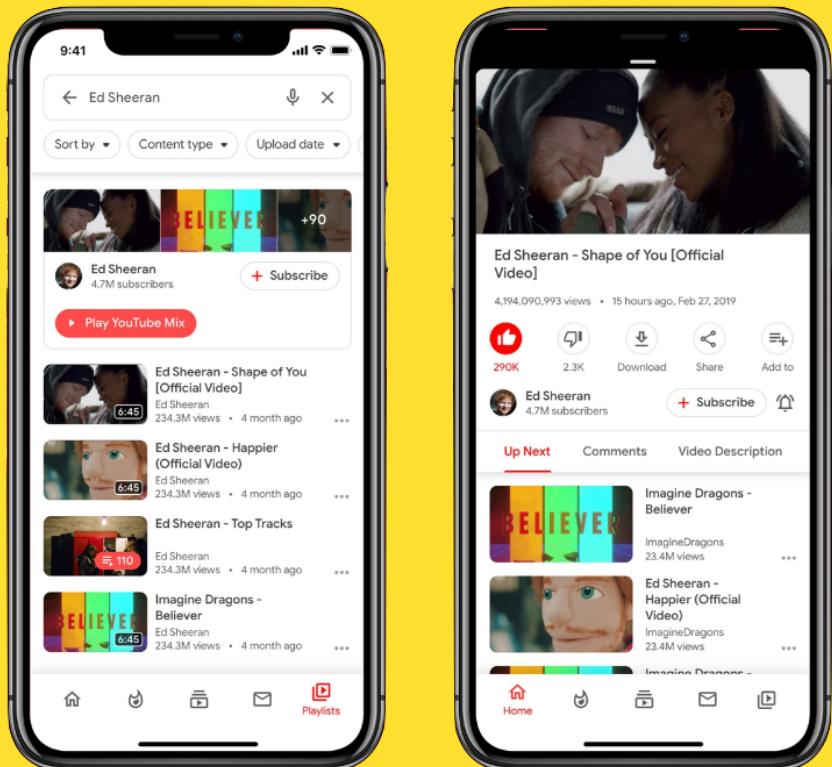


notJust

//development



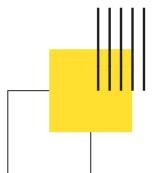
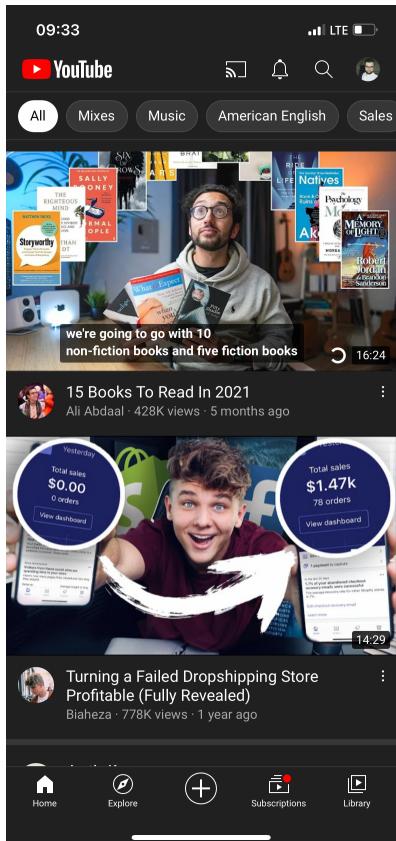
Youtube Clone

React Native Tutorial for Beginners



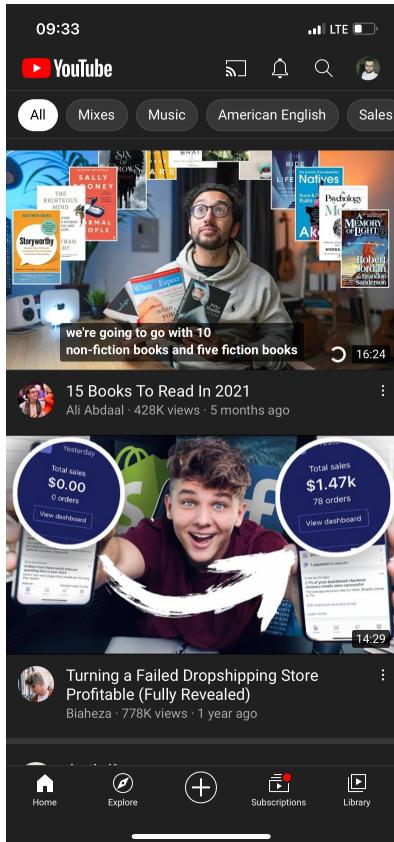
What will we build?

Home Screen

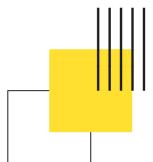
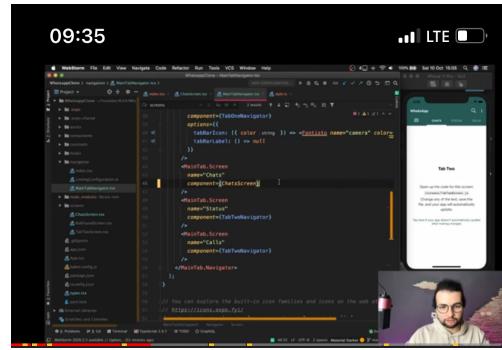


What will we build?

Home Screen

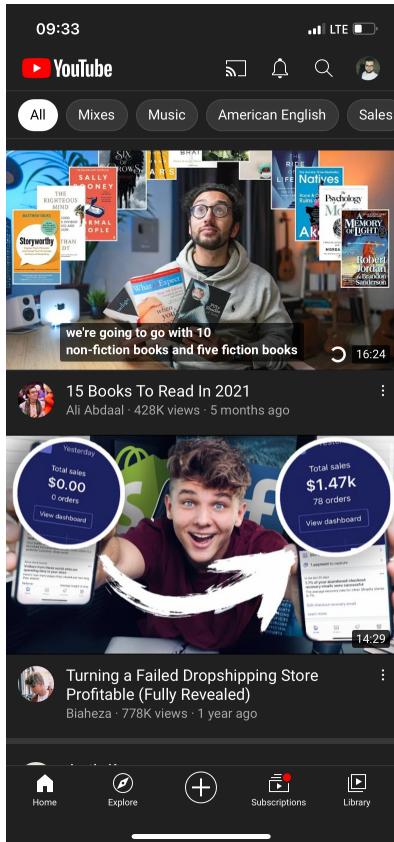


Video Player

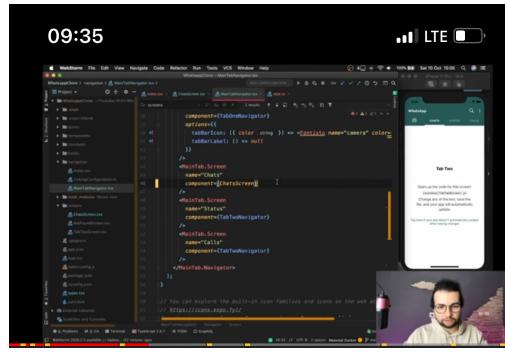


What will we build?

