



## ArabTube

- **Under the supervision of:**
  - Dr. Hossam Muftah
  - Teaching assistant: Shaimaa Sayed
- 
- **Prepared by:**
  - Ahmed Yaser Yousef
  - Ahmed Awad Omar
  - Abdulaziz Ahmed Abdulaziz
  - Abdelrahman Mahmoud AbdEltwab
  - Khaled Mohamed Badwy

# Agenda



**Introduction**



**Problem Definition**



**Implementation Techniques**



**Future Work**



**Conclusion**

# Introduction

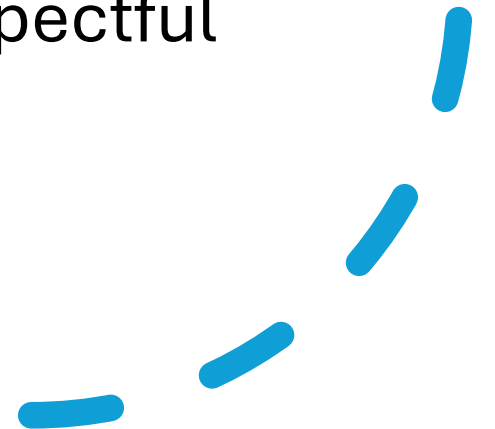
- **Problem:** Online video platforms have tons of content, but it can be inappropriate (NSFW videos, offensive comments). This is a big problem, especially for parents who want to keep their kids safe online.
- **Solution:** We need video platforms that prioritize safety and curate content. This would make the online environment more respectful for everyone.
- **Why this matters:** Safe online spaces are crucial, especially for children who are vulnerable to inappropriate content. A platform like ArabTube, with its focus on safety, is a great example.

# Problem Definition

- **Existing video platforms** are full of inappropriate content (offensive videos, comments).
- This exposes users, especially children, to potentially harmful and disrespectful experiences.
- Parents feel these platforms aren't safe enough and their trust is undermined.

# Problem Definition Cont.

- There's a gap in the market for a platform that prioritizes **content moderation and user safety**.
- Users want a secure space with **educational and entertaining content** suitable for all ages.
- A platform like ArabTube can address this need by offering a safe and respectful online environment.





# Our Solution

- **Goal 1: Advanced AI Moderation**

- Use cutting-edge AI to filter inappropriate content (videos & comments).
- Continuously improve AI to stay effective over time.

- **Goal 2: Family-Friendly Experience**

- Create a safe space for kids to explore and engage with content.
- Offer educational and enriching videos for a positive experience.



## Our Solution Cont.

- **Goal 3: Maximize Accessibility**
  - Offer web and mobile apps for easy access from any device.
  - Ensure a seamless user experience across platforms.
- **Goal 4: Positive Digital Environment**
  - Promote positive interactions and respectful communication.
  - Implement clear community guidelines and enforce them effectively.
  - Regularly train content moderators to handle inappropriate behavior.

# Implementation Techniques



**Deep Learning & Machine Learning Techniques**



**Ui/Ux Techniques**



**Web Techniques**



**Flutter Techniques**



**Back-end Techniques**



# Back-end Techniques : Why .NET ?

- 1. High Performance:** Explain .NET Core's efficiency in handling large volumes of requests.
- 2. Cross-Platform Development:** Highlight support for Windows, macOS, and Linux.
- 3. Integration with Azure and Microsoft Services:** Explain seamless integration with Azure and other Microsoft services.
- 4. Flexibility in Application Types :** Discuss support for web, mobile, desktop, gaming, IoT, and AI applications.

# Back-end Techniques : Tools

- **Azure Blob Storage**

- **Properties:**

- Cloud storage service for unstructured data
    - Supports hot, cool, and archive access tiers

- **Uses:**

- Storing large amounts of unstructured data like images, videos, and backups
    - Serving content directly to clients
    - Disaster recovery and archival storage



# Back-end Techniques : Tools Cont.

- **xFFMPEG.NET**

- **Properties:**

- .NET wrapper for the FFMPEG multimedia framework
    - Allows integration of audio and video processing in .NET applications
    - Supports various formats and codecs

- **Uses:**

- Video and audio transcoding
    - Make a copies of video in multiple quality



# Back-end Techniques : Tools Cont.

- **Entity Framework**
  - **Properties:**
    - Object-Relational Mapper (ORM) for .NET
    - Automates database operations (CRUD) using strongly-typed objects
    - Supports code-first and database-first approaches
  - **Uses:**
    - Simplifies database access and management
    - Promotes a clean separation between data and business logic
    - Facilitates rapid application development with database integration

# Back-end Techniques : Tools Cont.

- **JSON Web Token (JWT)**
  - **Properties:**
    - Open standard for securely transmitting information between parties as a JSON object
    - Signed and optionally encrypted
    - Compact and self-contained
  - **Uses:**
    - Authentication and authorization in web applications
    - Securely transmitting user information
    - Stateless session management

# Back-end Techniques : Tools Cont.

- **Serilog**
  - **Properties:**
    - Structured logging library for .NET
    - Supports various sinks (e.g., console, file, database)
    - Enriches logs with contextual information
  - **Uses:**
    - Enhancing application diagnostics and monitoring
    - Centralized logging with rich data structures
    - Analyzing and querying logs for troubleshooting

# Back-end Techniques : Tools Cont.

- **AutoMapper**
  - **Properties:**
    - Object-to-object mapping library for .NET
    - Automatically maps properties between objects
    - Configurable and customizable mappings
  - **Uses:**
    - Simplifies code for transferring data between objects
    - Reduces boilerplate code for object transformations
    - Supports complex mapping scenarios

# Back-end Techniques : Tools Cont.

- **Microsoft Identity**

- **Properties:**

- Authentication and authorization framework for .NET applications
    - Manages users, roles, and claims

- **Uses:**

- Implementing secure user authentication and management
    - Integrating third-party identity providers
    - Managing access control and authorization in applications





# Back-end Techniques : Tools Cont.

- **LINQ (Language Integrated Query)**
  - **Properties:**
    - Query syntax integrated into .NET languages
    - Supports querying various data sources like collections, SQL databases, XML, etc.
    - Provides a unified model for data access
  - **Uses:**
    - Simplifies data manipulation and querying
    - Enhances code readability and maintainability
    - Supports complex queries with minimal code

# Back-end Techniques : Data Storage

- **SQL Server**
  - **Properties:**
    - Relational Database Management System (RDBMS)
    - Structured Query Language (SQL) for database operations
    - Supports ACID (Atomicity, Consistency, Isolation, Durability) properties
    - Scalable with support for large databases and high transaction volumes
  - **Uses:**
    - Ideal for applications requiring complex queries and transactions
    - Suitable for enterprise-level applications with large datasets
    - Supports data integrity and relationships through constraints and foreign keys

# Back-end Techniques : Data Storage Cont.

- **In-Memory Caching**

- **Properties:**

- Stores data in the system's RAM for fast access
    - Volatile storage, data is lost on power down or restart
    - Supports key-value storage model for simplicity and speed
    - Can be distributed across multiple nodes for scalability

- **Uses:**

- Enhances application performance by reducing database load
    - Ideal for storing session data, user preferences, and frequently accessed data
    - Improves response times for high-traffic applications by serving data quickly from memory

# Back-end Techniques : Architecture Pattern

- N-Tier Architecture Pattern :
  - **Presentation Layer**
    - **Properties:** User interface layer.
    - **Uses:** Displays data to the user and collects user inputs.
  - **Logic Layer**
    - **Properties:** Business logic layer.
    - **Uses:** Processes user inputs and makes decisions.
  - **Data Access Layer**
    - **Properties:** Interface to the data layer.
    - **Uses:** Communicates between the logic layer and the data layer
  - **Data Layer**
    - **Properties:** Data storage layer.
    - **Uses:** Stores and retrieves data.

