



mru.memes

...

\*opens lecture slides\*

“Slide 1/97”



Liked by aruqi\_\_ and others

mru.memes when ur an hour and a half into the lecture  
and the prof does a quick “let’s see what we have left  
here” check

[View all 379 comments](#)

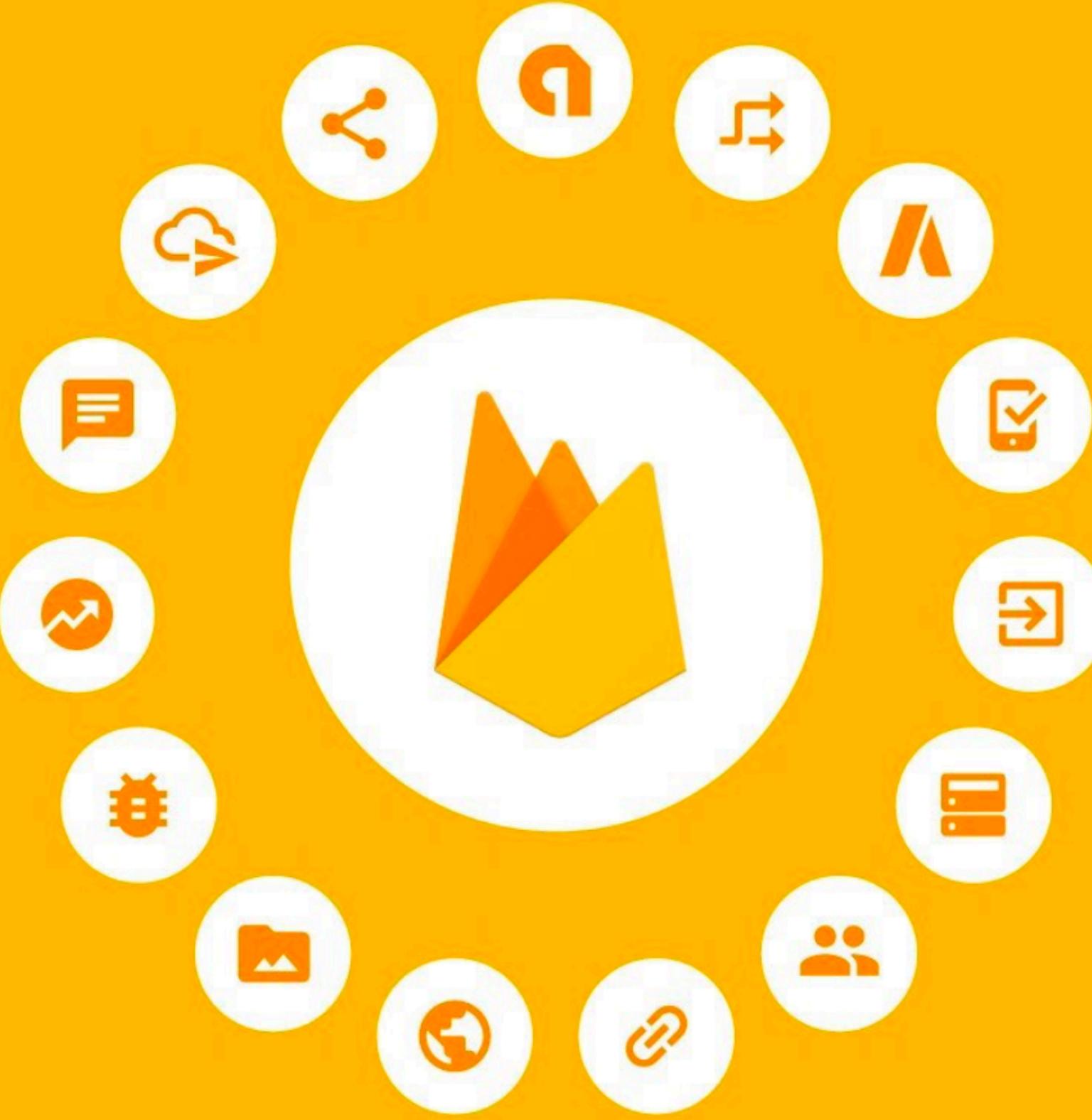
16 February

Sorry, 29 slides today only 



# FIREBASE & APP DEV

APP Development



# Agenda

From REST to Realtime DB SDK

Cloud Storage

Authentication & User Management

Push Notifications

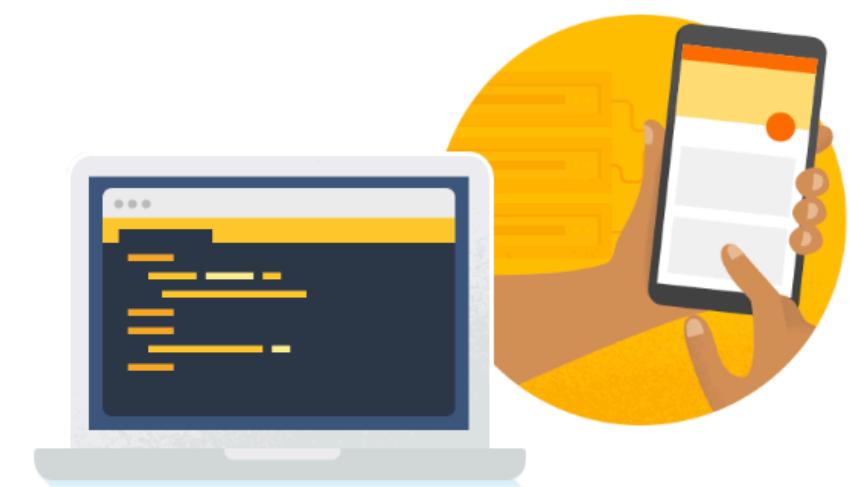
Ionic Project & Exam

Next Steps



# What is Firebase?

Platform & Backend-as-a-Service  
for Web & App Development



## Build better apps



### Cloud Firestore

Store and sync app data at global scale



### ML Kit BETA

Machine learning for mobile developers



### Cloud Functions

Run mobile backend code without managing servers



### Authentication

Authenticate users simply and securely



### Hosting

Deliver web app assets with speed and security



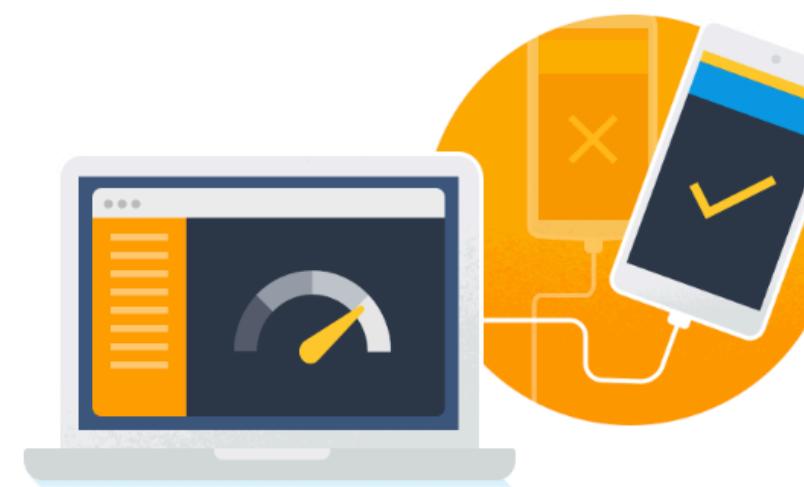
### Cloud Storage

Store and serve files at Google scale



### Realtime Database

Store and sync app data in milliseconds