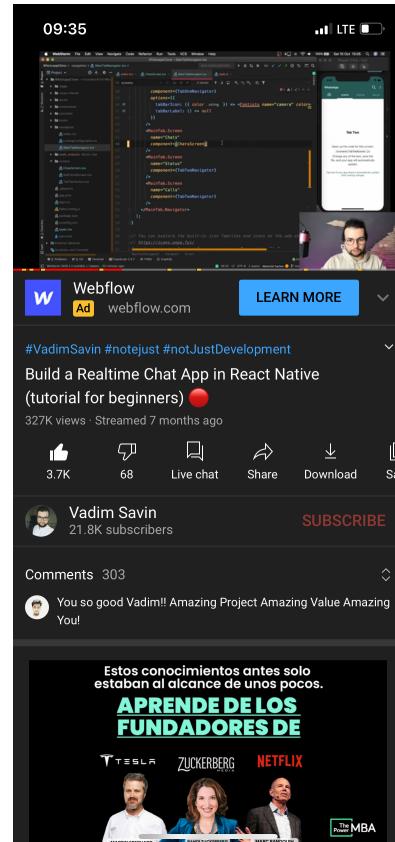
Home Screen



Video Player



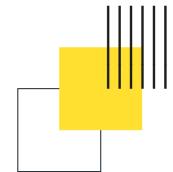
Video Page



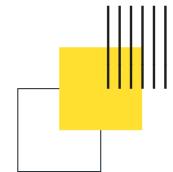
Front-end technologies



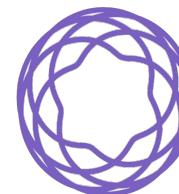
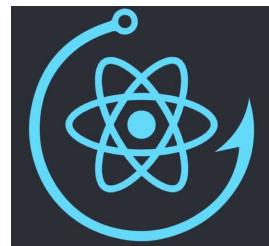
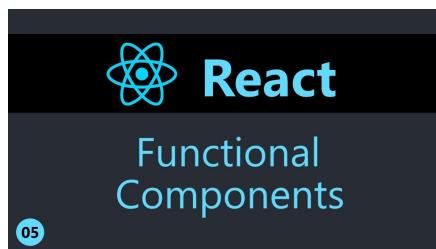
Expo



Front-end technologies



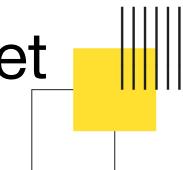
Front-end technologies



React Navigation

Expo AV

React Native Bottom Sheet



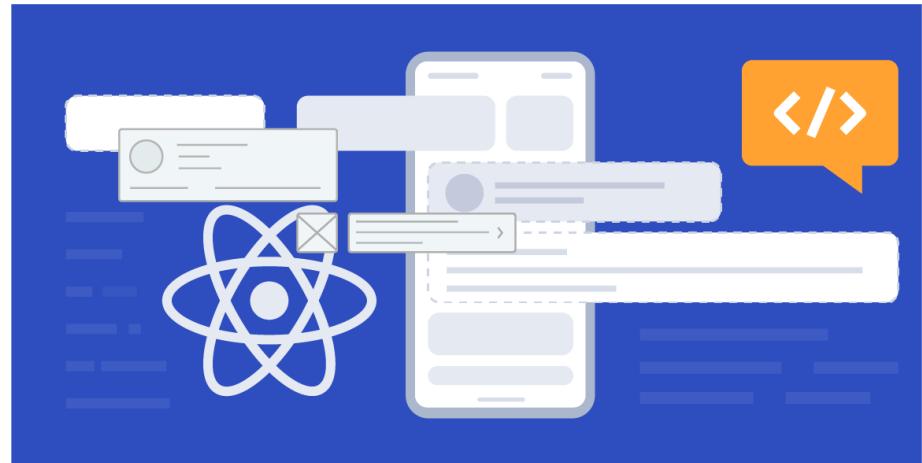
You will learn

- Setup a React Native project using Expo
- |||
- |||
- |||
- |||
- |||
- |||
- |||



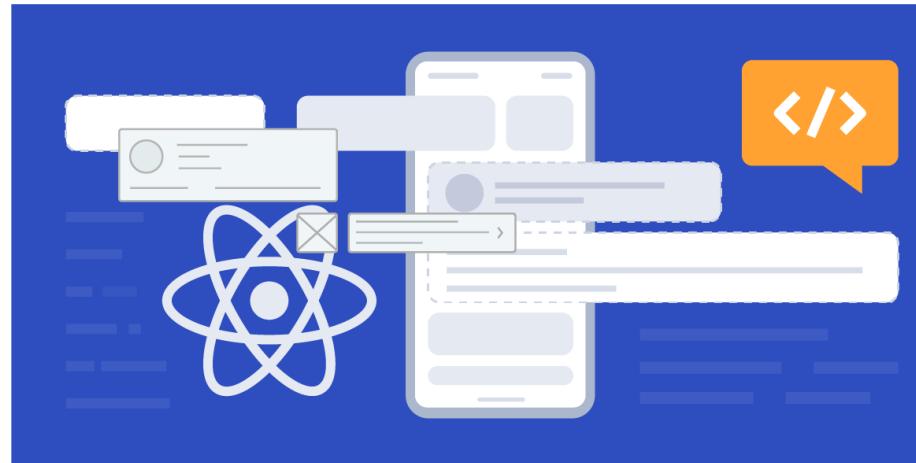
You will learn

- Setup a React Native project using Expo
- Render built-in elements/components



You will learn

- Setup a React Native project using Expo
- Render built-in elements/components
- Create your own components



You will learn

- ||| - Setup a React Native project using Expo
- ||| - Render built-in elements/components
- ||| - Create your own components
- ||| - Functional components & Hooks
- |||
- |||
- |||
- |||
- |||



You will learn

- ||| - Setup a React Native project using Expo
- ||| - Render built-in elements/components
- ||| - Create your own components
- ||| - Functional components & Hooks
- ||| - Props & State

- |||

- |||

- |||

- |||



You will learn

- ||| - Setup a React Native project using Expo
- ||| - Render built-in elements/components
- ||| - Create your own components
- ||| - Functional components & Hooks
- ||| - Props & State
- ||| - Navigation



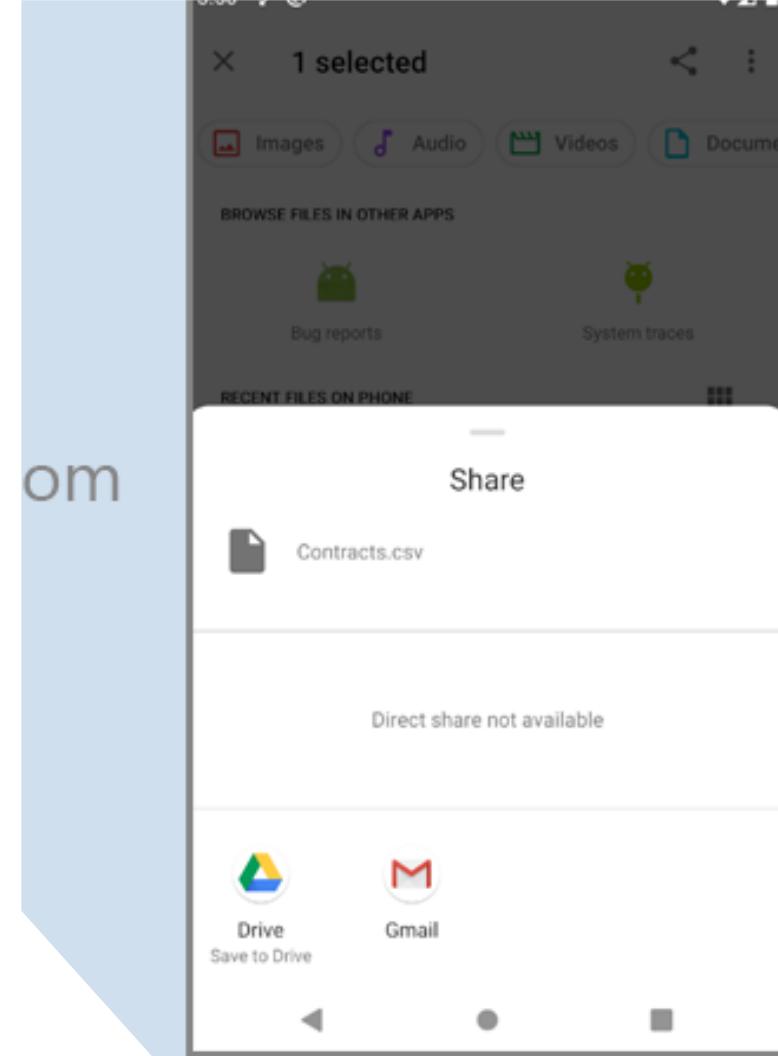
You will learn

- ||| - Setup a React Native project using Expo
- ||| - Render built-in elements/components
- ||| - Create your own components
- ||| - Functional components & Hooks
- ||| - Props & State
- ||| - Navigation
- ||| - Install 3rd party libraries