# Back-end Techniques : Design Patterns

- **Repository Pattern**

- **Properties:**

- Abstracts data access logic
    - Centralizes data access, avoiding duplication
    - Provides a collection-like interface for accessing domain objects

- **Uses:**

- Promotes a clean separation between the business logic and data access logic
    - Simplifies data access code, enhancing maintainability



# Back-end Techniques : Design Patterns

- **Unit of Work Pattern**

- **Properties:**

- Manages transactions and ensures consistency
    - Tracks changes to objects during a business transaction
    - Coordinates the writing out of changes and the resolution of concurrency issues

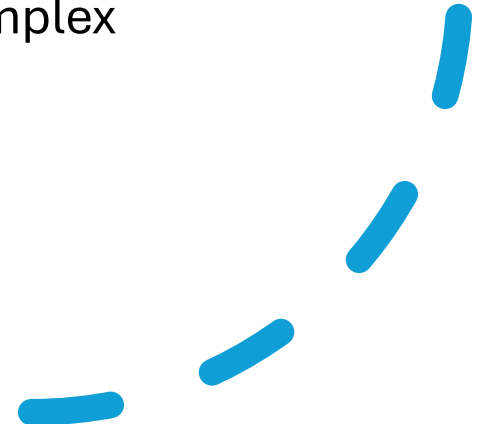
- **Uses:**

- Ensures that all changes within a transaction are committed or rolled back as a single unit
    - Improves performance by reducing multiple database calls
    - Simplifies complex transactional operations by managing them in a single class



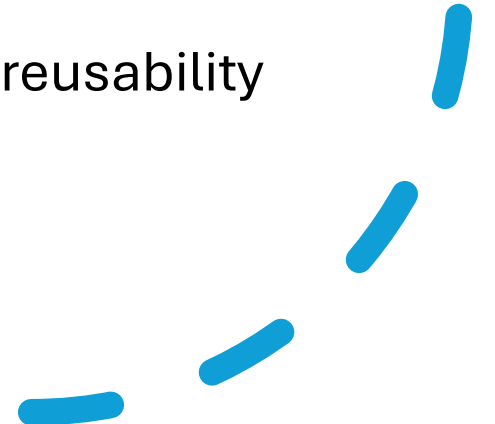
# Back-end Techniques : Design Patterns Cont.

- **Dependency Injection (DI)**
  - **Properties:**
    - Implements Inversion of Control (IoC) principle
    - Decouples object creation from business logic
    - Injects dependencies through constructors, setters, or interfaces
  - **Uses:**
    - Enhances testability and flexibility by allowing dependencies to be easily swapped
    - Promotes loose coupling and adheres to the Single Responsibility Principle
    - Facilitates the management of complex dependency graphs in application



# Back-end Techniques : Design Principles

- **Single Responsibility Principle (SRP)**
  - **Properties:**
    - A class should have only one reason to change
    - A class should only have one job or responsibility
  - **Uses:**
    - Simplifies debugging and maintenance by isolating changes
    - Enhances code readability and reusability





# Back-end Techniques : Design Principles Cont.

- **Interface Segregation Principle (ISP)**
  - **Properties:**
    - Interfaces should be small and specific to client needs
    - Promotes the creation of lean, focused interfaces
  - **Uses:**
    - Reduces the impact of changes by minimizing the scope of interfaces
    - Improves system flexibility and scalability
    - Facilitates easier implementation and substitution of components



# Back-end Techniques : Design Principles Cont.

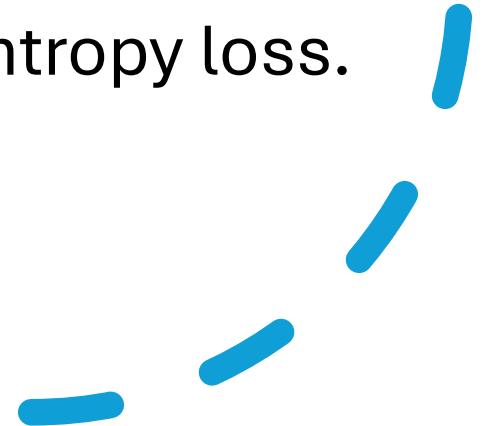
- **Dependency Inversion Principle (DIP)**
  - **Properties:**
    - High-level modules should not depend on low-level modules; both should depend on abstractions
    - Abstractions should not depend on details; details should depend on abstractions
    - Promotes decoupling and flexibility in the system design
  - **Uses:**
    - Enhances code stability and adaptability by minimizing dependency chains
    - Supports the use of dependency injection for managing dependencies
    - Facilitates the design of more modular and testable code

# Deep Learning Techniques : Automated Video Content Classification

- **Detecting pornography** in videos using computer vision.
- **Importance of Automated Classification for Content Moderation and Safety :**
  - Scale and efficiency: Rapidly analyze massive amounts of content beyond manual capability.
  - Risk mitigation: Identifying and rejecting harmful or inappropriate content.
  - User Trust and Experience: Maintaining a safe and respectful online environment enhances user satisfaction.

# Deep Learning Techniques : InceptionV3 Transfer Learning Model

- **Utilizes InceptionV3**, a pre-trained **CNN** on the ImageNet dataset.
- **Added layers for fine-tuning:**
  - Global Average Pooling, Dense layers for feature extraction.
  - Sigmoid output for binary classification.
- Compiled with **Adam optimizer** (lr=0.0001) and binary cross-entropy loss.



# Deep Learning Techniques :Training Process

- **Data sourced** from a curated dataset using **web scraping**.
- **Data augmentation** (e.g., rescaling, flipping) to improve generalization.
- **Model checkpointing** to save the best performing model based on validation loss.

