS platforms with a single codebase, saving time and resources.**

   II. **Rich Widgets Library** :

   - **Utilized Flutter's extensive widget library to create a highly responsive and visually appealing user interface, ensuring a smooth user experience.**

   III. **State Management with GetX** :

   - **Employed GetX for efficient state management, ensuring seamless user interactions and maintaining the app's performance across different screens and functionalities.**

   IV. **Asynchronous Programming with Dart** :

   - **Implemented Dart's asynchronous programming features to handle image uploads, API requests, and data processing efficiently, providing a smooth user experience.**

   V. **HTTP Requests** :

   - **Used the http package in Flutter to make API calls for image caption generation, ensuring real-time communication with the backend server.**

VI. **Animation & Customization** :

- **Leveraged Flutter's animation capabilities to create smooth transitions and engaging visual effects, enhancing the app's interactivity and user engagement.**

VII. **Responsive Design** :

- **Ensured that the app adapts seamlessly to different screen sizes and orientations, providing a consistent experience across all devices.**

VIII. **Image Handling** :

- **Integrated image picker and storage solutions to allow users to upload images from their device gallery, which are then processed to generate captions.**

IX. **Dependency Injection with GetX** :

- **Simplified dependency management and improved code organization using GetX's dependency injection feature, making the app more modular and maintainable.**

X. **Secure Data Storage** :

- **Used Flutter's secure storage and SharedPreferences to store user data locally, ensuring data persistence and security.**

XI. **Custom Dialogs and Error Handling** :

- **Implemented custom dialogs using Flutter and AwesomeDialog to provide users with meaningful feedback and error handling, improving the app's usability.**

XII. Flutter's Hot Reload :

- Utilized Flutter's hot reload feature to quickly iterate and refine the app's features, leading to a faster development cycle and more robust application .

2. PHP Technology

I. API Integration :

- The app uses PHP to handle backend logic, connecting the Flutter frontend with the server-side functionalities .
- PHP APIs facilitate user authentication, image processing, and caption generation.

II. Data Handling :

- PHP handles incoming HTTP requests, processes data, and returns responses to the Flutter app .
- It securely manages user sessions and data storage.

III. Server-Side Scripting :

- PHP is utilized for scripting server-side logic, such as processing images, generating captions, and storing or retrieving data.

IV. Security Measures :

- PHP includes built-in security features like input validation and session management, crucial for protecting user data and ensuring secure communication between the app and server .

V. **Integration with Databases** :

- **PHP interacts with databases to store user information, login sessions, and ensuring seamless data management.**

3. Python Technology

I. **Versatility and Flexibility** :

- **Python is a highly versatile language, suitable for a wide range of applications, from web development to data analysis and machine learning. Its flexibility allows developers to choose the best tools and libraries for their specific needs.**

II. **Rich Ecosystem of Libraries** :

- **Python boasts a vast ecosystem of libraries and frameworks, such as Flask for web development, TensorFlow and PyTorch for machine learning, and PIL for image processing, making it easier to implement complex functionalities with less effort.**

III. **Ease of Learning and Use** :

- **Python's simple and readable syntax makes it accessible to both beginners and experienced developers. This ease of use accelerates the development process and reduces the likelihood of errors**

IV. **Strong Community Support** :

- **Python has a large and active community that contributes to a wealth of resources, tutorials, and documentation. This strong community support ensures that developers can quickly find solutions to any challenges they encounter.**

V. **Cross-Platform Compatibility** :

- **Python is a cross-platform language, meaning that code written in Python can run on various operating systems, such as Windows, macOS, and Linux, without modification.**

VI. **Integration Capabilities** :

- **Python can easily integrate with other languages and technologies, including Dart (Flutter), C/C++, Java, and .NET, as well as with databases, APIs, and web services, making it a powerful tool for building complex, multi-layered applications.**

VII. **Rapid Prototyping** :

- **Python's simplicity and the availability of high-level libraries allow developers to quickly prototype and iterate on ideas, speeding up the development of new features and innovations.**

VIII. **Scalability :**

- **Python's scalability ensures that applications can grow in complexity and user base without requiring a complete overhaul of the underlying code, making it ideal for both small projects and large-scale applications.**

3. System Analysis

3.1. Existing System and its Drawbacks

❖ **Existing Systems :**

1. **Manual Image Captioning :**

   - **Most systems rely on users to manually add captions to their images. This is often time-consuming and prone to human error, leading to inconsistent and less informative captions.**

2. **Basic Photo Tagging Applications :**

   - **Some photo apps offer basic tagging functionality where users can manually label objects or events in photos. However, this requires prior knowledge of the content and lacks intelligent automation.**

3. **Limited AI-Powered Captioning Tools :**

   - **While a few AI-powered captioning tools exist, they are typically costly, not user-friendly, or lack integration with mobile platforms. These solutions are often designed for niche use cases, such as accessibility for the visually impaired or content analysis in the marketing sector.**

❖ Drawbacks of the Existing System :

1. **Manual Effort and Time Consumption :**

   - **Users are required to input captions manually, which is not only time-consuming but can also lead to incomplete or inaccurate descriptions. This makes organizing and retrieving images based on descriptions difficult.**

2. **Lack of Automation and Intelligence :**

   - **Most existing systems lack automation, and even the few AI-based tools have limited accuracy in generating contextual captions. These systems often fail to recognize the nuances of image content, such as emotions, actions, or specific objects within the image.**

3. **Limited Accessibility :**

   - **Existing image captioning tools, especially those that leverage AI, are often inaccessible to regular users due to high costs, technical complexity, or platform limitations. Many require integration with specialized software, making them unsuitable for everyday use.**

4. **Generic Descriptions :**

   - **AI-powered captioning systems tend to generate generic or overly simple descriptions (e.g., "a dog" or "a tree") that may not capture the complexity or context of an image. This reduces the usefulness of such captions, especially for professional use cases like marketing or content creation.**

5. **Poor User Experience :**

- **The user interfaces of many existing apps and tools are often unintuitive, making the image captioning process difficult for non-technical users. This further limits the widespread adoption of such solutions.**

6. **Data Privacy Concerns :**

- **Some systems that rely on cloud-based AI services raise concerns about data privacy, as users must upload their images to third-party servers, potentially exposing sensitive or personal content.**

3.2.Expected Advantages

1. **Automated Caption Generation :**

- **The app uses AI-powered caption generation, significantly reducing manual effort and saving time for users. Instead of typing captions, users can instantly generate multiple, contextually relevant captions for any image uploaded.**

2. **Enhanced Accuracy with AI :**

- **By utilizing advanced machine learning models (such as the VisionEncoderDecoderModel), the app generates more accurate and descriptive captions compared to traditional manual tagging or existing AI tools. It can identify objects, actions, and context, creating more meaningful captions.**

3. **User-Friendly Interface :**

   - **The app features an intuitive and clean interface, making it easy for users to upload images and receive captions. Its simplicity makes it accessible for users with varying levels of technical expertise, enhancing the overall user experience.**

4. **Real-Time Caption Generation:**

   - **Captions are generated almost instantly after image upload, offering a fast and seamless process for users. This real-time functionality is particularly advantageous for professionals who require quick and accurate image descriptions, such as content creators or marketers.**

5. **Cross-Platform Accessibility :**

   - **The app can be easily adapted for multiple platforms like iOS and Android, providing a consistent experience for users across different devices.**

4. Proposed System

4.1. Scope

1. **User-Friendly Interface :**

   - **The app is designed to offer a seamless, user-friendly experience, making it accessible to users of all technical backgrounds. This includes a simple image upload process, real-time caption generation, and easy navigation.**

2. **Automated Image Captioning :**

   - **The core functionality of the app is to automatically generate captions for uploaded images using advanced AI models, which significantly enhances productivity for users, including content creators, social media managers, and photographers.**

3. **Cross-Platform Availability :**

   - **The app can be developed for both mobile (iOS, Android) and web platforms, enabling users to access it from various devices. This cross-platform availability ensures that the app reaches a wide audience.**

4. **AI-Driven Accuracy :**

   - **Using machine learning models like VisionEncoderDecoderModel and ViT-based feature extraction, the app can provide accurate and contextually rich captions. This feature enhances its usefulness in a range of industries, from marketing to journalism.**

5. **Real-Time Performance :**

   - **The app is built to deliver captions in real-time, making it highly efficient for users who need quick turnaround times, such as during live events, product launches, or fast-paced social media campaigns.**

6. **Future Expansion:**

   - **The app's AI model can be continuously improved with more training data, and features like voice-to-text captioning, image filters, or advanced image editing tools can be added in future versions.**

4.2. Project Modules

1. **User Interface (UI) Module** :

   - **A Flutter-based user interface where users can upload images, view captions, and interact with the system.**
   - **Clean and simple design for easy navigation and interaction.**
   - **Includes the option to view the generated captions in real-time.**