You will learn

- ||| - Setup a React Native project using Expo
- ||| - Render built-in elements/components
- ||| - Create your own components
- ||| - Functional components & Hooks
- ||| - Props & State
- ||| - Navigation
- ||| - Install 3rd party libraries
- ||| - Bottom Sheet



You will learn

- ||| - Setup a React Native project using Expo
- ||| - Render built-in elements/components
- ||| - Create your own components
- ||| - Functional components & Hooks
- ||| - Props & State
- ||| - Navigation
- ||| - Install 3rd party libraries
- ||| - Bottom Sheet
- ||| - React Native Best Practices





Who am I



- Full stack developer for >7yrs
- CTO @Fitinium
- ex Amazon SDE
- Passionate about coding and building impactful startups.



I AM **notJust**
//developer



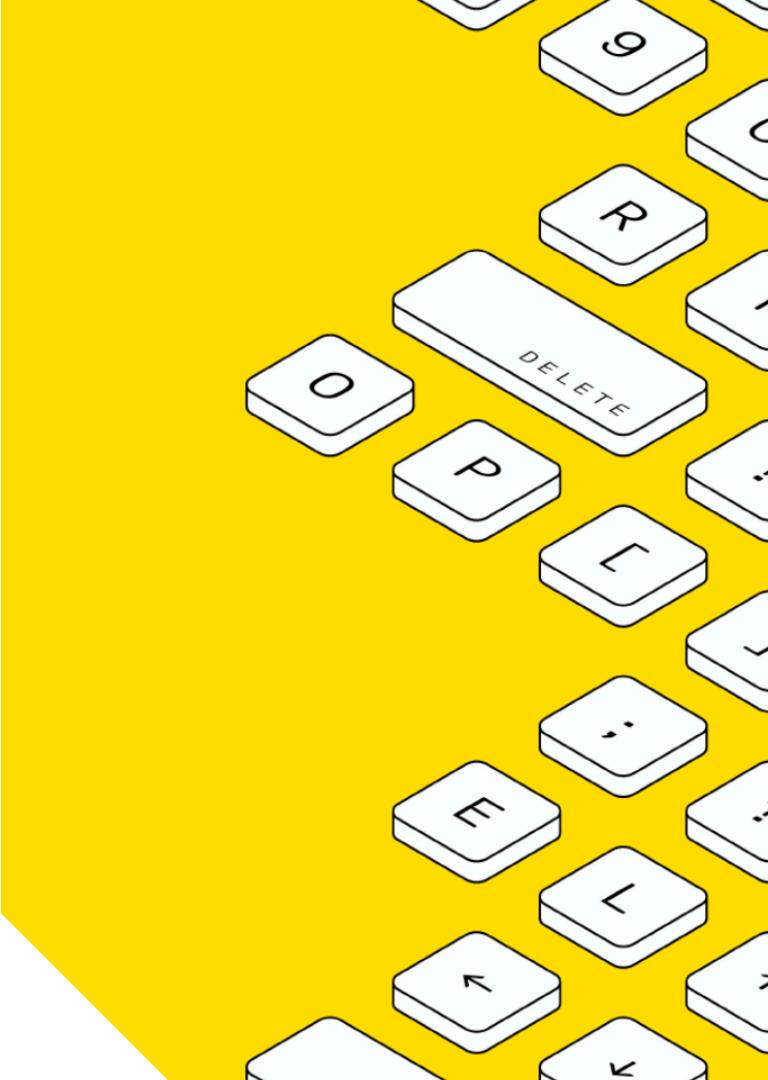
@VadimNotJustDev



New channel members

O	Olawale Lasisi	notJust Developer Junior
	Narik Vibes	notJust Developer Mid
	VIAJA COSTA RICA	notJust Developer Junior
	MOHAMMED KAMAL	notJust Developer Mid
	webgoonie	notJust Developer Junior
	Sebastian Pienio	notJust Developer Senior
	Olanrewaju Olatunde	notJust Developer Mid
	Gbenga Fagbemi	notJust Developer Junior

I AM **notJust**
//developer



Prerequisites

1. Asset Bundle (dummy data, images, icons, PDF presentation, unlimited karma):



<https://assets.notjust.dev/youtube>



2. Expo CLI installed



<https://docs.expo.io/get-started/installation/>



notJust

//development



Let's get started



Let's get started

- ||| 1. Initialise the expo project:

```
$ expo init Youtube
```

|| ! Make sure to chose the “tabs (TypeScript)” template

- ||| 2. Run the development server

```
$ yarn start      $ npm start
```

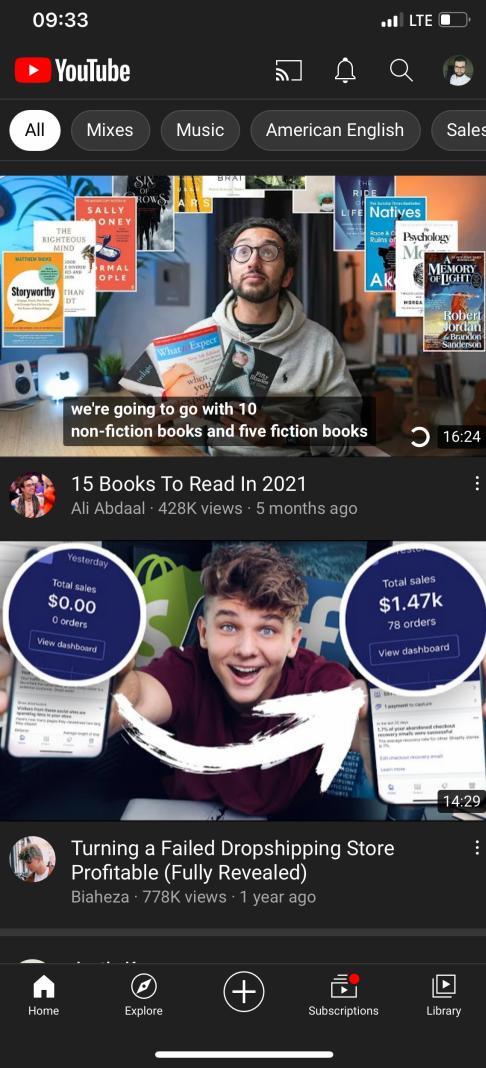
- ||| 3. Clean the project



Home Screen



- List of Videos
- Video:
 - Thumbnail
 - Time
 - Title row



Video Component



- Thumbnail
- Time
- Title row:



Avatar



Title



Subtitle (name, views, posted at)



Menu Icon



Turning a Failed Dropshipping Store
Profitable (Fully Revealed)

Biaheza · 778K views · 1 year ago

⋮



Video Page



- Video Player
- Video Details (title, views, ...)
- Video Actions
- Comments
- Video Recommendations



A screenshot of a computer screen displaying a React Native development environment. The left side shows a code editor with several files open, including 'index.js' and 'App.js'. The right side shows a mobile application running on an iPhone X simulator, displaying a simple interface with tabs and a status bar. A small video overlay in the bottom right corner shows a man with glasses and a beard, likely the developer, speaking.

A banner for Webflow, featuring the Webflow logo (a blue 'W' icon), the word 'Ad', the URL 'webflow.com', and a 'LEARN MORE' button.