# Deep Learning Techniques: Model Performance

- Evaluated on a separate test set using **metrics['accuracy']**
- **Achieved test accuracy: [87.5].**
- **significance of accuracy in relation to the problem domain:**
  - **Accuracy is Critical:** Ensures reliable identification of inappropriate or harmful content, crucial for maintaining platform integrity and user safety.
  - **Impact on User Experience:** High accuracy minimizes false positives, reducing the chance of legitimate content being incorrectly flagged or removed.

# Deep Learning Techniques : Future Scope

- **Integration with Live streaming .**
- **Continuous learning with user feedback for better accuracy.**



# Machine Learning Techniques : Twitter Hate Speech Detection Model

- **Build a machine learning model to detect hate speech, offensive language, and non-offensive.**
- **Dataset:** Collected tweets labeled as:
  - 0: Hate Speech Detected
  - 1: Offensive language detected
  - 2: No hate and offensive speech
- **Data Preparation:** Cleaned the tweets and mapped labels for easier interpretation.





# Machine Learning Techniques : Data Cleaning and Preprocessing

- **Text Cleaning Steps:**

- Convert to lowercase
- Remove URLs, HTML tags, punctuation, numbers, and stopwords
- Lemmatize words to their base form

- **TF-IDF Vectorization:**

- Transformed text data into numerical features using TF-IDF
- Limited to 5000 features for better performance



# Machine Learning Techniques : Model Building and Training

- **Algorithms Used:**
  - Support Vector Machine (**SVM**)
  - Logistic Regression
- **Parameter Tuning:**
  - Used **GridSearchCV** to find the best parameters for SVM
  - Best parameters: {'C': x, 'kernel': y, 'gamma': z}
- **Training:**
  - Split data into training (**67%**) and testing (**33%**) sets
  - Trained individual models and an ensemble model using **VotingClassifier**

# Machine Learning Techniques : Model Evaluation and Results

- **Ensemble Model:** Combined SVM, RandomForest, and Logistic Regression
- **Performance Metrics :**
  - **Accuracy:**
    - Ensemble Model: **90.0**
    - Cross-validated Accuracy: **90.0**
- **Interpretation:** High accuracy and balanced performance across classes

# Machine Learning Techniques: Predictions and Conclusion

- **Example Predictions:**
  - "I will kill you" -> Hate Speech Detected
  - "You are awesome" -> No hate and offensive speech
- **Conclusion:**
  - The ensemble model effectively detects hate speech and offensive language
  - Potential for real-world applications in moderating social media content

# UI/Ux Techniques : Understanding

- **UI/UX design** refers to designing user interfaces and experiences for digital products such as websites, mobile apps, and software applications.
- **Goal:** To create visually appealing, functional, and intuitive interfaces that enhance user experience.
- **UI Design:** Focuses on visual layout and aesthetics: color, typography, icons.
  - **Role:** Create visually consistent designs aligning with brand and user needs.
- **UX Design:** Focuses on overall user experience: information architecture, user flows, interactions.
  - **Role:** Create a seamless and intuitive experience enabling users to achieve their goals efficiently.

# UI/Ux Techniques :Design Process

- **1. User Research:**
  - - Understand the target audience's needs and behaviors.
- **2. Wireframing:**
  - - Create initial layouts to outline the structure and functionality.
- **3. Prototyping:**
  - - Develop interactive models to test user interactions and flows.
- **4. Visual Design:**
  - - Apply final visual elements to create an engaging and cohesive interface.
- **5. Usability Testing:**
  - - Continuously test with users and iterate on feedback.

# UI/Ux Techniques :Tools

- **Component State :**
  - **des:** Enable you to create multiple copies of the same component with minor changes in each copy, such as buttons in different colors or with different text values.
  - **using:** It will serve to save time and make the design easier to manage, as you can modify a single component and the changes will be reflected in all copies
- **Overlay:**
  - **des :**It allows you to add elements on top of your existing design, such as pop-up menus or animated windows.
  - **using :** It is used to create interactive experiences in which windows or menus appear over the existing design without navigating to a new page, improving the user experience

# UI/Ux Techniques :Tools

## Animate :

- **des:** A tool for adding animations and transitions between different states or screens in your design.
- **using:** They are used to make the design more dynamic and attractive, such as smooth transitions between screens or simple movements of elements to attract the user's attention.

## Hover :

- **des:** Allows you to change the state of components when the user hovers over them with the mouse.
- **Using :**Used to show immediate user interactions, such as changing the color of a button when the mouse hovers over it



# UI/Ux Techniques :Tools

## Plugins :

**des:** Plugins are external tools that you can add to Figma to expand its capabilities.

**using:** Plugins are used to improve workflow and make it easier to do certain tasks, such as quickly adding icons or images to designs.

## Iconify for icons :

**des:** Iconify is a plugin that allows you to search and add icons to your designs easily.

**Using:** This plugin is used to quickly insert a wide range of icons into designs without having to search for them outside of Figma, saving time and making designing easier.

# UI/Ux Techniques :Tools

## **Unsplash for photos :**

**des:** It is a plugin that gives you access to a huge library of free high-quality images.

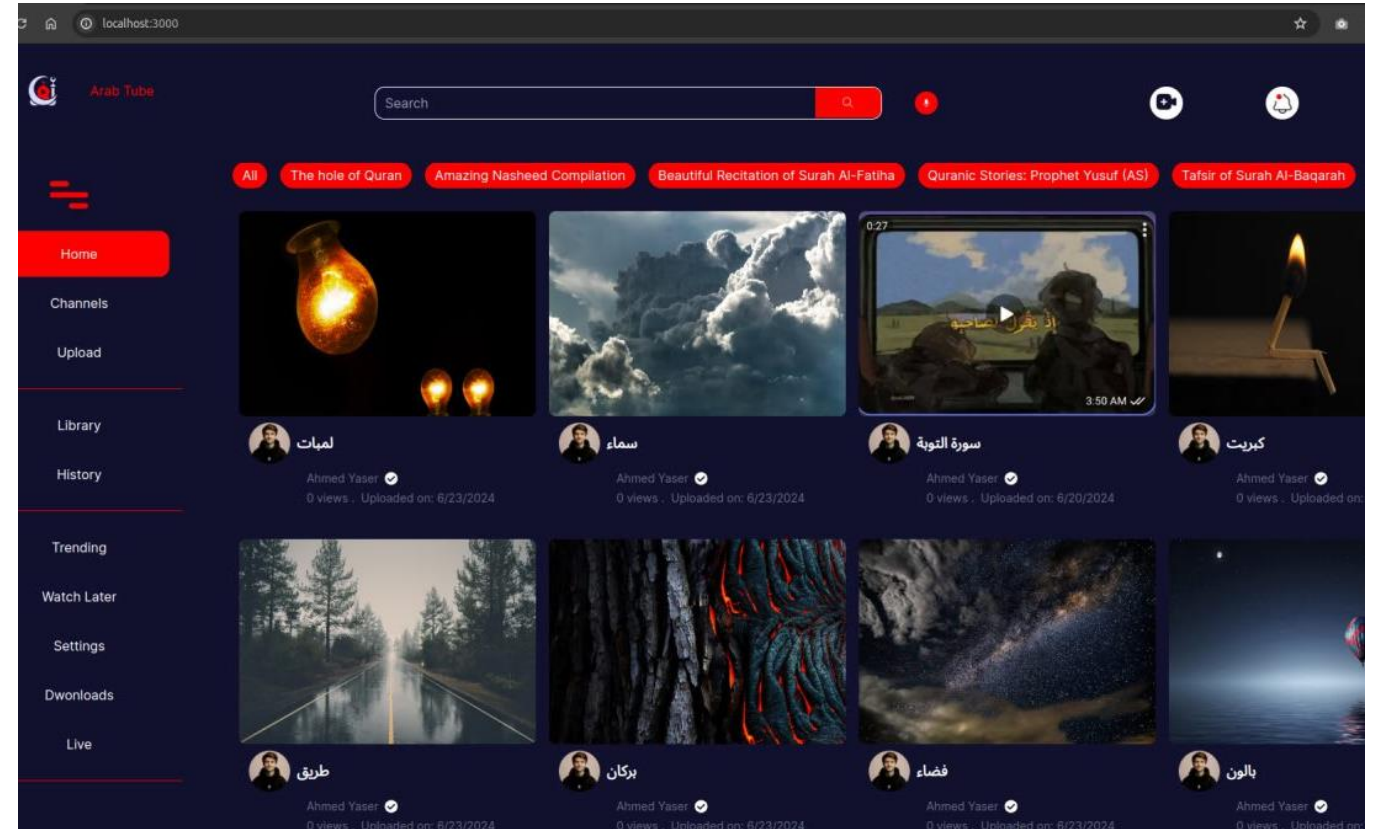
**using:** This plugin is used to add beautiful, high-quality images to designs directly from your Unsplash library, making it easier to find the right images to enhance your design.