2. **User Authentication Module** :

   I. **Signup :**

      - **New users can register using the signup API.**
      - **Includes email validation and password storage using PHP backend.**

II. Login :

- **New users can register using the signup API.**
- **Includes email validation and password storage using PHP backend.**

III. OTP Verification Module :

- **Sends OTP to the registered email via gmail for verification when the user requests a login verification.**
- **Verifies OTP entered by the user and allows access to confirm login.**

IV. Forgot Password :

- **Allows users to reset their password via email.**
- **When email existing in database after that user can set new password.**

3. Home Module :

- **Once use can successfully login then open home screen in this screen user can upload image.**
- **Imgae can only support JPEG , JPG , PNG and WEBP files.**

4. Image Upload and Caption Generation Module :

- **Users can upload images from their device.**
- **Images are sent to the backend (Python API) for processing.**
- **Uses a machine learning code hosted in a Python API to generate captions for uploaded images.**
- **Displays generated captions on the app.**

5. **Session Management Module :**
   - **Uses session to maintain a persistent login state across app restarts.**
   - **User no need to logout , In this app user automatically logout when user's session is expire**
   - **Once user's session is expire then user redirected to the login page throgh the attractive alert dialogue for the re-login**

4.3. Objectives / Functionalities

1. **User Authentication** :

   - **Users can sign up by providing their email and password.**
   - **The system ensures secure login for returning users.**
   - **Implements OTP-based verification for login security.**
   - **Provides a "Forgot Password" feature, allowing users to reset their passwords securely.**

2. **Image Uploading** :

   - **Users can upload images directly from their device to the app.**
   - **The app supports image selection and preview before uploading.**

3. **AI-Powered Caption Generation** :

   - **Uses a Python-based API to generate captions for uploaded images using machine learning models.**
   - **Generates multiple unique captions for a single image.**

4. **Secure Data Management** :

- **Utilizes secure password hashing for user credentials.**
- **Data is managed efficiently, ensuring privacy and security for sensitive information such as passwords and email.**

5. **User-Friendly Interface** :

- **Simple and attractive UI, allowing users to easily navigate through login, signup, image uploading, and viewing captions.**

6. **Error Handling and Notifications** :

- **Alerts users about any issues, such as wrong credentials, or failed image uploads, using responsive dialogs and notifications.**

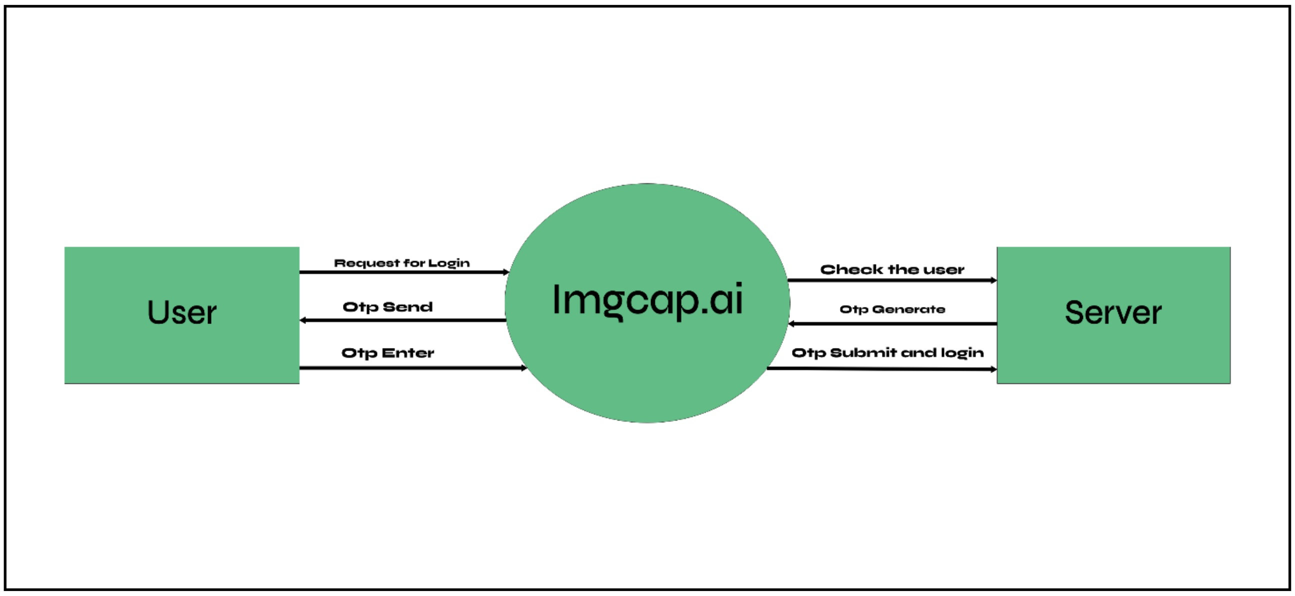
7. **Session Management** :

- **Ensures persistent login sessions, allowing users to stay logged in without having to re-enter credentials frequently.**