#VadimSavin #notejust #notJustDevelopment

Build a Realtime Chat App in React Native
(tutorial for beginners) 🍒

327K views · Streamed 7 months ago



3.7K



68



Live chat



Share



Download



Vadim Savin
21.8K subscribers

SUBSCRIBE

Comments 303

You so good Vadim!! Amazing Project Amazing Value Amazing You!

Estos conocimientos antes solo estaban al alcance de unos pocos.

APRENDE DE LOS FUNDADORES DE



ZUCKERBERG

NETFLIX

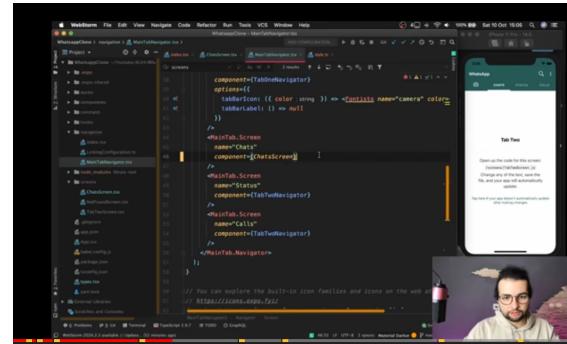


The MBA

Video Player



Render videos using Expo AV
<https://docs.expo.io/versions/latest/sdk/av/>





Bottom Sheet Modal



Official docs and installation guide:
<https://gorhom.github.io/react-native-bottom-sheet/>



A screenshot of a YouTube mobile application. At the top, the time is 13:20 and there are icons for signal strength, LTE connectivity, battery level, and notifications. The main area shows a video thumbnail for "How to Film Videos WITH YOUR IPHONE". The video title is "How to Film YouTube Videos on Your Phone" by Ali Abdal, with 302K views and posted 1 year ago. Below the video are several interaction options: "Save to Watch Later", "Save to playlist", "Share", "Not interested", "Don't recommend channel", "Report", and "Cancel". The "All" tab is selected in the navigation bar at the top.



//development

Video Comments



- List of comments in the Bottom Sheet
- Add new comment



09:36

```
import React from 'react';
import { TabBar } from '@notjustdev/react-native-tabbar';
import { colors } from '@notjustdev/react-native-theme';

const MainTabBarScreen = () => {
  return (
    <TabBar>
      <TabBarTab color="white" title="Chats" icon="chat" activeColor="black" activeIcon="chat-active" />
      <TabBarTab color="white" title="Status" icon="status" activeColor="black" activeIcon="status-active" />
      <TabBarTab color="white" title="Calls" icon="call" activeColor="black" activeIcon="call-active" />
      <TabBarTab color="white" title="Profile" icon="profile" activeColor="black" activeIcon="profile-active" />
    </TabBar>
  );
};

export default MainTabBarScreen;
```

Comments 303

Remember to keep comments respectful and to follow our [Community Guidelines](#)



Add a public comment...



Pinned by Vadim Savin

Vadim Savin · 5 months ago

Download the Asset Bundle (Images, Dummy data, PDF presentation): <https://assets.notjust.dev/whatsapp>

16



6



6 REPLIES



Nikhil Gupta · 4 months ago

You so good Vadim!! Amazing Project Amazing Value Amazing You!

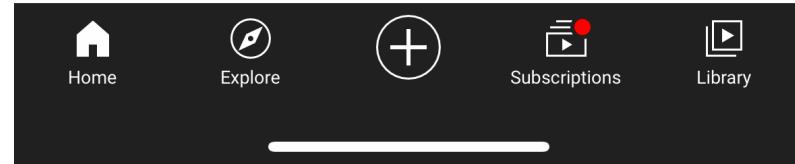
7



3



• 3 REPLIES



Bottom Tab Navigator



1. Setup the tabs with their icons



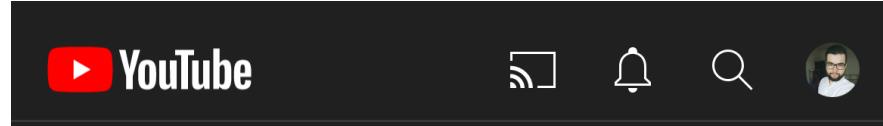
Find icons on: <https://icons.expo.fyi/>



```
import { Foundation, Ionicons, AntDesign, MaterialIcons } from '@expo/vector-icons';

<Foundation name="home" size={24} color="black" />
<Ionicons name="compass-outline" size={24} color="black" />
<AntDesign name="pluscircleo" size={24} color="black" />
<MaterialIcons name="subscriptions" size={24} color="black" />
<MaterialIcons name="video-collection" size={24} color="black" />
```





Top Header



1. Setup the header bar with their icons



Find icons on: <https://icons.expo.fyi/>



```
<Feather name="cast" size={24} color="black" />
<AntDesign name="bells" size={24} color="black" />
<AntDesign name="search1" size={24} color="black" />
<FontAwesome name="user-circle-o" size={24} color="black" />
```



notJust
//development



Q&A



notJust

//development



Next Friday: 11 June, 14:00 GMT

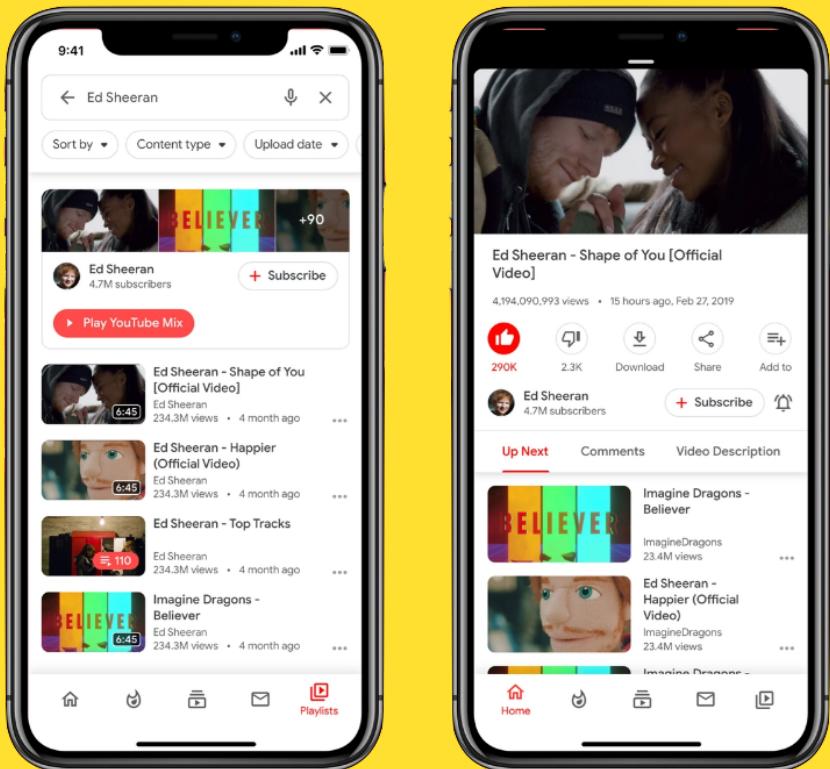
Youtube Backend

- Authentication
- API & Database
- Video Storage
- and more



notJust

//development



Youtube Backend



AWS Amplify & Admin UI



notJust

//development

Admin UI



Admin UI

YoutubePrep > staging ▾

Deploying... ⏲

The screenshot shows the Admin UI interface for the 'YoutubePrep' app in the 'staging' environment. The left sidebar has sections for 'Manage' (Home, Content, User management) and 'Set up' (Data, Authentication, Functions, Storage, GraphQL API, REST API). The main area displays a welcome message: 'Welcome back to **YoutubePrep's** **staging** environment'. Below it is a 'Things to do next' section with four items: 'Data model' (with a 'Create data model' button), 'View and edit app content' (with a 'Manage app content' button), 'Configure log in and sign up' (with a 'View authentication settings' button and a 'Deployed' status indicator), and 'Users and groups' (with a 'Manage groups' button). A decorative graphic of colored circles and stars is in the top right corner.