## **Images extension svg :**

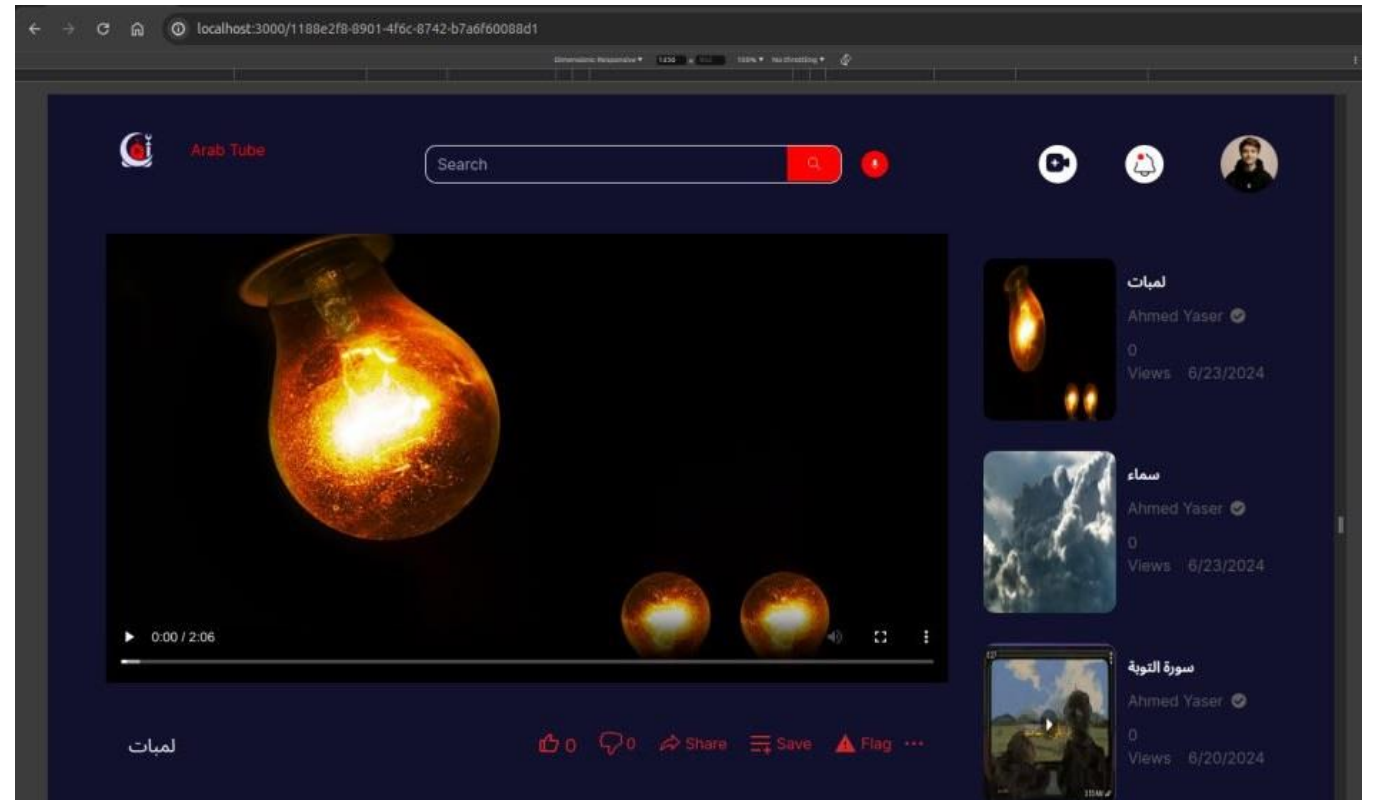
**des:** The SVG image extension allows you to add and edit vector graphics in your designs

**using :** It is used to insert scalable vector graphics without losing quality, making it ideal for use in logos, icons, and illustrations that need high definition regardless of their size.