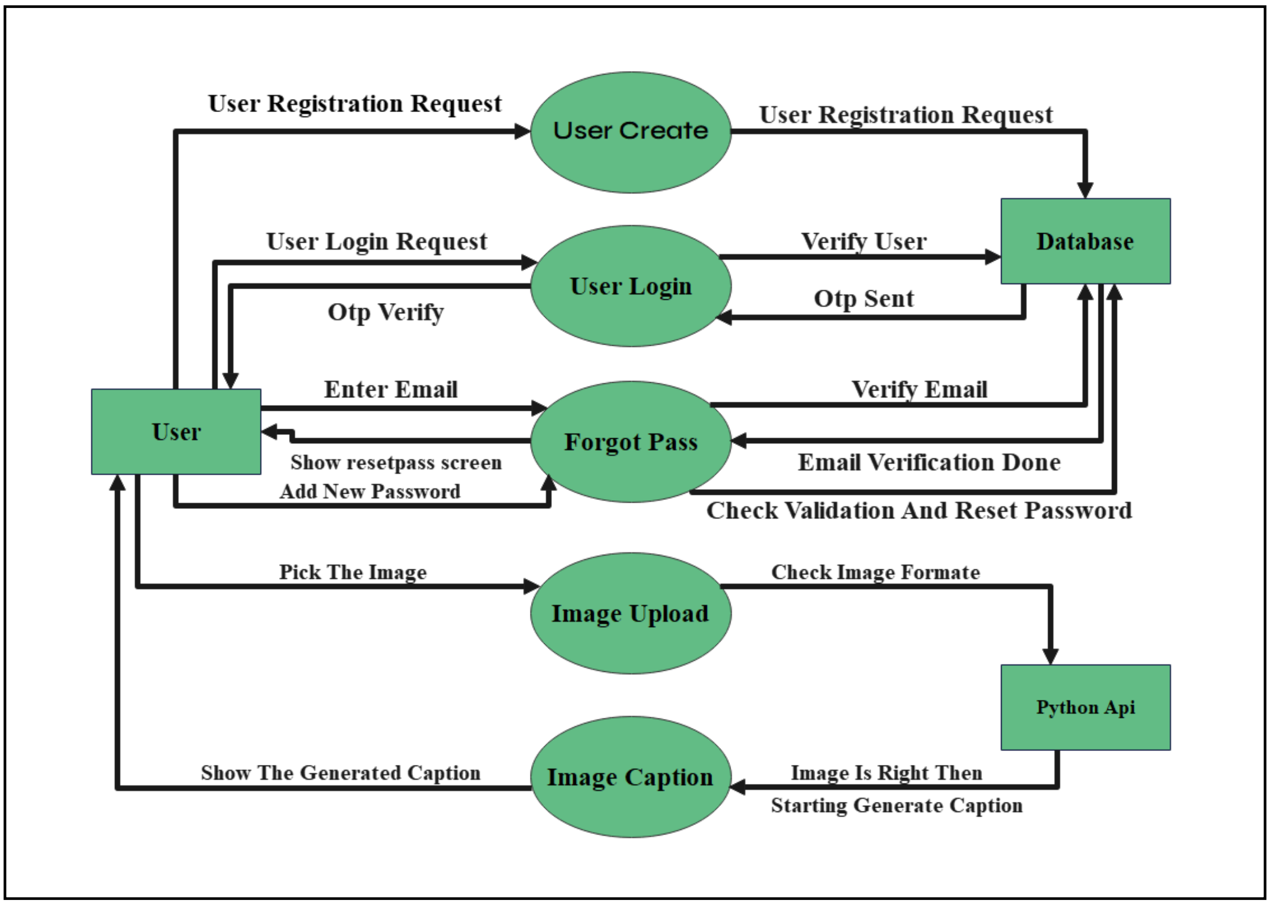
5. Detail Planning

5.1. Data Flow Diagram / UML

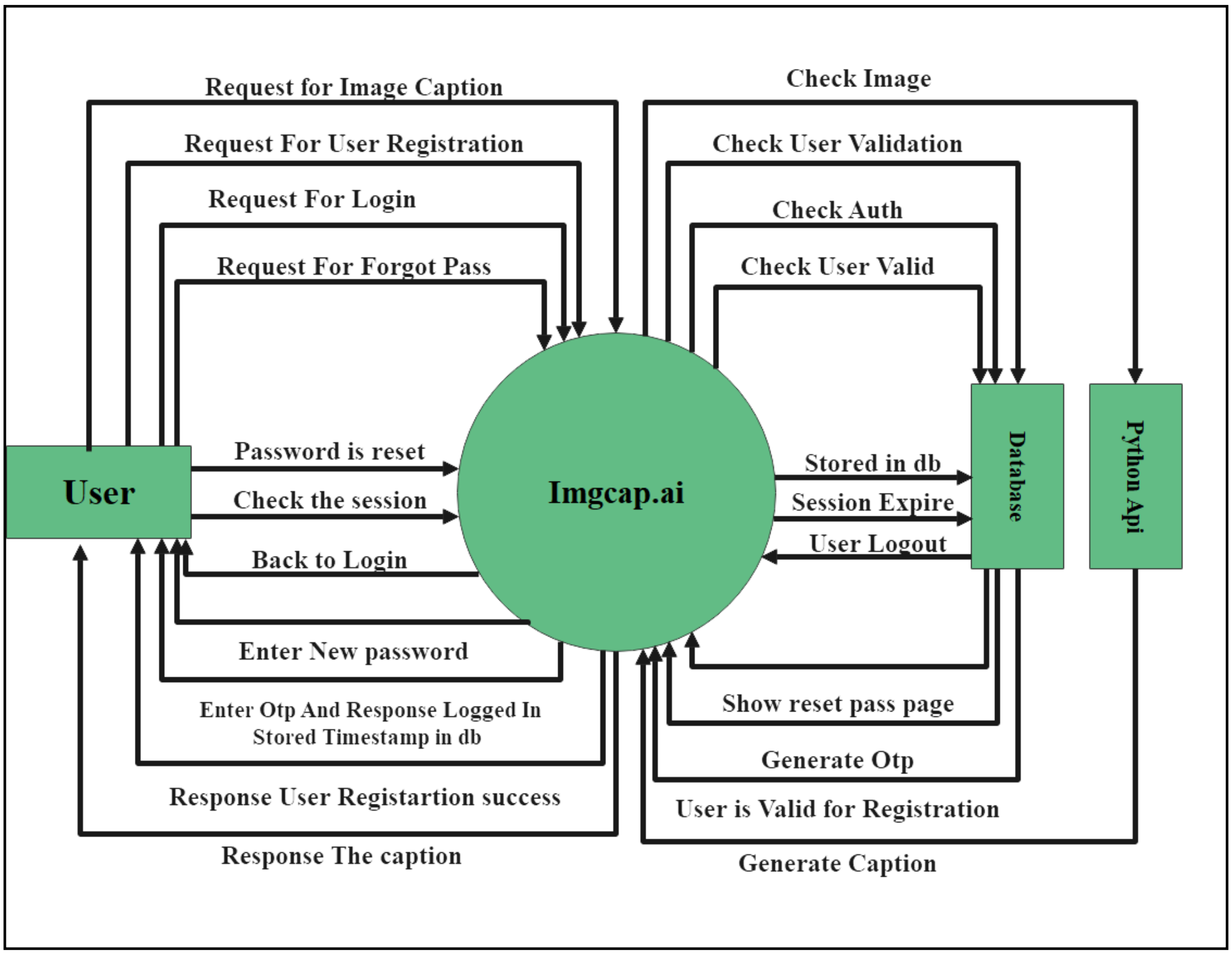
I. Zero – Level DFD :



II. First – Level DFD :



III. Secound – Level DFD :



5.2. Data Dictionary

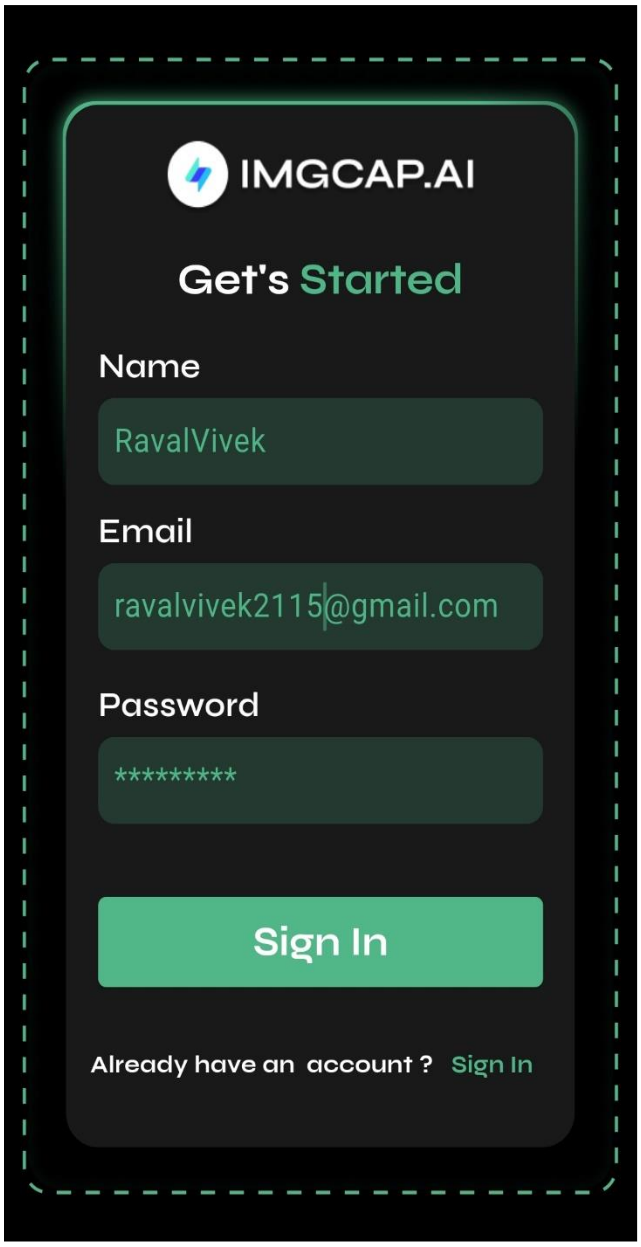
User Table				
Field Name	**Data Type**	**Field Length**	**Constraint**	**Description**
Id	Int	250	Primary Key	Id of user
Name	Varchar	250	Not null	Name of user
Email	Varchar	250	Not null	Email of user
Password	Varchar	250	Not null	User Password
Otp	Int	6	Not null	Generated Otp
Otp_Generated_at	timestamp	Timestamp	Not null	Login timestamp