Welcome back to **YoutubePrep's** **staging** environment

Things to do next

Data model

Iterate on your app's data model. Create relationships between models and set up authorization rules.

View and edit app content

Use our content editor to create new records or manage existing ones.

Configure log in and sign up

Configure password-protected login for your app or leverage 3rd party authentication providers.

Users and groups

Create users or manage your existing user base. Add users to groups, edit user attributes, and suspend users.

Deploying

Create data model

Manage app content

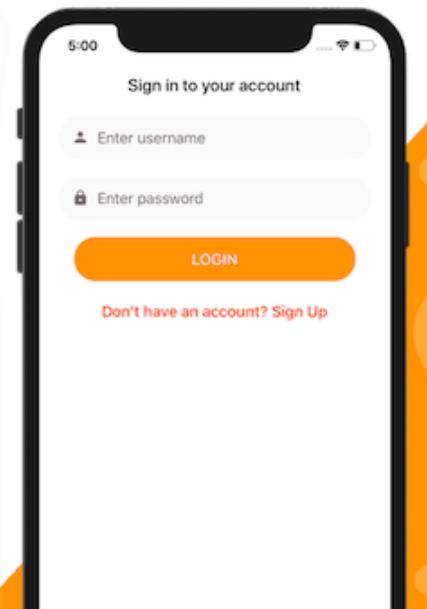
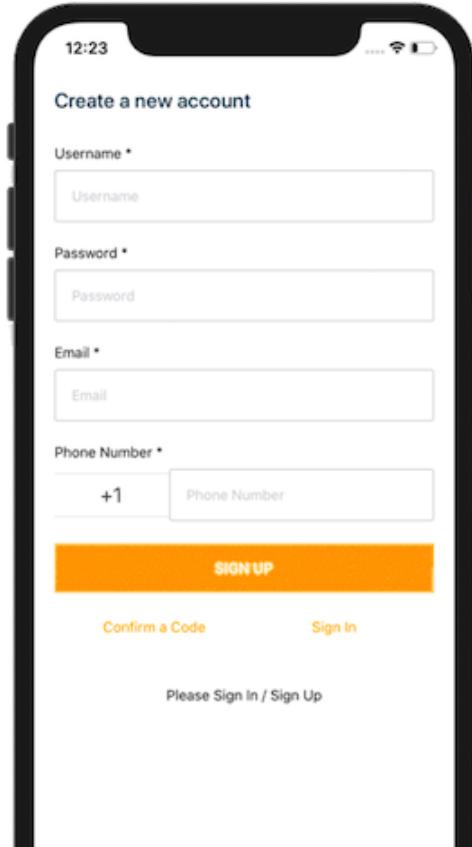
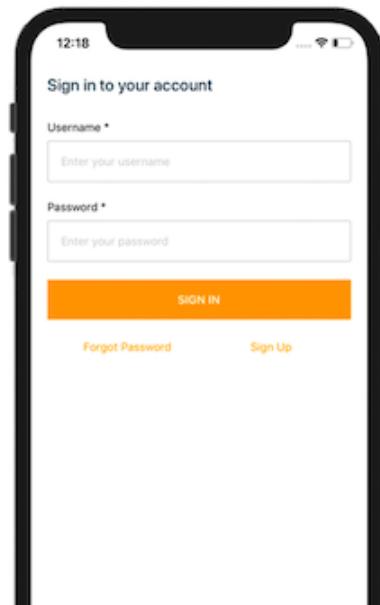
View authentication settings

Manage groups

Deployed



Authentication



Data modeling



Data modeling

🕒 Deployed View and edit your app's data models, add fields, and relationships.

Auth

Co

Add model

Save and deploy

Comment

Field name

id

Type

ID!

videoID

Relationship Source

+ Add a field

Relationship name

Video

Related to

Video

Cardinality

1:1 one Comment to one Vi...

+ Add a relationship

Video

Field name

id

Type

ID!