## Improve app quality



### Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



### Performance Monitoring

Gain insight into your app's performance



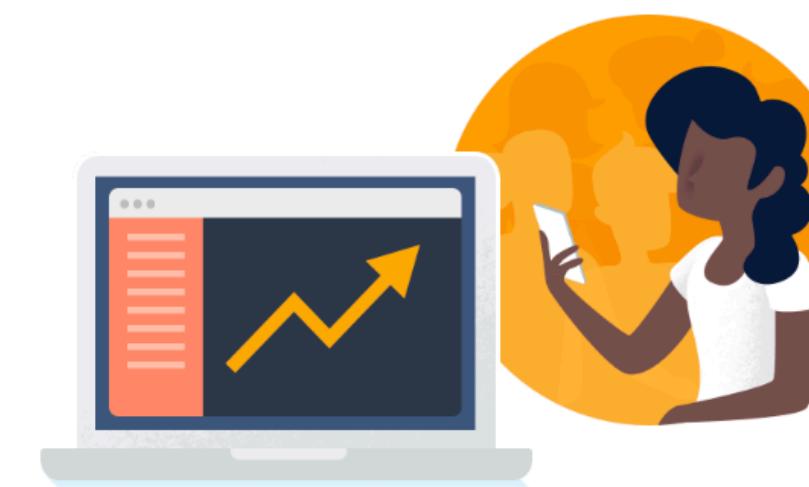
### Test Lab

Test your app on devices hosted by Google



### App Distribution BETA

Distribute pre-release versions of your app to your trusted testers



## Grow your business



### In-App Messaging BETA

Engage active app users with contextual messages



### Google Analytics

Get free and unlimited app analytics



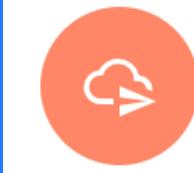
### Predictions

Smart user segmentation based on predicted behavior



### A/B Testing BETA

Optimize your app experience through experimentation



### Cloud Messaging

Send targeted messages and notifications



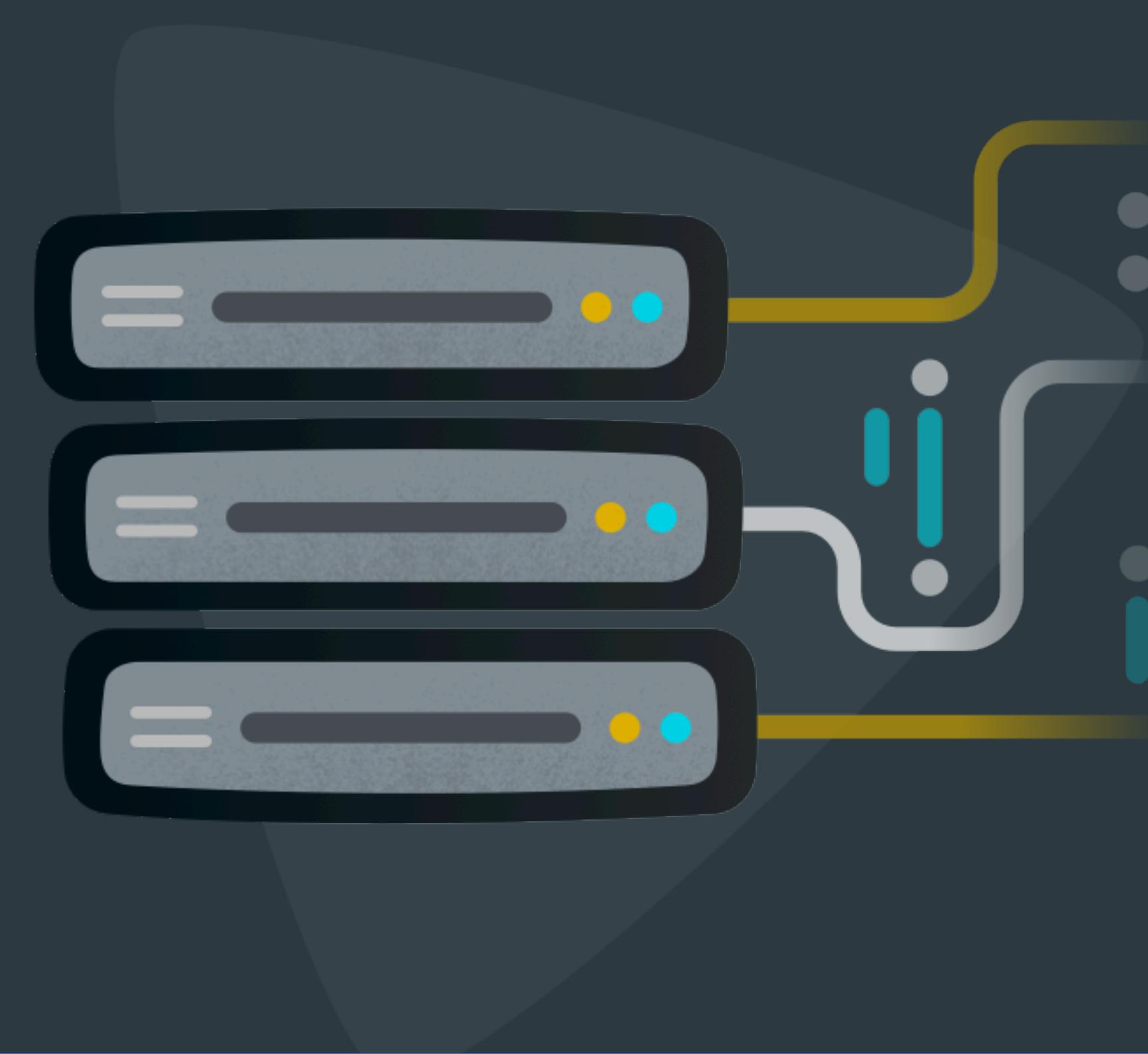
### Remote Config

Modify your app without deploying a new version



### Dynamic Links

Drive growth by using deep links with attribution



# Realtime Database & REST API

Store and sync data in real time  
REST API or SDK

<https://firebase.google.com/products/realtime-database>

# Realtime Database REST API

[https://firebase.google.com/docs/  
database/rest/start](https://firebase.google.com/docs/database/rest/start)

```
export default function PostsPage() {
  const [posts, setPosts] = useState([]);
  const [showLoader, dismissLoader] = useIonLoading();

  async function getPosts() {
    const response = await fetch("https://race-rest-default-rtbd.firebaseio.com/posts.json");
    const data = await response.json();
    // map object into an array with objects
    const postsArray = Object.keys(data).map(key => ({ id: key, ...data[key] }));
    return postsArray;
  }

  async function getUsers() {
    const response = await fetch("https://race-rest-default-rtbd.firebaseio.com/users.json");
    const data = await response.json();
    // map object into an array with objects
    const users = Object.keys(data).map(key => ({ id: key, ...data[key] }));
    return users;
  }
}
```

# Individual

ionic-post-app-firebase-rest-v6

1. Run the App and check out the project and code.
2. Focus on /components/ and /pages/

# In pairs

ionic-post-app-firebase-rest-v6

1. What are the differences from the projects we worked on last week?
2. In order to keep data synced and updated, what is the challenge when using a REST API?