6. System Design

6.1. Input Design

1. **Splash Screen :**

2. **Signup and Login Screen :**


IMGCAP.AI

Get's Started

Name

RavalVivek

Email

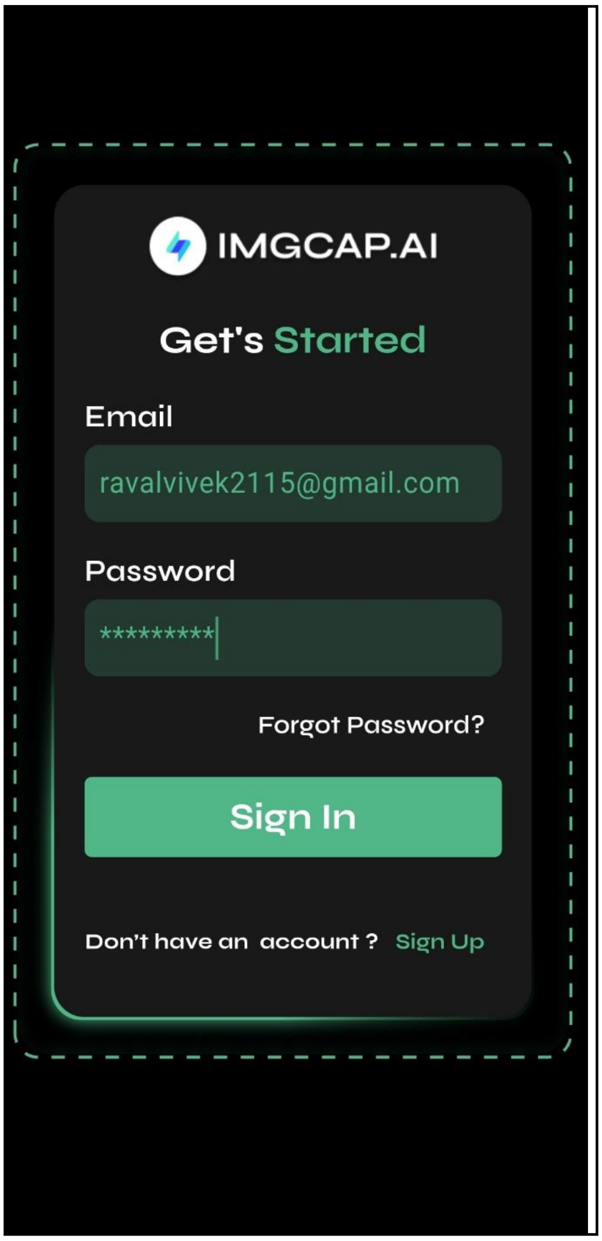
ravalvivek2115@gmail.com

Password

****

Sign In

Already have an account ? Sign In


IMGCAP.AI

Get's Started

Email

ravalvivek2115@gmail.com

Password

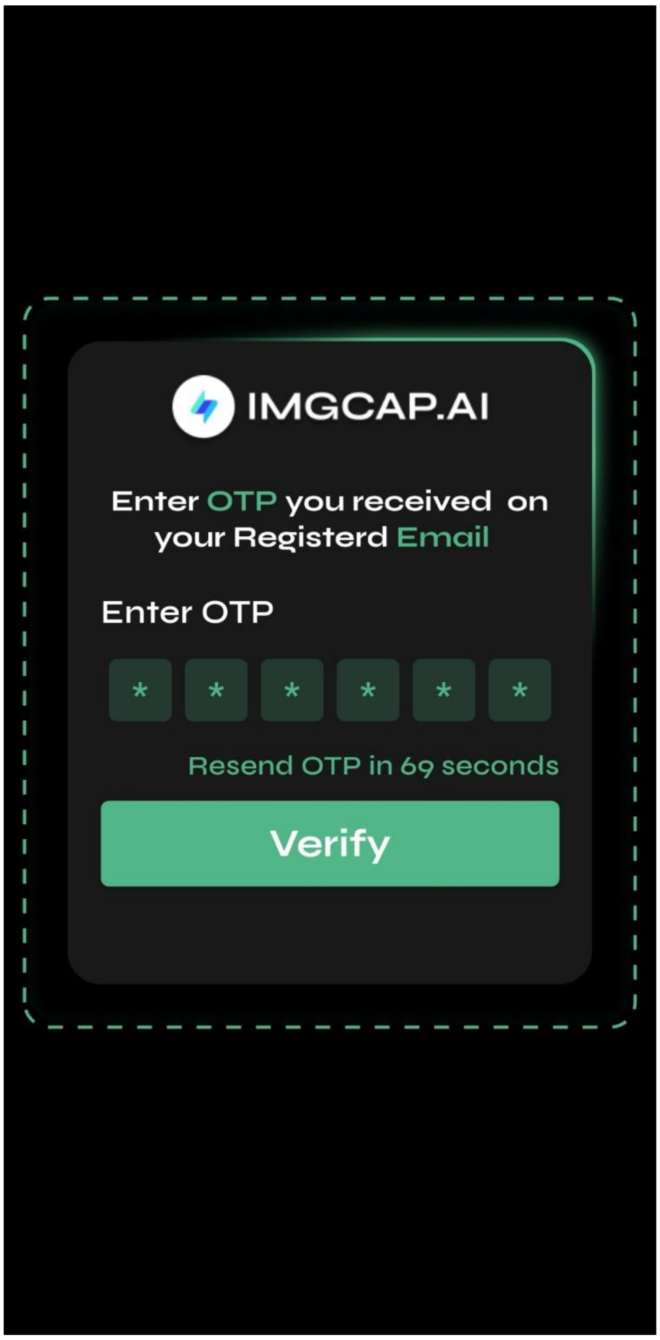
****

Forgot Password?