# Web Techniques : Home Page



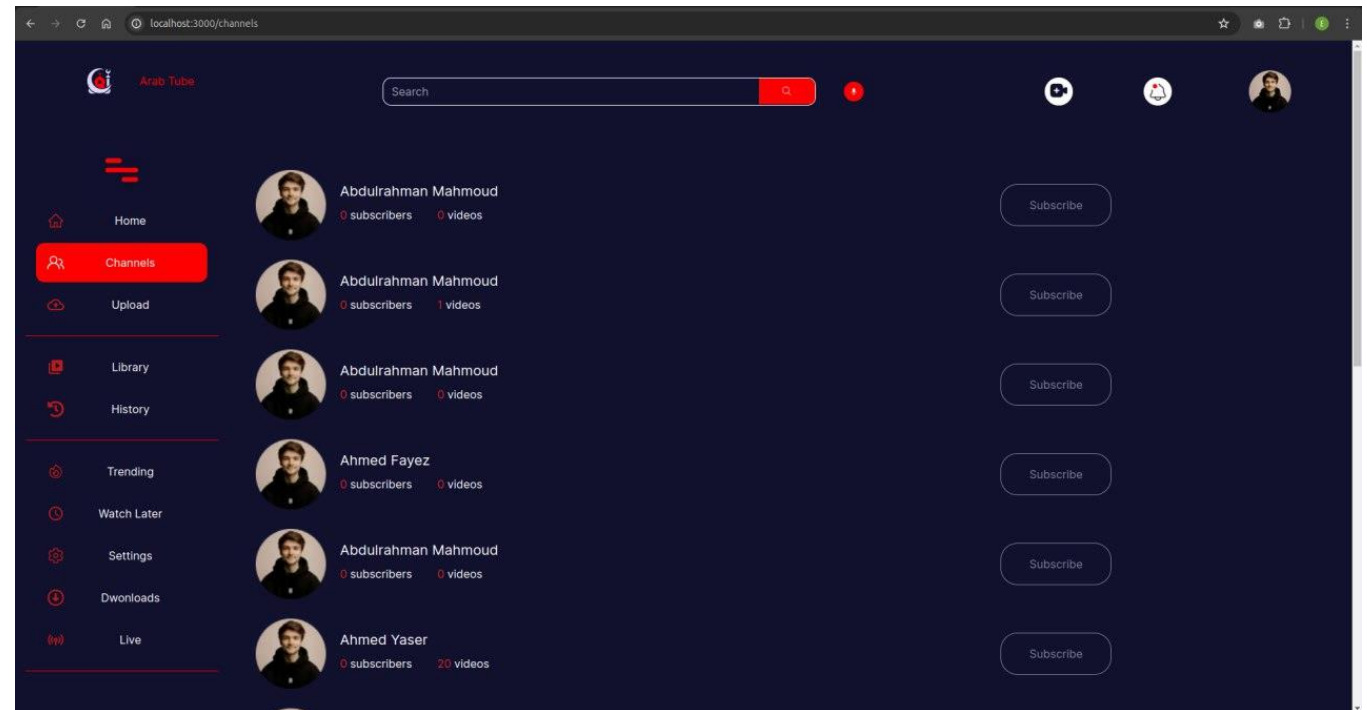
# Web Techniques : Video page



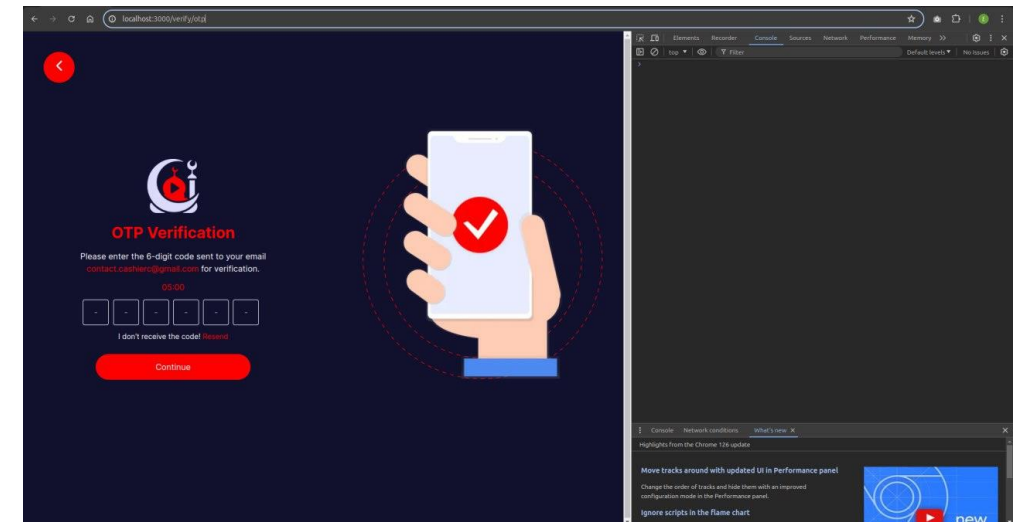
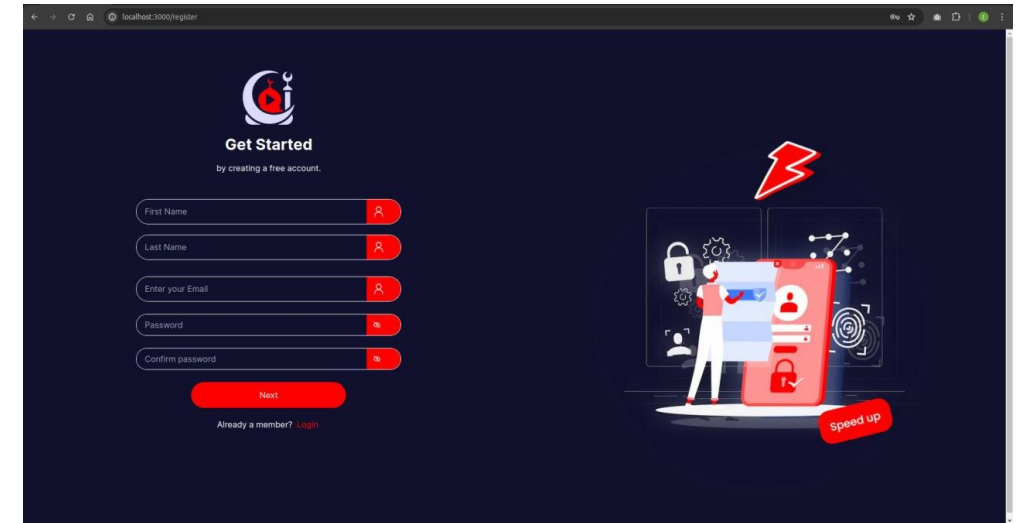
# Web Techniques : Upload Video