# Round The Table

ionic-post-app-firebase-rest-v6

1. Discuss your findings.
2. Note questions & comments.



# Database

# Ionic Post App w/ Realtime Database

Step Guide

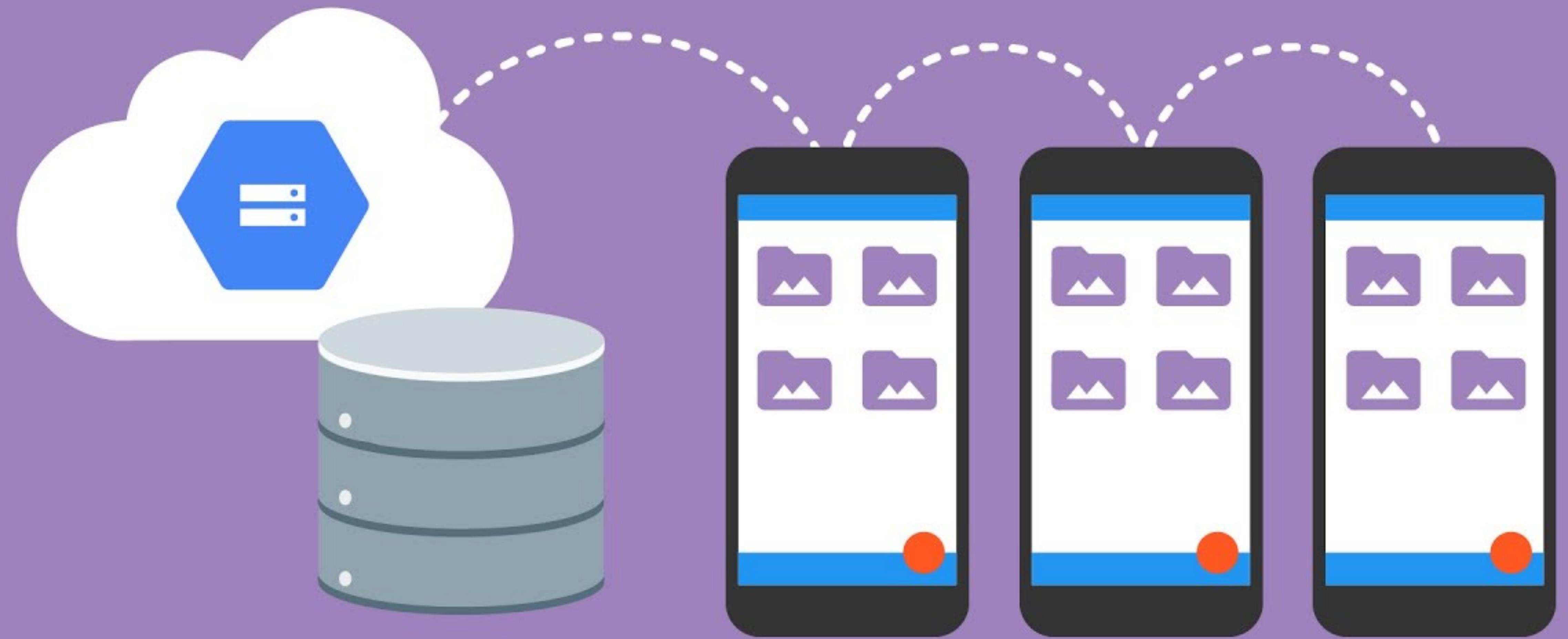
Implement the Realtime Database SDK



# Cloud Storage

Store and serve content, images, files, etc.