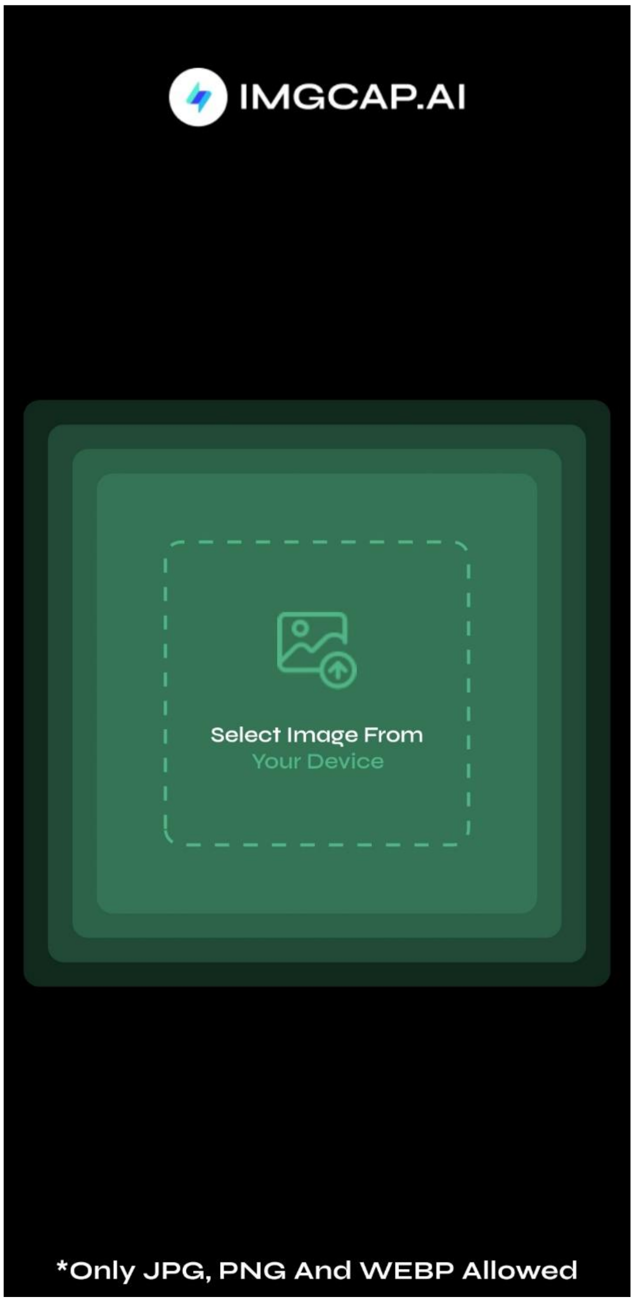
Sign In

Don't have an account ? Sign Up

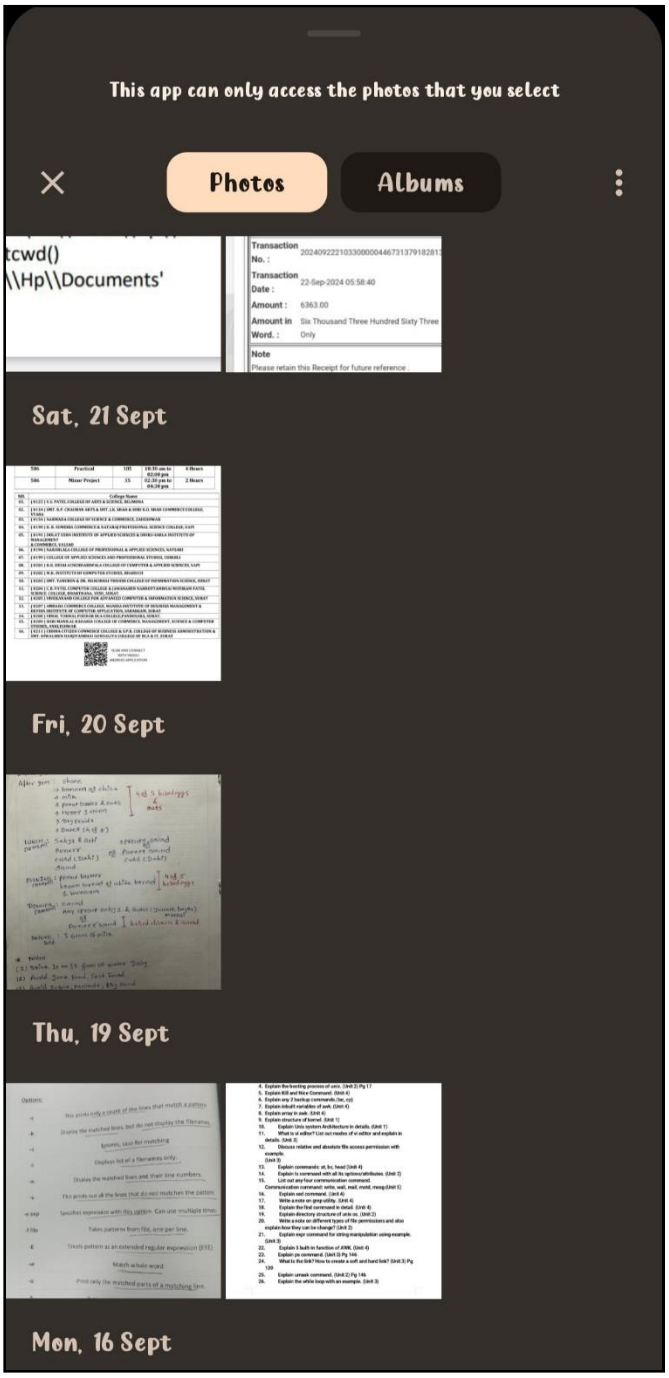
3. **Otp Verification Screen :**



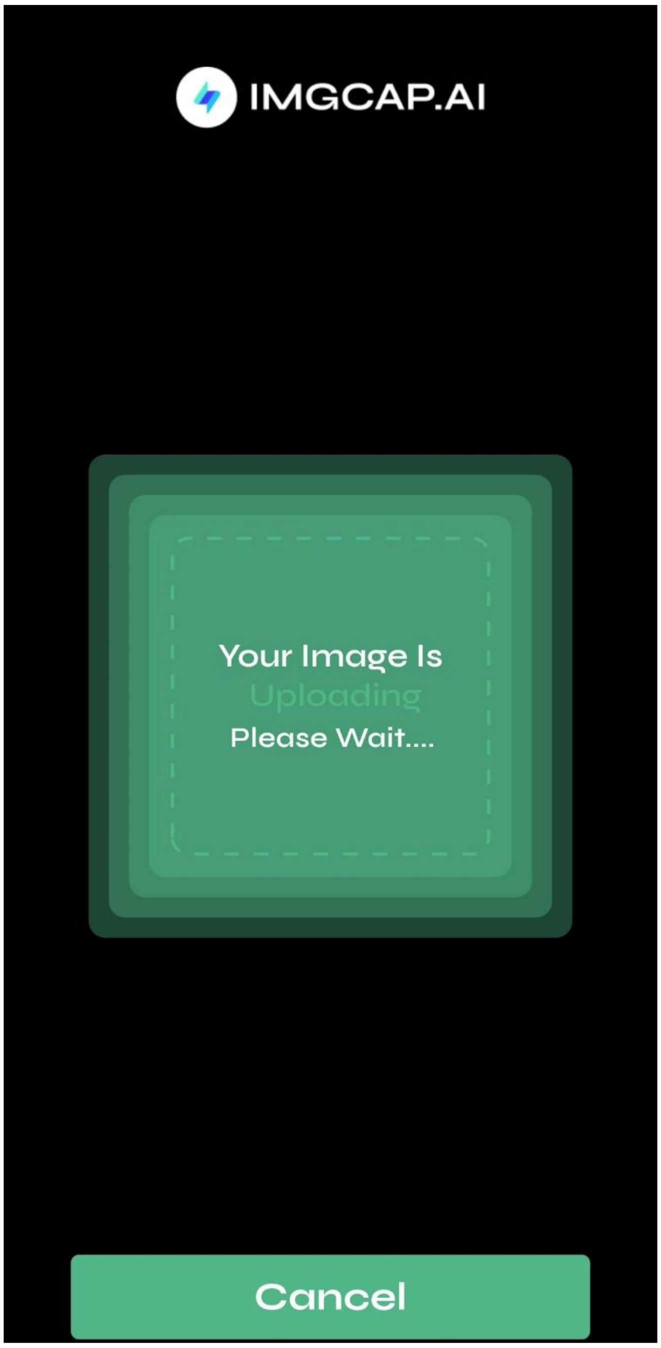
4. **Home Screen / Image Upload Screen:**



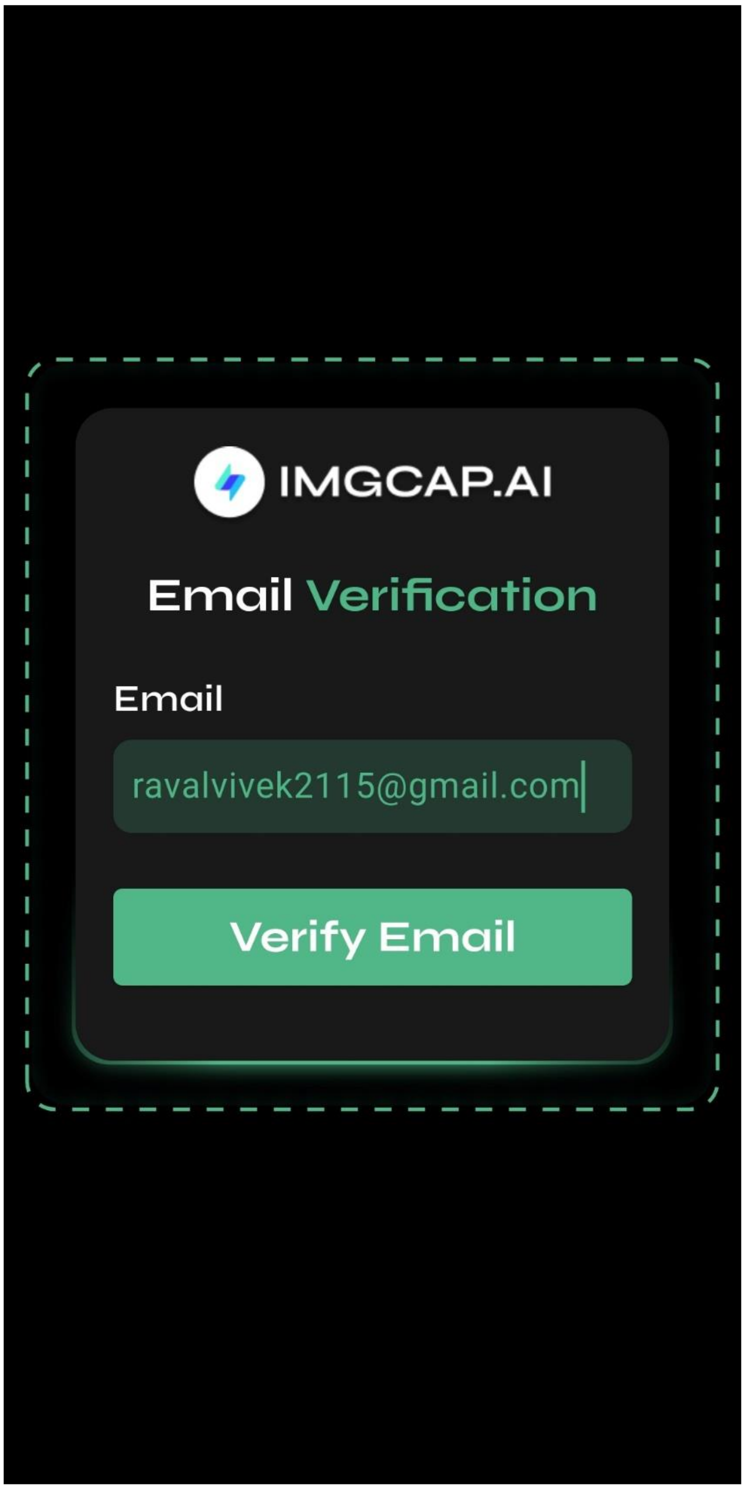
5. **Image Picker screen :**



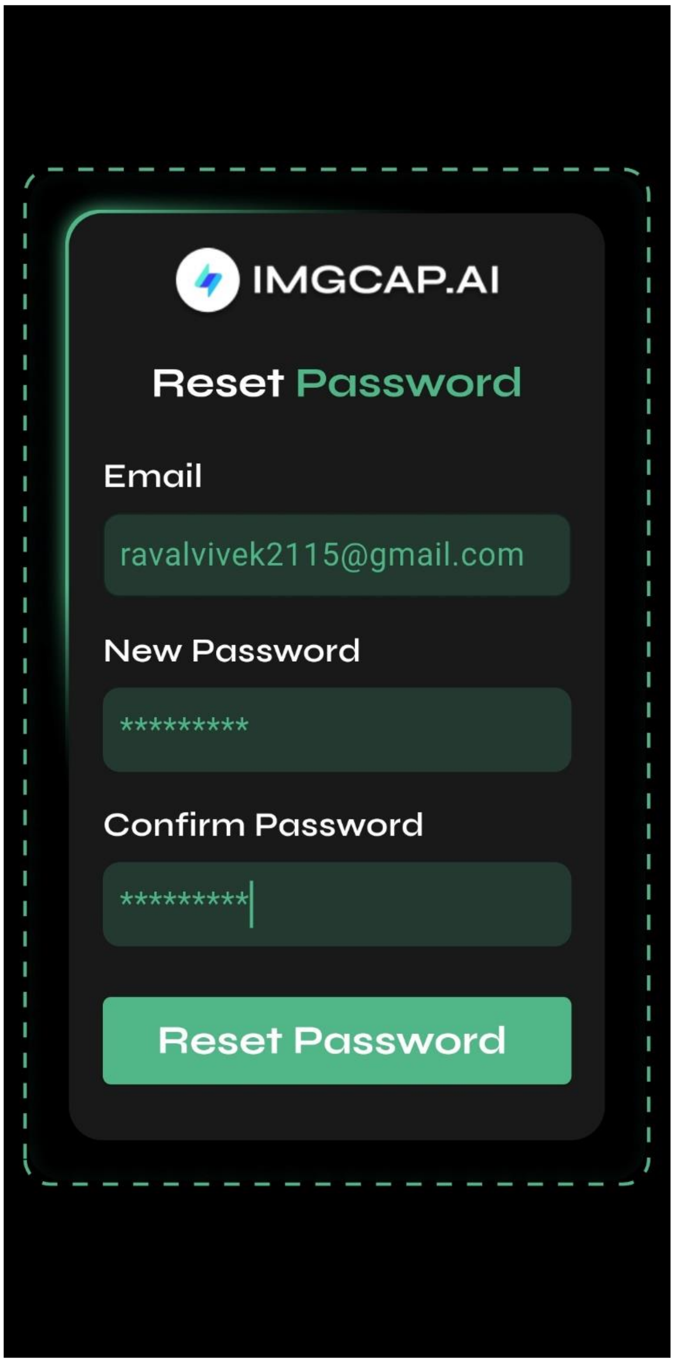
6. **Image Uploading Progress Screen :**