content

Type

String

title

Type

String

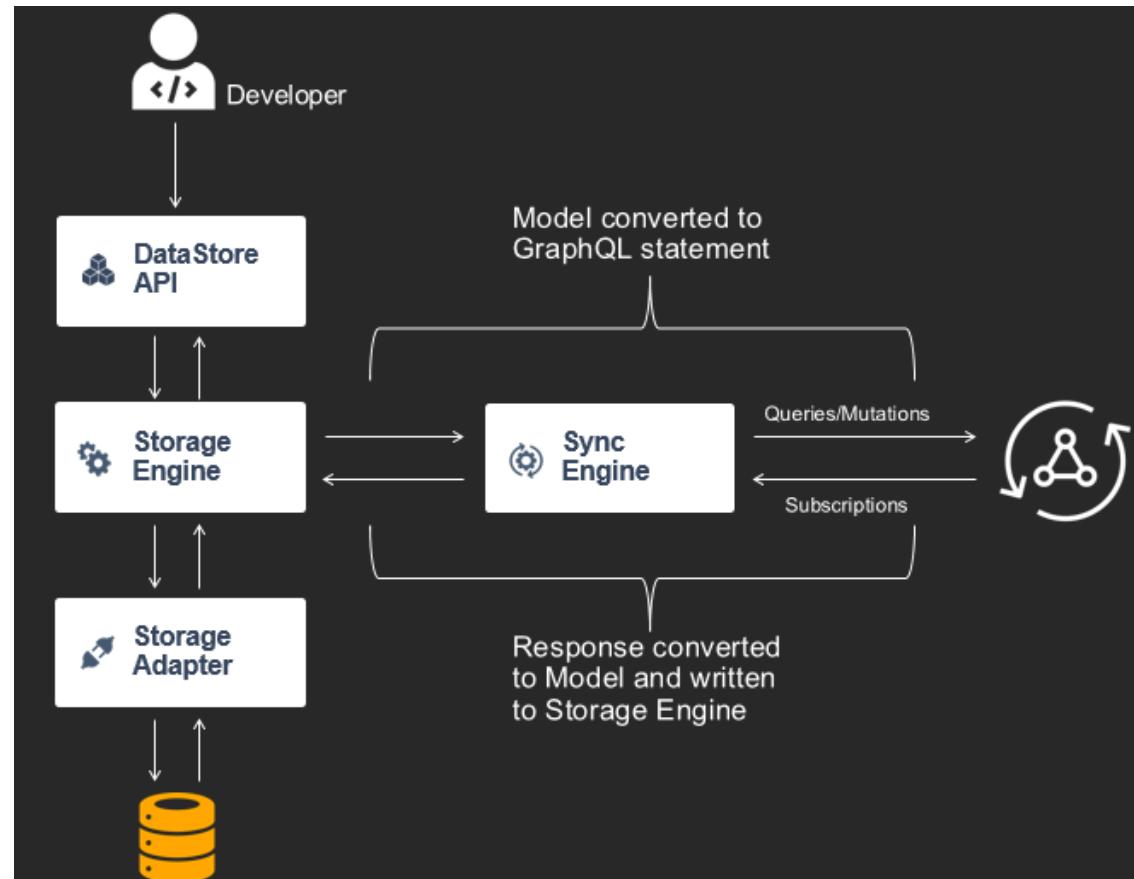
untitledfield

Type

String

Datastore

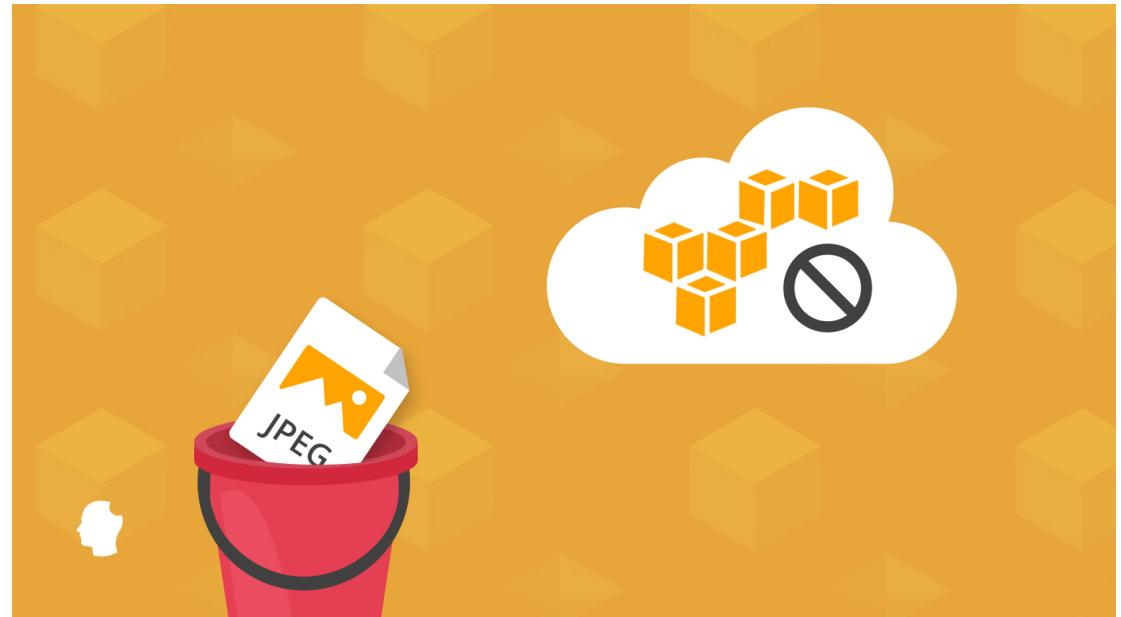
- ||| 1. Query Videos
- ||| 2. Query Comments
- ||| 3. Create Comments
- |||
- |||
- |||
- |||



Media Storage



AWS S3 storage

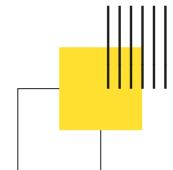


Front-end technologies



Back-end technologies

Expo



Front-end technologies



Expo



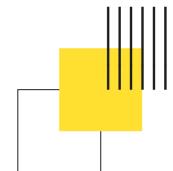
Back-end technologies

AWS Amplify

For Next-Gen
App Development



amazon
S3





Who am I



- Full stack developer for >7yrs
- CTO @Fitinium
- ex Amazon SDE
- Passionate about coding and building impactful startups.

I AM **notJust**
//developer



@VadimNotJustDev



//development

New channel members

Member

Current level



Esteban Cordero

notJust Developer Mid

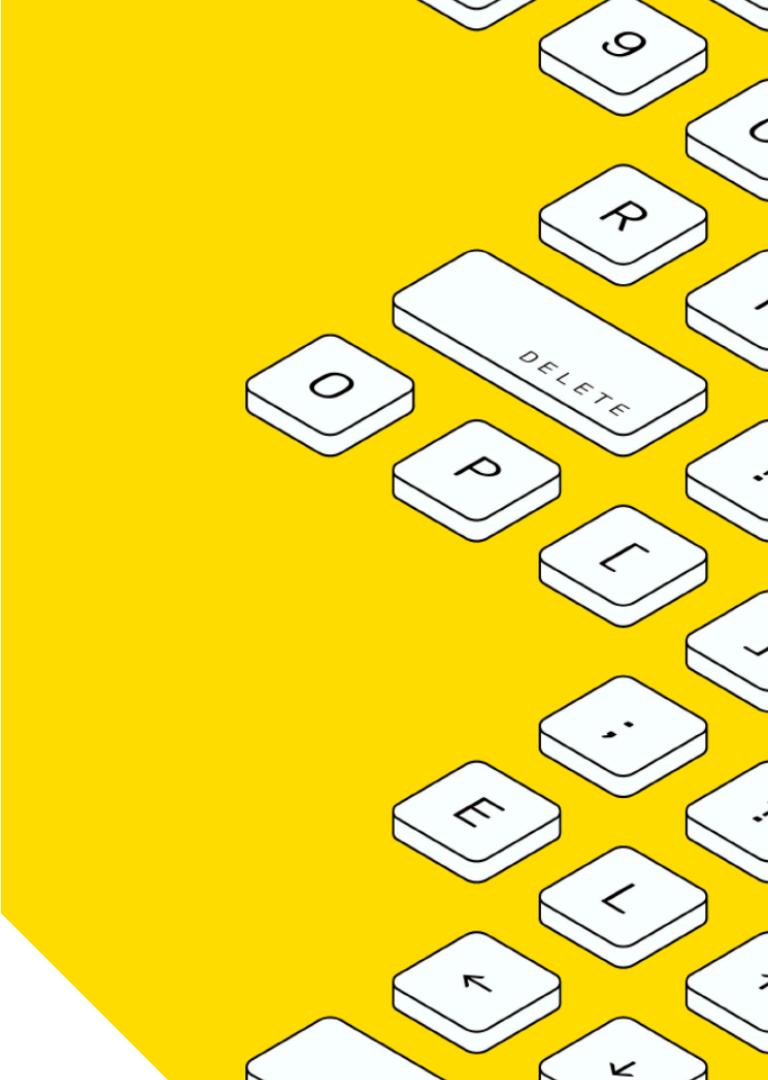


Chan KevinCTC

notJust Developer Senior



I AM notJust
//developer



Prerequisites

1. Expo environment set up
2. AWS Account
3. Dummy data, images, icons, PDF presentation, cookies:
<https://assets.notjust.dev/youtube>



https://youtu.be/CQGV_FDvS-Q

Starting point

1. Part 1





https://youtu.be/CQGV_FDvS-Q

Starting point

1. Part 1

Didn't follow part 1, but want to follow along now?

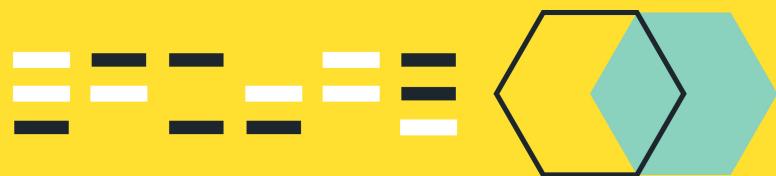
2. Clone the repo (UI branch): <https://github.com/Savinvadim1312/YoutubeClone/tree/ui>
3. Install dependencies (yarn install, npx pod-install)
4. Run the project (yarn start)

notJust

//development



Let's get started



Create Amplify Project



- Navigate to Amplify page (search on AWS Console)
- New app -> Create app backend
- Open Admin UI



Admin UI DeleteMe >

Welcome back to **DeleteMe** staging environment

Things to do next

Data model
Iterate on your app's data model. Create relationships between models and set up authorization rules.

Create data model

View and edit app content
Use our content editor to create records or manage existing ones.

Manage app content

Deployment activity

Search deployment activity

Category	Timestamp

Setup Authentication



Admin UI YoutubePrep > staging ▾

Home
Manage
Content
User management
Set up
Data
Authentication
Functions
Storage
GraphQL API
REST API
Analytics
Documentation ?
Support ?

Authentication

Deployed Configure how your users log in and sign up to your app.

1. Configure login

Add login mechanisms

Authentication has been deployed so username, email, and phone number mechanisms can't be changed.

Email

Enable your users to sign in with their email address. Email login is protected by password challenge.

Add login mechanism ▾

Multi-factor authentication

2. Configure sign up

Select which attributes you require from your customers.