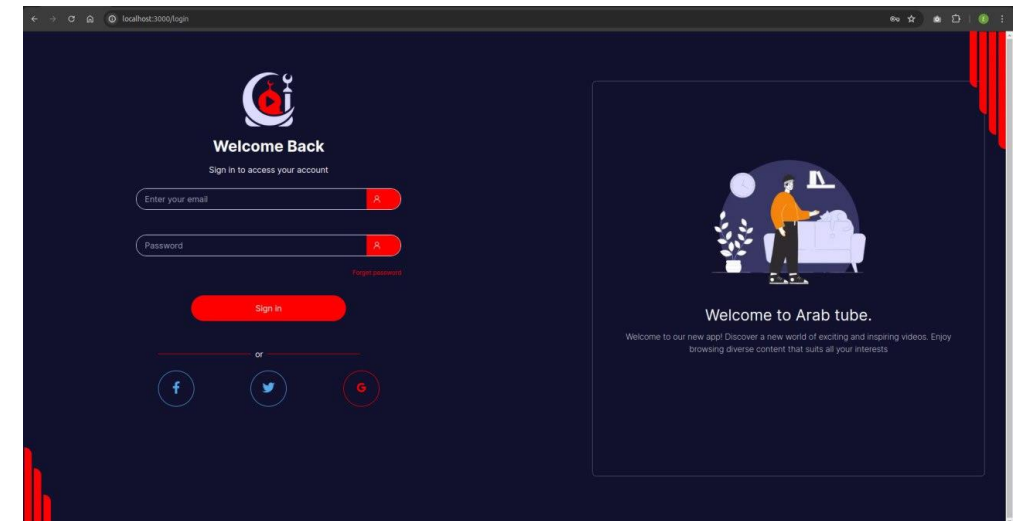
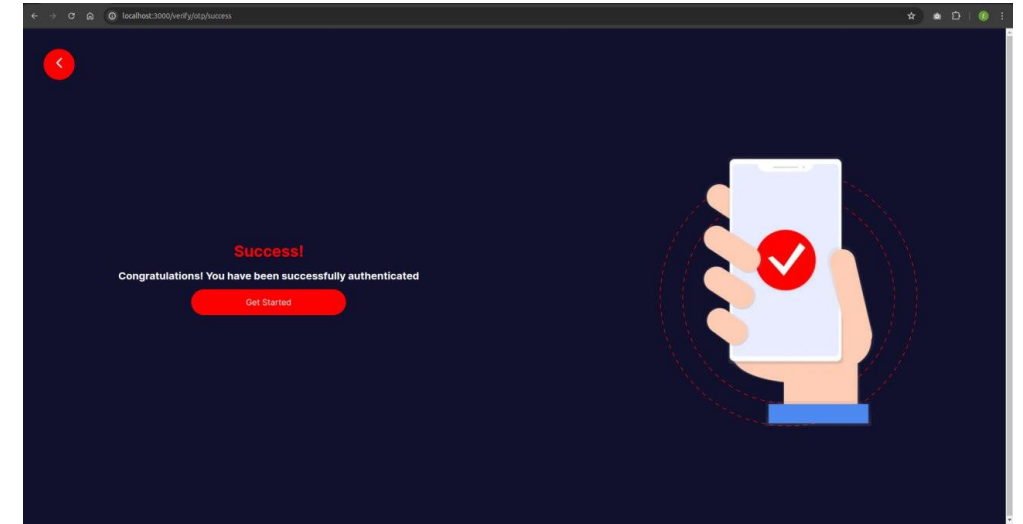
# Web Techniques : Channels Page



# Web Techniques : Register / Confirm Email Pages



# Web Techniques : Success Confirm / Login Pages





# Web Techniques :Technologies Overview

- **HTML :**
  - **Properties:** Markup language for creating web pages.
  - **Uses:** Structure content on the web (text, images, links).
- **CSS**
  - **Properties:** Style sheet language.
  - **Uses:** Design and layout of web pages (colors, fonts, spacing).
- **Tailwind CSS**
  - **Properties:** Utility-first CSS framework.
  - **Uses:** Rapidly build custom designs using utility classes.

# Web Techniques :Technologies Overview Cont.

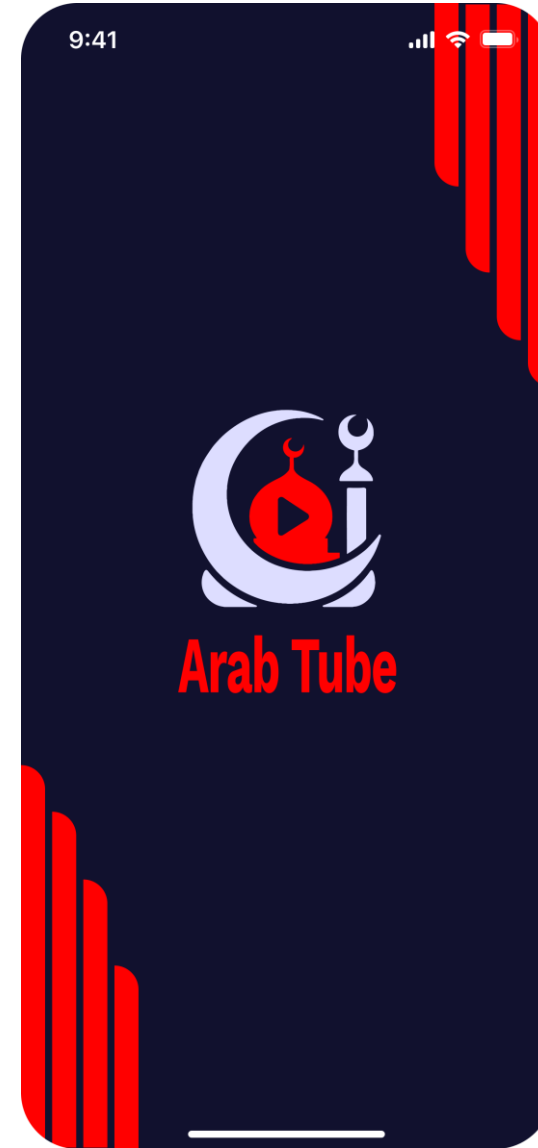
- **JavaScript (JS)**
  - **Properties:** High-level, interpreted programming language.
  - **Uses:** Adds interactivity to web pages (animations, form validation).
- **React 18**
  - **Properties:** JavaScript library for building user interfaces.
  - **Uses:** Create reusable UI components, manage application state.
- **Next.js 14**
  - **Properties:** React framework for server-rendered applications.
  - **Uses:** Build fast, SEO-friendly web applications.

# Web Techniques :Technologies Overview Cont.

- **App Router**
  - **Properties:** Routing system in Next.js.
  - **Uses:** Navigate between pages in a Next.js application.
- **Axios**
  - **Properties:** Promise-based HTTP client.
  - **Uses:** Perform HTTP requests to interact with APIs.
- **React Icons**
  - **Properties:** Library of popular icons.
  - **Uses:** Easily integrate icons into React applications.