7. **Email Verify Screen For Forgot Password :**



8. Set New Password Screen :



Email

ravalvivek2115@gmail.com

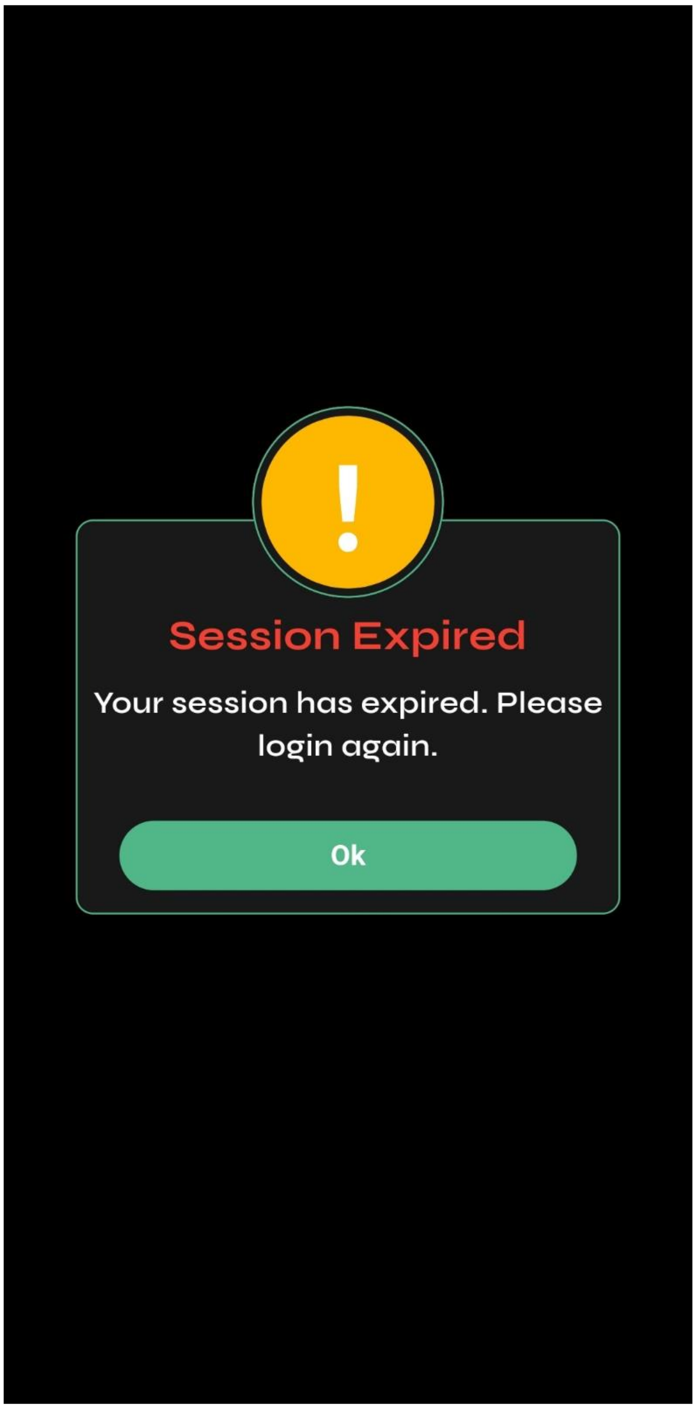
New Password

****

Confirm Password

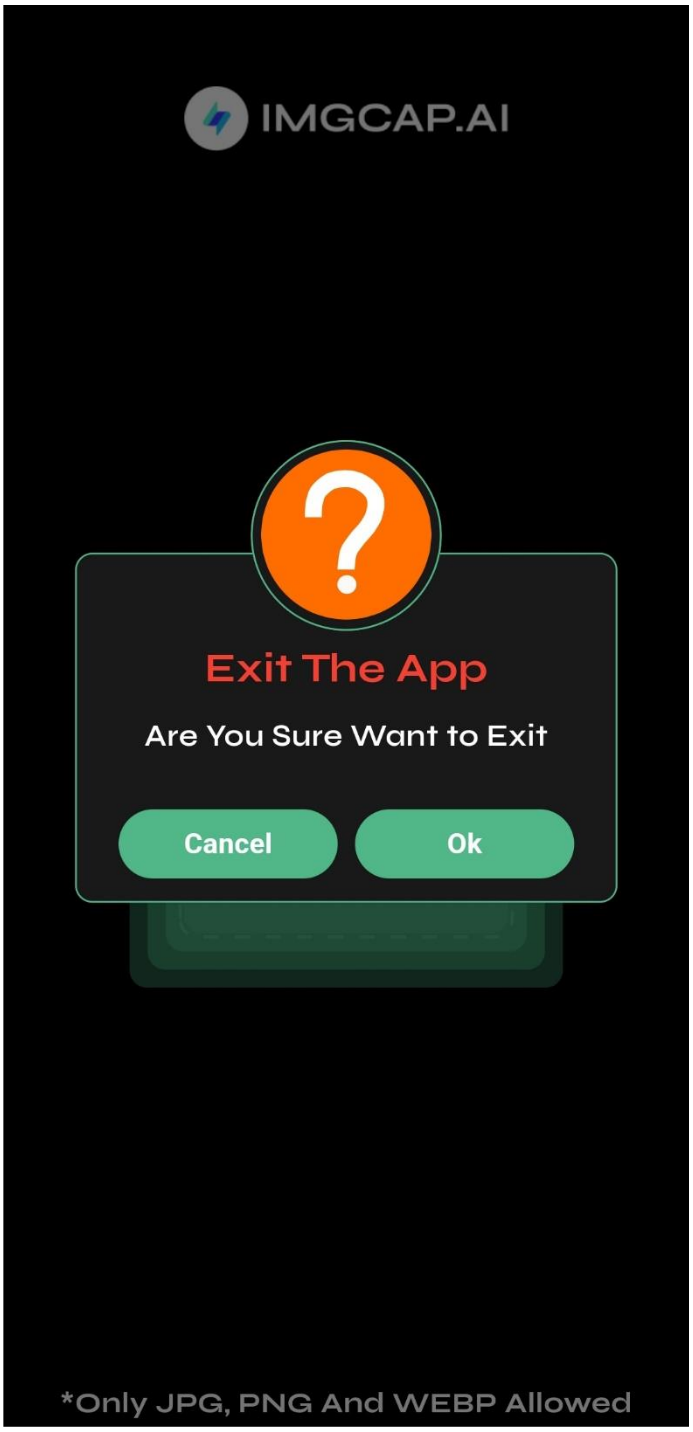
****

9. **Session Expire Screen :**


Session Expired

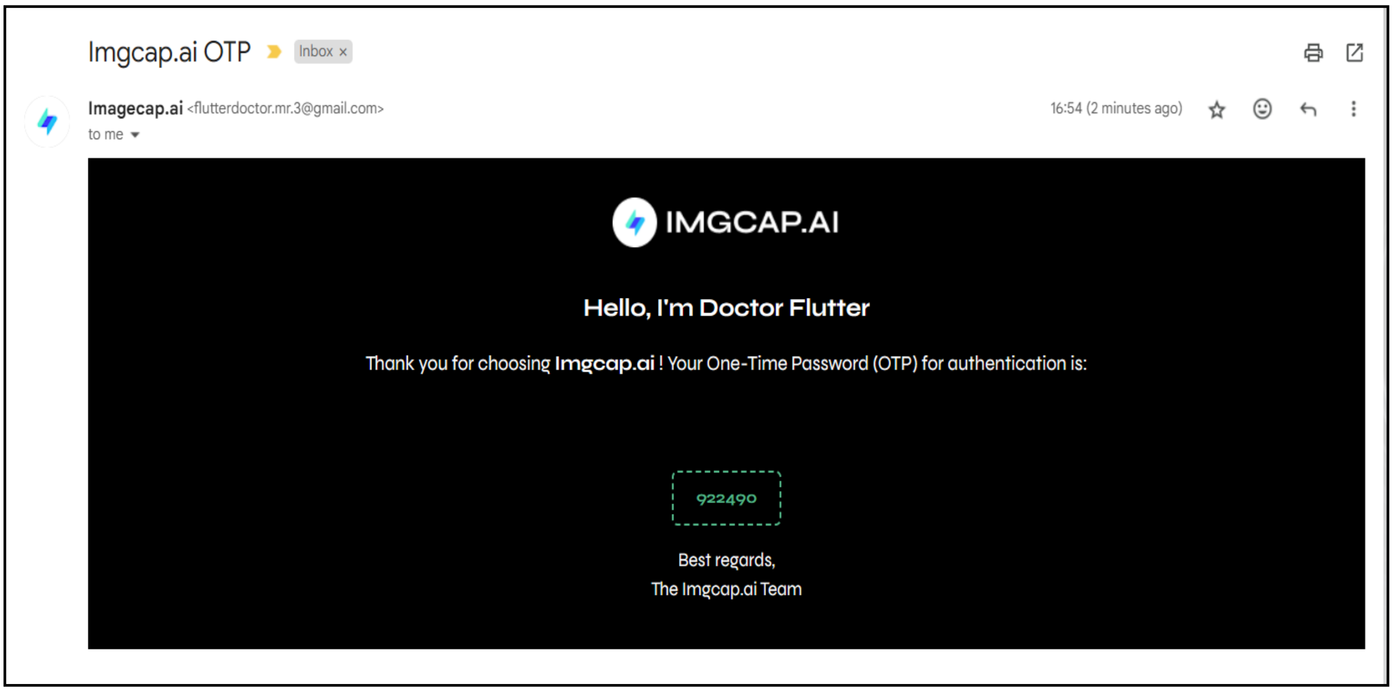
Your session has expired. Please login again.

10. Exit The App Screen :



6.2. Output Design

1. **Otp Recive Screen :**