Sign up settings can't be changed after authentication has been deployed

Email

Add attribute ▾

Password protection settings

Reset all authentication settings and users

Save and deploy

Data Modeling



Admin UI YoutubePrep > staging ▾ [Documentation](#) [Feedback](#)

Home

Manage

- Content
- User management

Set up

- Data**
- Authentication
- Functions
- Storage
- GraphQL API
- REST API
- Analytics

Documentation [Feedback](#)

Feedback [Documentation](#)

Data modeling

Deployed View and edit your app's data models, add fields, and relationships.

Add model Save and deploy

Comment

Field name	Type
id	ID!
videoID	Relationship Source

+ Add a field

Relationship name **Related to** **Cardinality**

Relationship name	Related to	Cardinality
Video	Video	1:1 one Comment to one Vi...

+ Add a relationship

Video

Field name	Type
id	ID!
content	String
title	String
untitledfield	String

Pull the changes



Install Amplify CLI (if you don't have it):

```
$ curl -sL https://aws-amplify.github.io/amplify-cli/install | bash && $SHELL
```



Pull the backend into React Native app:

```
$ amplify pull --appId APP_ID --envName staging
```



Configure Amplify



In App.tsx:



```
import Amplify from "aws-amplify";
import config from "./src/aws-exports";
Amplify.configure(config);
```



Configure Auth

In App.tsx:

```
import { withAuthenticator } from "aws-amplify-react-native";  
...  
export default withAuthenticator(App);
```

Integrate Datastore



- Fetch videos
- Fetch comments
- Add comment



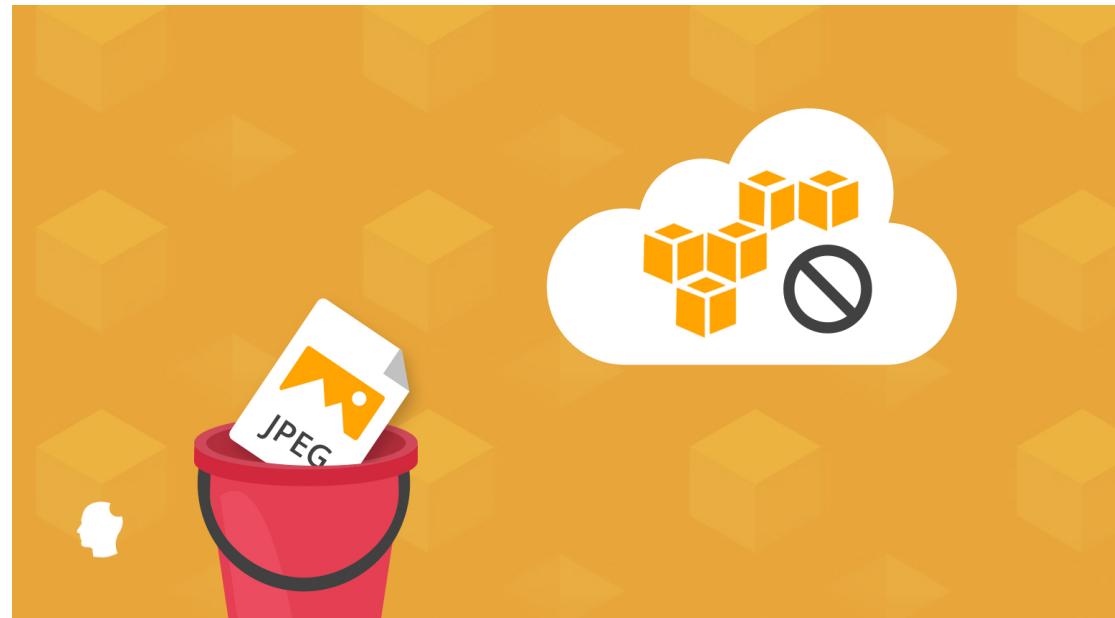
Next Week connect comments <> users



Media Storage



AWS S3 storage



notJust
//development



Q&A



Troubleshooting

```
$ amplify pull --appId dpgg4504xtt7g --envName staging
\Scanning for plugins...
Plugin scan successful
^R
Opening link: https://eu-west-1.admin.amplifyapp.com/admin/dpgg4504xtt7g/staging/verify/
✓ Successfully received Amplify Admin tokens.
Amplify AppID found: dpgg4504xtt7g. Amplify App name is: YoutubePrep
Backend environment staging found in Amplify Console app: YoutubePrep
Failed to pull the backend.
Invalid feature flag configuration

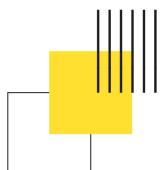
These feature flags are defined in the "amplify/cli.json" configuration file and are unknown to the currently running Amplify CLI:
- auth.breakcirculardependency
- codegen.addtimestampfields
- codegen.handlelistnullabilitytransparently

This issue likely happens when the project has been pushed with a newer version of Amplify CLI, try updating to a newer version.

Learn more about feature flags: https://docs.amplify.aws/cli/reference/feature-flags
```

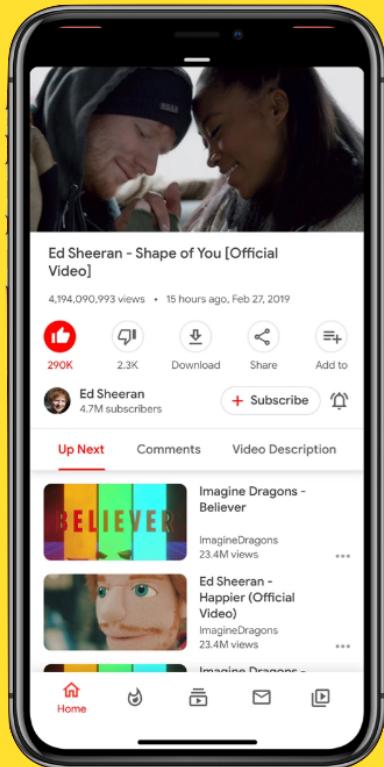
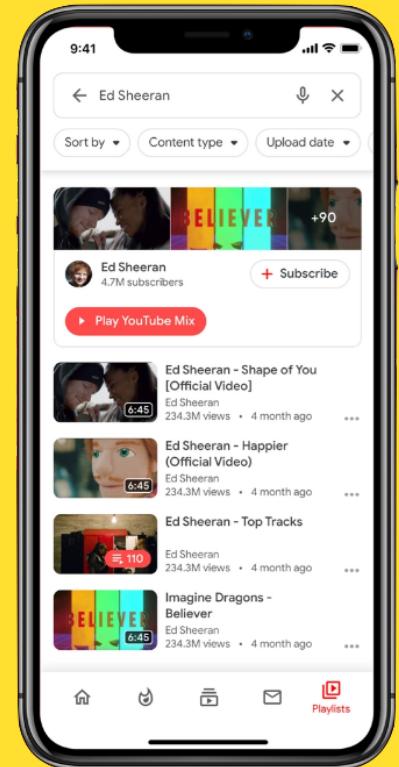
Solution:

```
npm install -g @aws-amplify/cli
```



notJust

//development



Youtube Backend 2

AWS Amplify & Admin UI



In this video:

- Save User in database



User X

Field name	Type
id	ID!
name	String!
image	String
subscribers	Int

[+ Add a field](#)

Relationship name	Related to	Cardinality
Videos	Video	1:n one User ... ▾

[+ Add a relationship](#)

In this video:

- Save User in database
- User <> Comment relationship

The screenshot shows a database schema editor interface with two entities: **Comment** and **User**.

Comment Entity Fields:

Field name	Type
id	ID!
comment	String!
likes	Int!
dislikes	Int!
replies	Int!

Add a field

User Entity Fields:

Field name	Type
id	ID!
name	String!
image	String
subscribers	Int

Add a relationship

Relationships:

Relationship name	Related to	Cardinality
User	User	1:1 one Com...