# Flutter Techniques : splash\_screen

---



# Flutter Techniques : Home Page

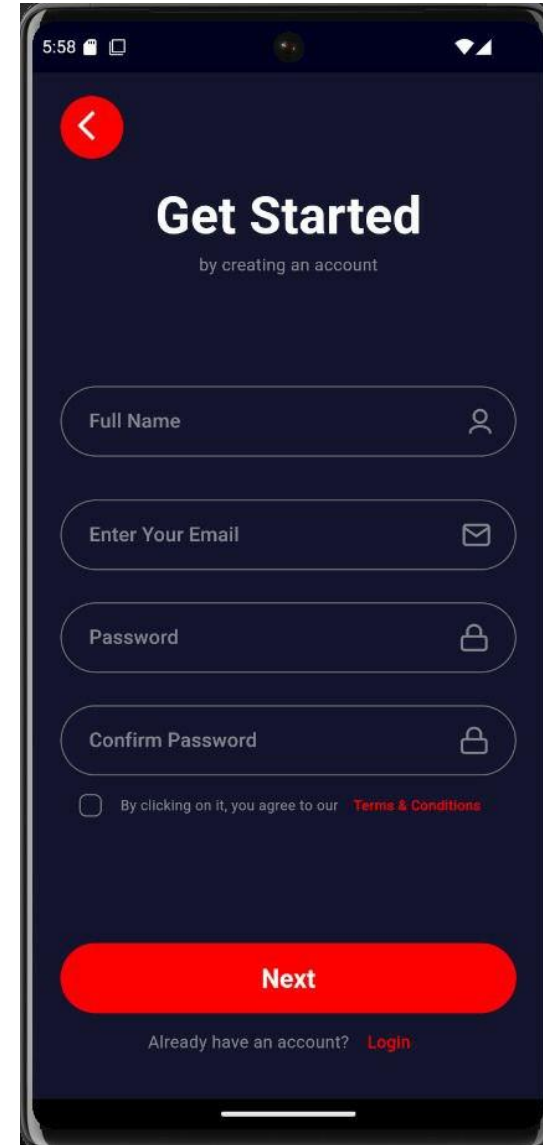


# Flutter Techniques : History Page

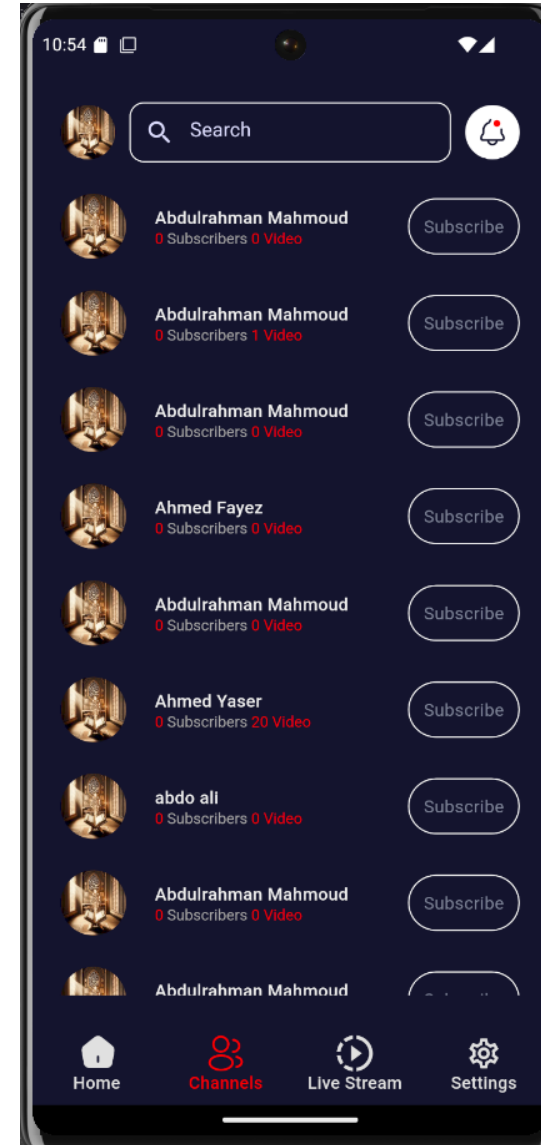


# Flutter Techniques : Upload Page

---



# Flutter Techniques : Channels Page






# Flutter Techniques : Register Page


9:41


<


## Get Started


by creating a free account.


Full Name 

Date of Birth 12/12/91 

Gender\* Gender 

Enter your email 

Password 

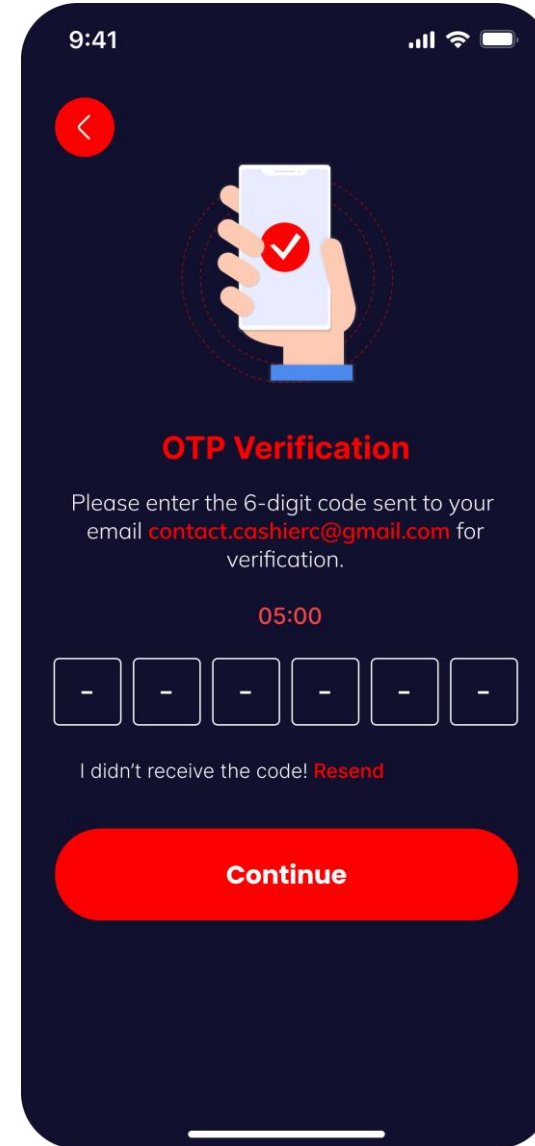
Confirm password 

☐ By checking the box you agree to our [Terms](#) and [Conditions](#).

Next

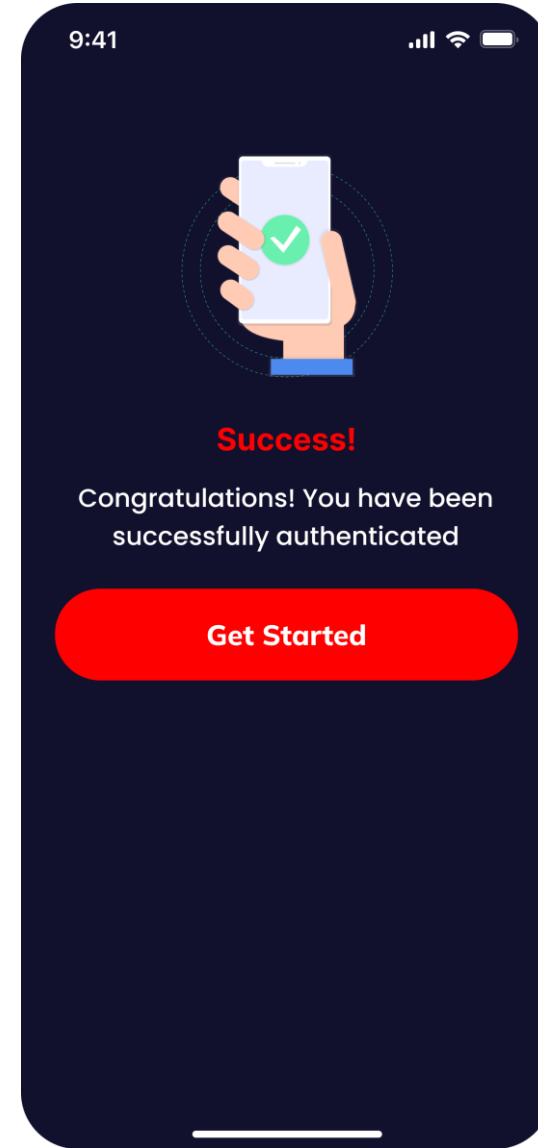
Already a member? [Log In](#)