Imgcap.ai OTP Inbox ×

Imagecap.ai <flutterdoctor.mr.3@gmail.com>
to me

16:54 (2 minutes ago)

IMGCAP.AI

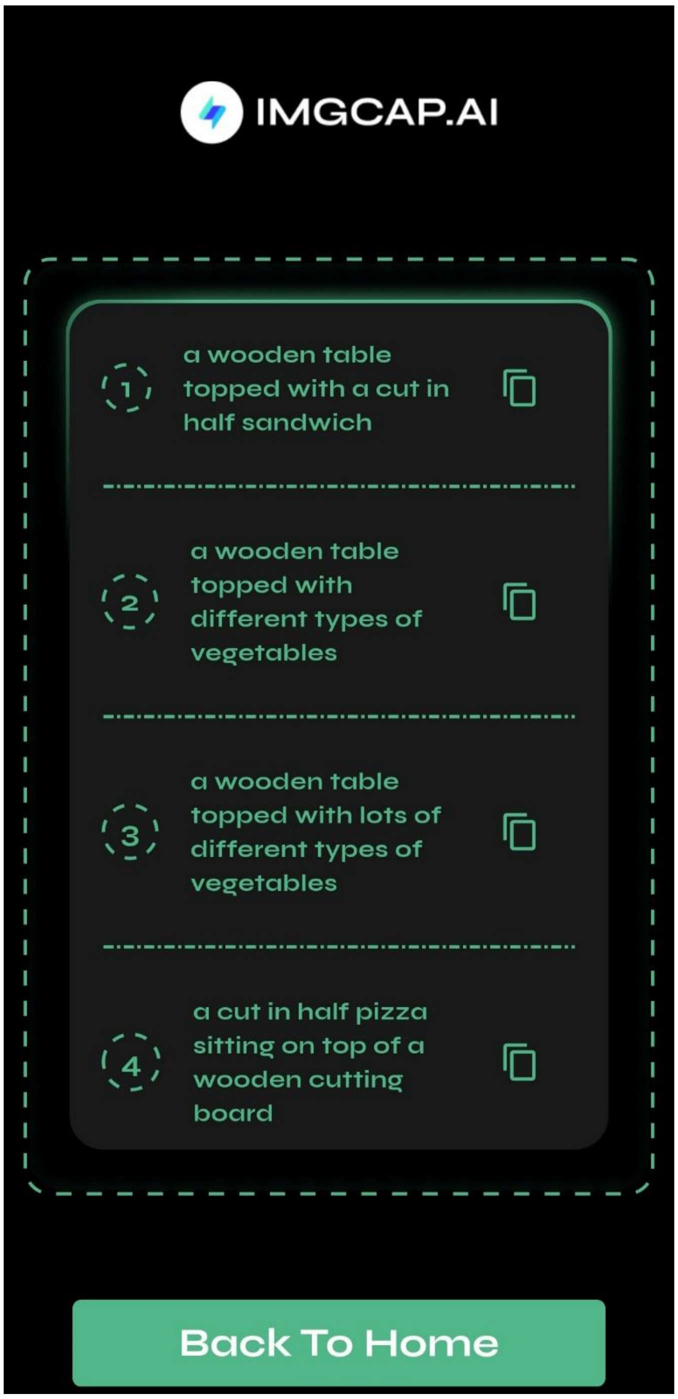
Hello, I'm Doctor Flutter

Thank you for choosing **Imgcap.ai** ! Your One-Time Password (OTP) for authentication is:

922490

Best regards,
The Imgcap.ai Team

2. **Generate Caption Screen :**


IMGCAP.AI

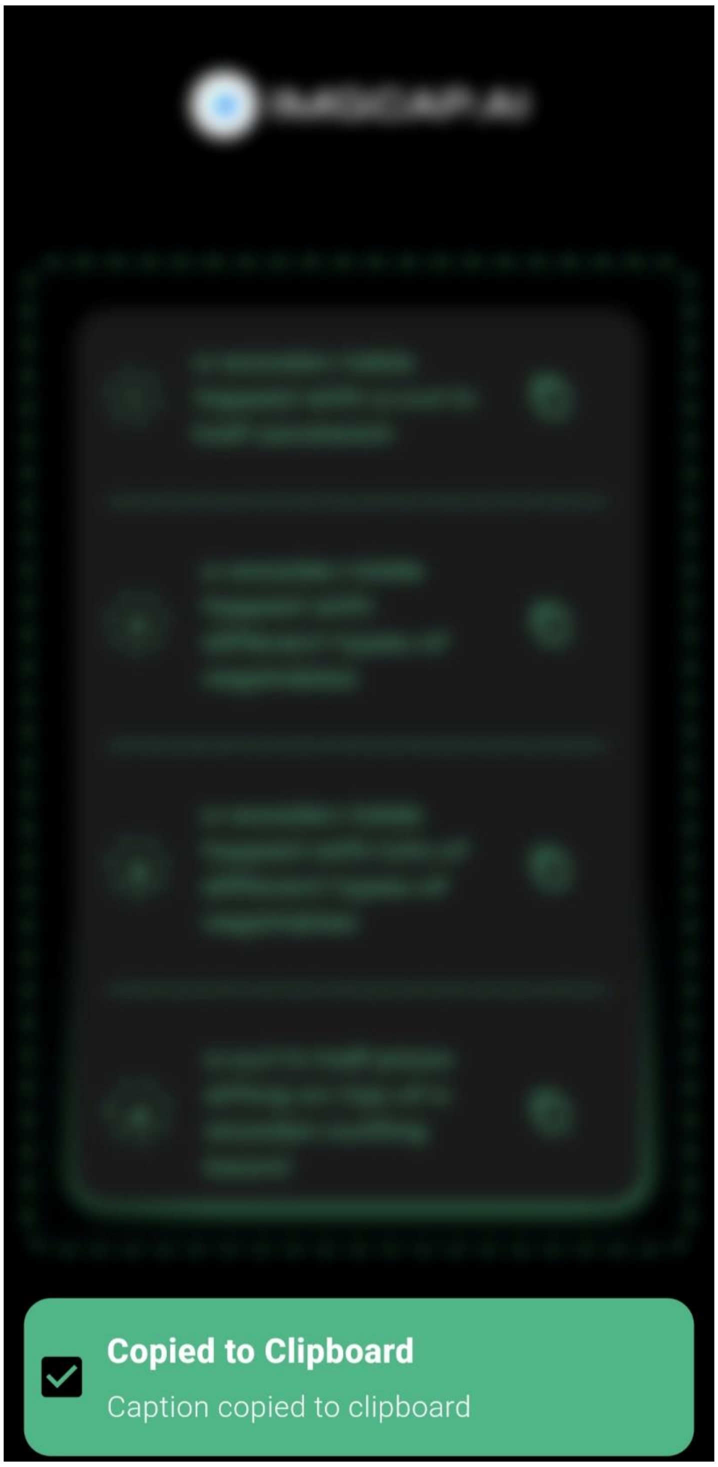
1 a wooden table topped with a cut in half sandwich