In this video:



- Save User in database
- User <> Comment relationship
- Upload Video
 - Design the Upload screen UI
 - Select Video from device
- Add Storage provider (S3)
- Save video in storage
- Generate thumbnails
- Save Video post to database



The image shows the AWS Amplify Storage landing page. It features the AWS logo (a yellow triangle) and the text "AWS Amplify" in large, bold, dark blue letters. Below that is the word "Storage". To the right, there is an illustration of two people interacting with a large smartphone. One person is sitting on a chair, holding a laptop and pointing at the phone's screen, which displays a user interface with icons like a map, a person, and a location pin. Another person is standing next to them, also pointing at the phone. The background is light gray.

In this video:

- Save User in database
- User <> Comment relationship
- Upload Video
 - Design the Upload screen UI
 - Select Video from device
 - Add Storage provider (S3)
 - Save video in storage
 - Generate thumbnails
 - Save Video post to database
- * Bonus feature





Who am I



- Full stack developer for >7yrs
- CTO @Fitinium
- ex Amazon SDE
- Passionate about coding and building impactful startups.



I AM **notJust**
//developer



@VadimNotJustDev



New channel members

Member



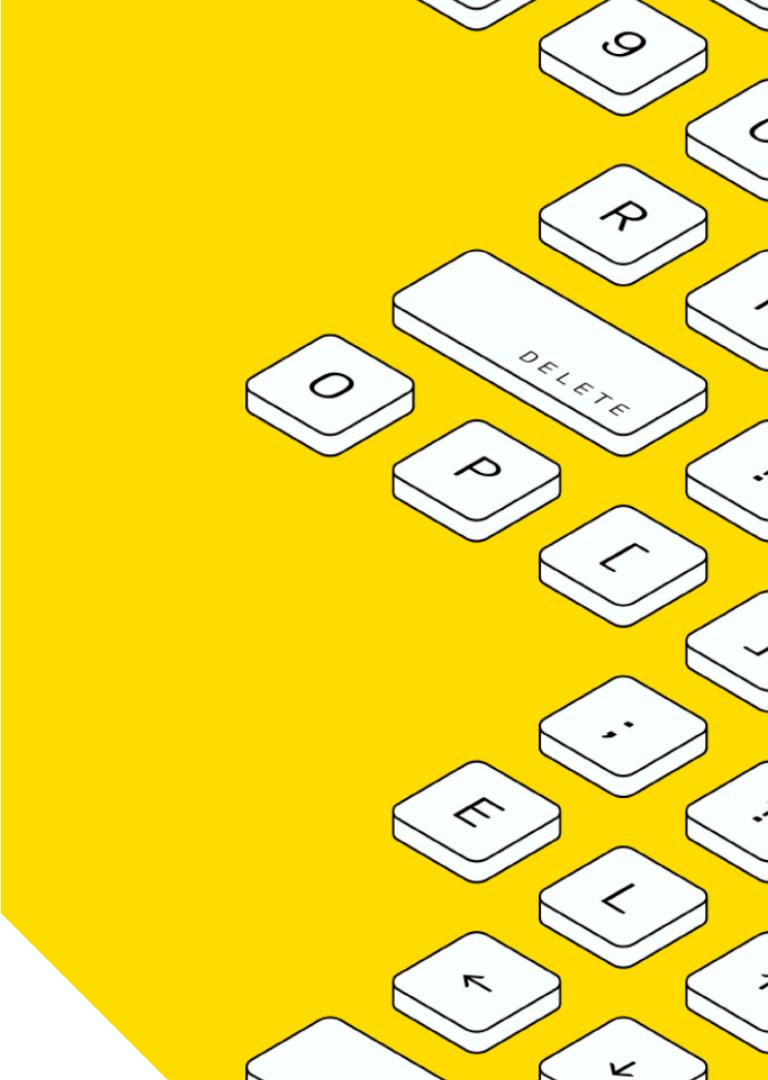
Kester Belgrove

Current level

notJust Developer Junior



I AM notJust
//developer

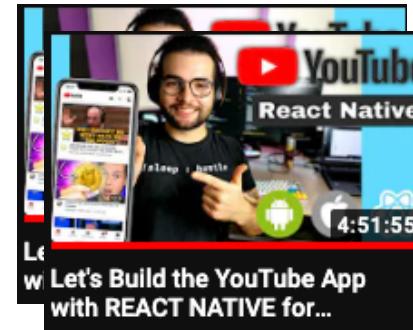


Prerequisites

1. Expo environment set up
2. AWS Account
3. Dummy data, images, icons, PDF presentation, cookies:
<https://assets.notjust.dev/youtube>

Starting point

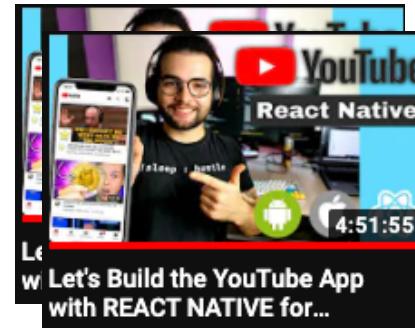
1. Part 1 & 2



[https://www.youtube.com/
playlist?
list=PLY3ncAV1dSVBT_4wTi4E7
4anpg7dgAJWI](https://www.youtube.com/playlist?list=PLY3ncAV1dSVBT_4wTi4E74anpg7dgAJWI)

Starting point

1. Part 1 & 2



[https://www.youtube.com/playlist?
list=PLY3ncAV1dSVBT_4wTi4E7
4anpg7dgAJWI](https://www.youtube.com/playlist?list=PLY3ncAV1dSVBT_4wTi4E74anpg7dgAJWI)

- ||| Didn't follow part 1&2, but want to follow along now?
2. Clone the repo (UI branch): <https://github.com/Savinvadim1312/YoutubeClone/tree/part-2>
 3. Install dependencies (yarn install, npx pod-install)
 4. Init Amplify (amplify init) then push to AWS cloud (amplify push)
 5. Run the project (yarn start)
- |||
- |||
- |||

notJust

//development



Let's get started



Save User in database

Every Authenticated user (from Cognito) should be mapped to one User in database



User

Field name	Type
id	ID!
name	String!
image	String
subscribers	Int

+ Add a field

Relationship name	Related to	Cardinality
Videos	Video	1:n one User ...

+ Add a relationship

User - Comment relationship

The screenshot shows a data modeling interface with two entities: Comment and User.

Comment Entity Fields:

Field name	Type
id	Id!
comment	String!
likes	Int!
dislikes	Int!
replies	Int!

Add a field

User Entity Fields:

Field name	Type
id	Id!
name	String!
image	String
subscribers	Int

Add a relationship

Relationship settings:

Relationship name	Related to	Cardinality
User	User	1:1 one Com...

1. Add relationship in data modelling
2. Save user when creating a comment
3. Fetch and display the user owner of the comment



Upload Videos



1. Design the Upload screen UI
2. Select a video from device
[ImagePicker - Expo Documentation](#)
3. Add Storage
4. Save video in storage
[Storage - Upload files - Amplify Docs](#)
5. Generate thumbnails
6. Insert Video Post in database

The screenshot shows a database schema editor with two entities defined:

- Comment Entity:** Contains fields: id (ID), comment (String), likes (Int), dislikes (Int), and replies (Int). It has a relationship named "User" related to another "User" entity with a cardinality of "1:1 one Com...".
- User Entity:** Contains fields: id (ID), name (String), image (String), and subscribers (Int).

Analytics

The Analytics category enables you to collect analytics data for your App.



Goal

To setup and configure your application with Amplify Analytics and record an analytics event.

<https://docs.amplify.aws/lib/analytics/getting-started/q/platform/js>

notJust
//development



Q&A