# Flutter Techniques : Confirm Email Page

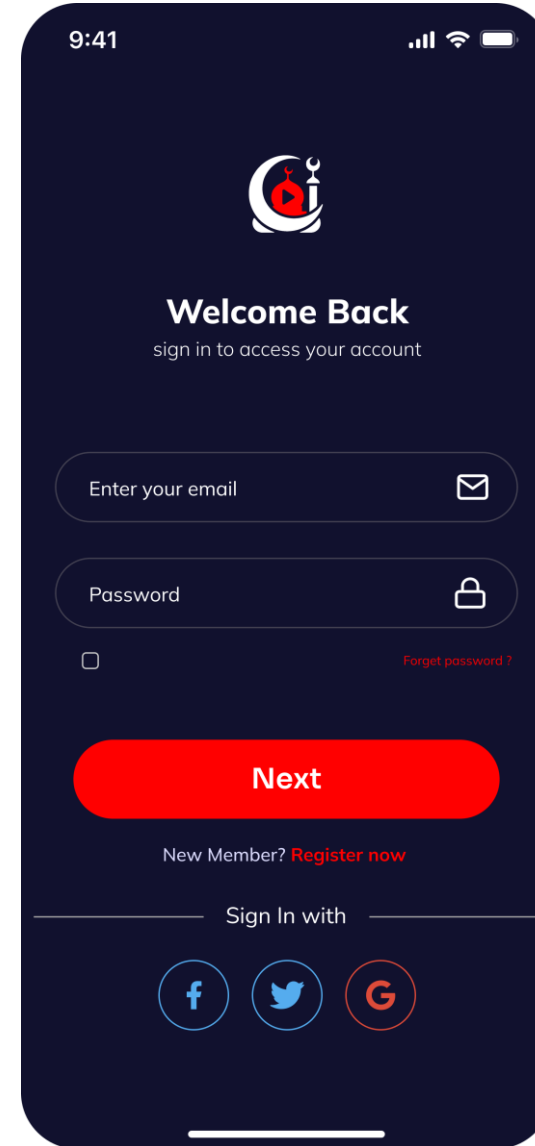


# Flutter Techniques : Success Confirm Page

---



# Flutter Techniques : Login Page



# Flutter Techniques : pubspec.yaml


```
1 name: arabtube
2 description: "A new Flutter project."
3
4 publish_to: 'none'
5
6 version: 1.0.0+1
7
8 environment:
9   sdk: '>=3.4.0-202.1.beta <4.0.0'
10
11 dependencies:
12   animated_bottom_navigation_bar: ^1.3.3
13   cached_network_image: ^3.3.0
14   chewie: ^1.8.1
15   cupertino_icons: ^1.0.6
16   dart: ^0.0.1
17   dio: ^5.4.3+1
18   flutter:
19     sdk: flutter
20   flutter_bloc: ^8.1.6
21   flutter_datetime_picker: ^5.1.0
22   flutter_launcher_icons: ^0.13.1
23   flutter_native_splash: ^2.4.0
24   flutter_riverpod: ^2.5.1
25   flutter_screenutil: ^5.9.0
26   flutter_speed_dial: ^7.0.0
27   flutter_svg: ^2.0.10+1
28   get_it: ^7.6.0
29   image_picker: ^1.1.2
30   internet_connection_checker: ^1.0.0+1
31   miniplayer: ^1.0.1
32   pininput: ^4.0.0
33   shared_preferences: ^2.0.15
34   slimmer: ^3.0.0
35   timezone: ^3.6.1
36   video_player: ^2.8.7
37   rago_util_prebuilt_live_streaming: ^3.0.4
38
39 flutter_native_splash:
40   color: "#111122"
41   background_image: "assets/images/splash_screen.png"
42   image: "assets/images/arabtube.png"
43   android: true
44   ios: true
45
46 flutter_launcher_icons:
47   android: "launcher_icon"
48   ios: true
49   image_path: "assets/images/arabtube.png"
50   min_sdk_android: 21 # android min sdk min:16, default 21
51   web:
52     generate: true
53     image_path: "assets/images/arabtube.png"
54     background_color: "#111122"
55     theme_color: "#111122"
56   windows:
57     generate: true
58     image_path: "assets/images/arabtube.png"
59     # icon_size: 48 # min:48, max:256, default: 48
60   macos:
61     generate: true
62     image_path: "assets/images/arabtube.png"
63
64 dev_dependencies:
65   flutter_lints: ^3.0.0
66   flutter_test:
67     sdk: flutter
68
69 flutter:
70
71   uses-material-design: true
72
73   assets:
74     - assets/images/
75     - assets/icons/
76     - .env
77   # fonts:
78   #   - family: Schyler
79   #     fonts:
80   #       - asset: fonts/Schyler-Regular.ttf
81   #       - asset: fonts/Schyler-Italic.ttf
82   #       - asset: fonts/Schyler-Italic.ttf
83   #     - family: Trajan Pro
84   #     fonts:
85   #       - asset: fonts/TrajanPro.ttf
86   #       - asset: fonts/TrajanPro_Bold.ttf
87   #     weight: 700
88
```

## Conclusion:

- **this project has presented a promising approach to creating a safer and more positive user experience on YouTube. By leveraging machine learning for content detection, the application can offer significant benefits in reducing exposure to harmful content. Future work should focus on refining the accuracy of the models, addressing contextual nuances, and ensuring transparency and user control. The successful development and implementation of this application have the potential to significantly improve the online video viewing experience for a vast global audience.**

A large orange circle is positioned on the left side of the slide, partially cut off by the edge.

## Future Work :

- **Integration of Distributed Memory Cache (e.g., Redis)**
  - **Upgrade to Distributed System Instead of Monolithic System**
  - **Utilization of Microservices Architecture**
  - **Detection of Sexual Content in Live Streams [AI]**
- 
- A blue dashed line is located in the bottom right corner of the slide, consisting of several short, curved segments.

A large, solid orange circle occupies the left side of the frame, partially cut off by the edge.

THANKS

ArabTube Team.