2 a wooden table topped with different types of vegetables

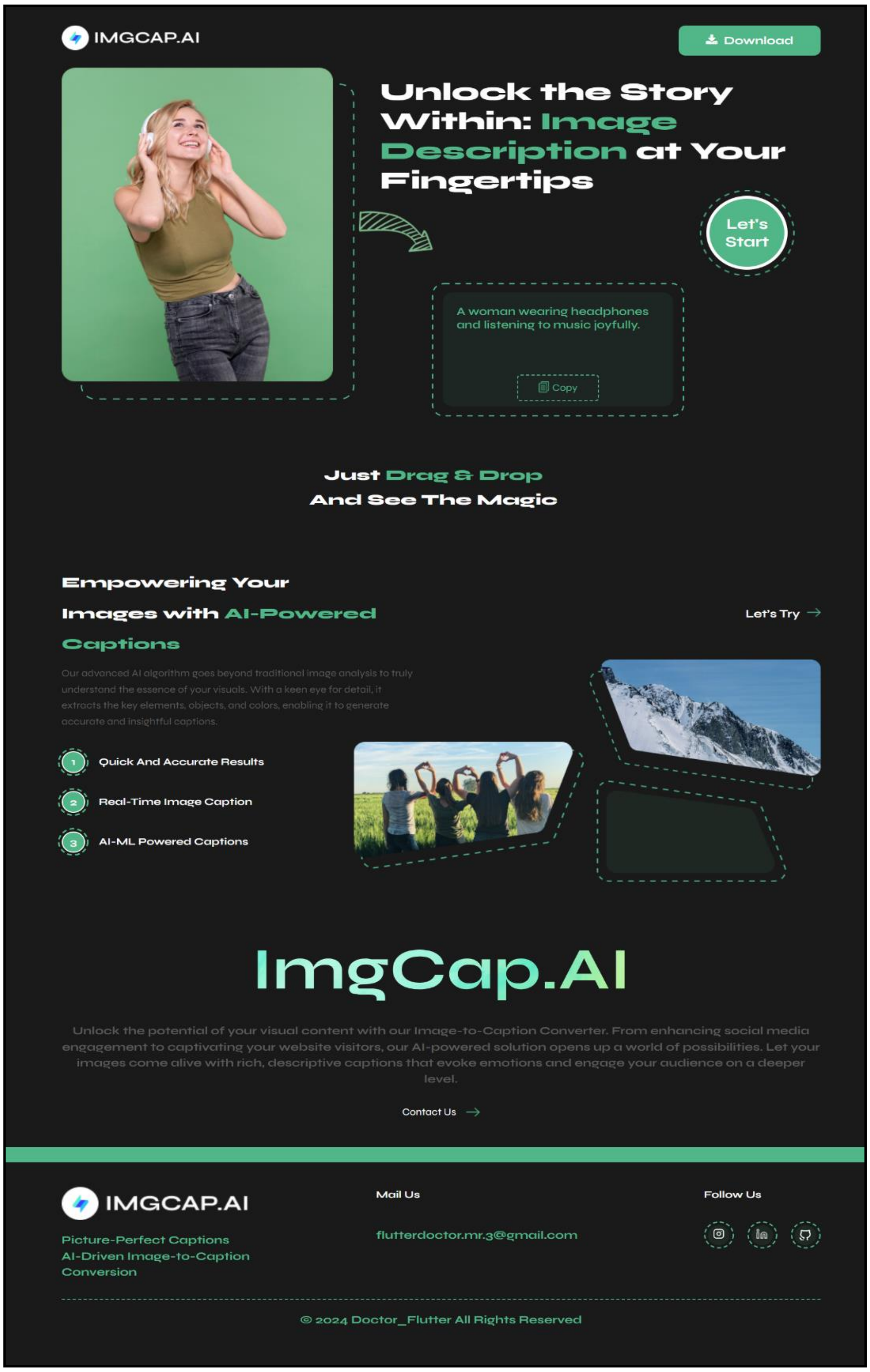
3 a wooden table topped with lots of different types of vegetables

4 a cut in half pizza sitting on top of a wooden cutting board

3. **Copy Caption Screen :**



4. **Website UI where user can download Imagecap.ai App :**



Unlock the Story Within: Image Description at Your Fingertips

A woman wearing headphones and listening to music joyfully.

Just Drag & Drop And See The Magic

Empowering Your Images with AI-Powered Captions

Our advanced AI algorithm goes beyond traditional image analysis to truly understand the essence of your visuals. With a keen eye for detail, it extracts the key elements, objects, and colors, enabling it to generate accurate and insightful captions.

1. Quick And Accurate Results
2. Real-Time Image Caption
3. AI-ML Powered Captions

ImgCap.AI

Unlock the potential of your visual content with our Image-to-Caption Converter. From enhancing social media engagement to captivating your website visitors, our AI-powered solution opens up a world of possibilities. Let your images come alive with rich, descriptive captions that evoke emotions and engage your audience on a deeper level.

IMGCAP.AI

Picture-Perfect Captions
AI-Driven Image-to-Caption Conversion

flutterdoctor.mr.3@gmail.com

© 2024 Doctor_Flutter All Rights Reserved

http://imgcapai.rf.gd/

7. Limitations and Future Scope of Enhancements

❖ Limitations

1. **Limited to Caption Generation :**

   - **The current system is designed specifically for generating captions based on images. It doesn't extend to other forms of media, such as videos or text-based inputs.**

2. **Dependent on API Connectivity** :

   - **The app relies on external APIs for its functionality, such as image captioning and authentication, which means it won't work without internet access.**

3. **Basic Authentication System** :

   - **The login, signup, and forgot password functionalities are implemented, but it lacks advanced features like two-factor authentication or social media login options.**

4. **Limited Error Handling** :

   - **Error messages during image upload, caption generation, or API communication are basic, with potential room for better user guidance.**

5. **Performance with Larger Images** :
   - **Uploading larger images might lead to delays due to server-side image processing.**

6. **Limited Scalability** :

   - **The app may not be able to handle a large number of concurrent users efficiently in its current form due to the threading and API limitations.**

❖ Future Scope of Enhancements

1. **Advanced Caption Customization** :

   - **Introduce more user options for customizing the captions (e.g., humor, sentiment, length) or allowing users to select from multiple models.**

2. **Video Captioning** :

   - **Extend the functionality to support video captioning in addition to images, using advanced deep learning techniques.**

3. **Offline Capabilities** :

   - **Develop offline modes for certain functionalities such as authentication, caching images, and generating captions for previously used datasets**

4. **Integration with Cloud Storage** :

   - **Allow users to store their generated captions and images in cloud services like Google Drive or Dropbox, providing easy access across devices.**

5. **Enhanced Security :**

   - **Implement features like two-factor authentication, password recovery via security questions, or biometric authentication for better security.**

6. **Better UI/UX Design :**

   - **Improve the overall look and feel of the app, with more animations and easier navigation for non-technical users.**

7. **Role-Based Access Control :**

   - **Add role-based login for users and admins, with special privileges for admin users to manage the system more efficiently.**

8. **Scalability for Enterprise Use :**

   - **Enhance the app to handle a higher number of users, with better load balancing and server-side optimizations.**

9. **User Analytics :**

   - **Implement user behavior analytics to track how users interact with the app, providing insights for further improvement.**

10. **Multilingual Support :**

    - **Support different languages for captions and the app's interface, expanding its usability to a global audience.**

8. References

1. **Microsoft Azure Cognitive Services** :

   - **Azure's Computer Vision API provides an advanced service to analyze content in images and generates descriptive captions automatically.**
   - **Website : Azure Computer Vision**

2. **Google Cloud Vision API** :

   - **Google's Cloud Vision API can detect objects, scenes, and generate captions for images using powerful machine learning models.**
   - **Website : Google Cloud Vision**

3. **Clarifai** :

   - **Clarifai offers AI-powered solutions, including image captioning, through its visual recognition tools and APIs.**
   - **Website : Clarifai**

4. **DeepAI Image Captioning** :

   - **This API allows users to generate captions for images using neural networks. It's simple and easy to integrate with existing apps.**
   - **Website : DeepAI**

5. **ImageAI** :
   - **ImageAI is a Python library that can be used to generate captions and analyze images, often used in custom applications.**
   - **Website : ImageAI**