<https://firebase.google.com/products/storage>



# Storage

# Ionic Post App w/ Cloud Storage

Step Guide

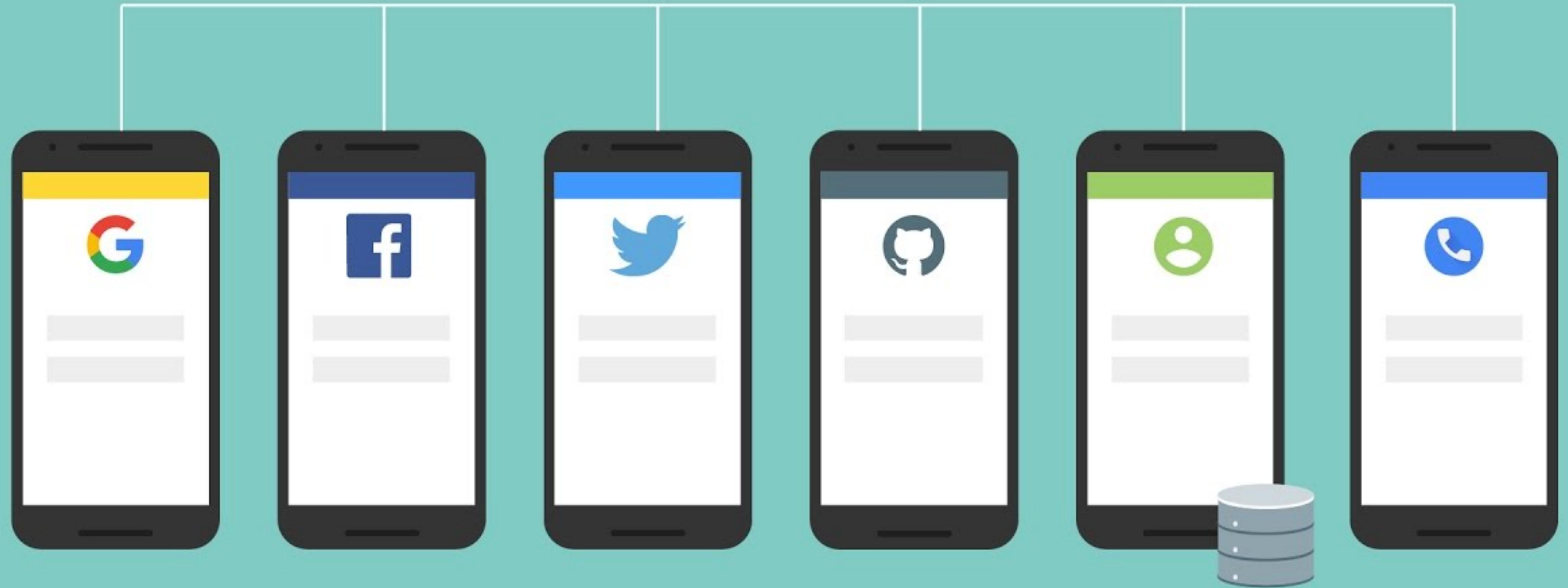
Implement the Cloud Storage SDK



# Firebase Authentication

Sign Up, Sign In and authenticate users

<https://firebase.google.com/products/auth>

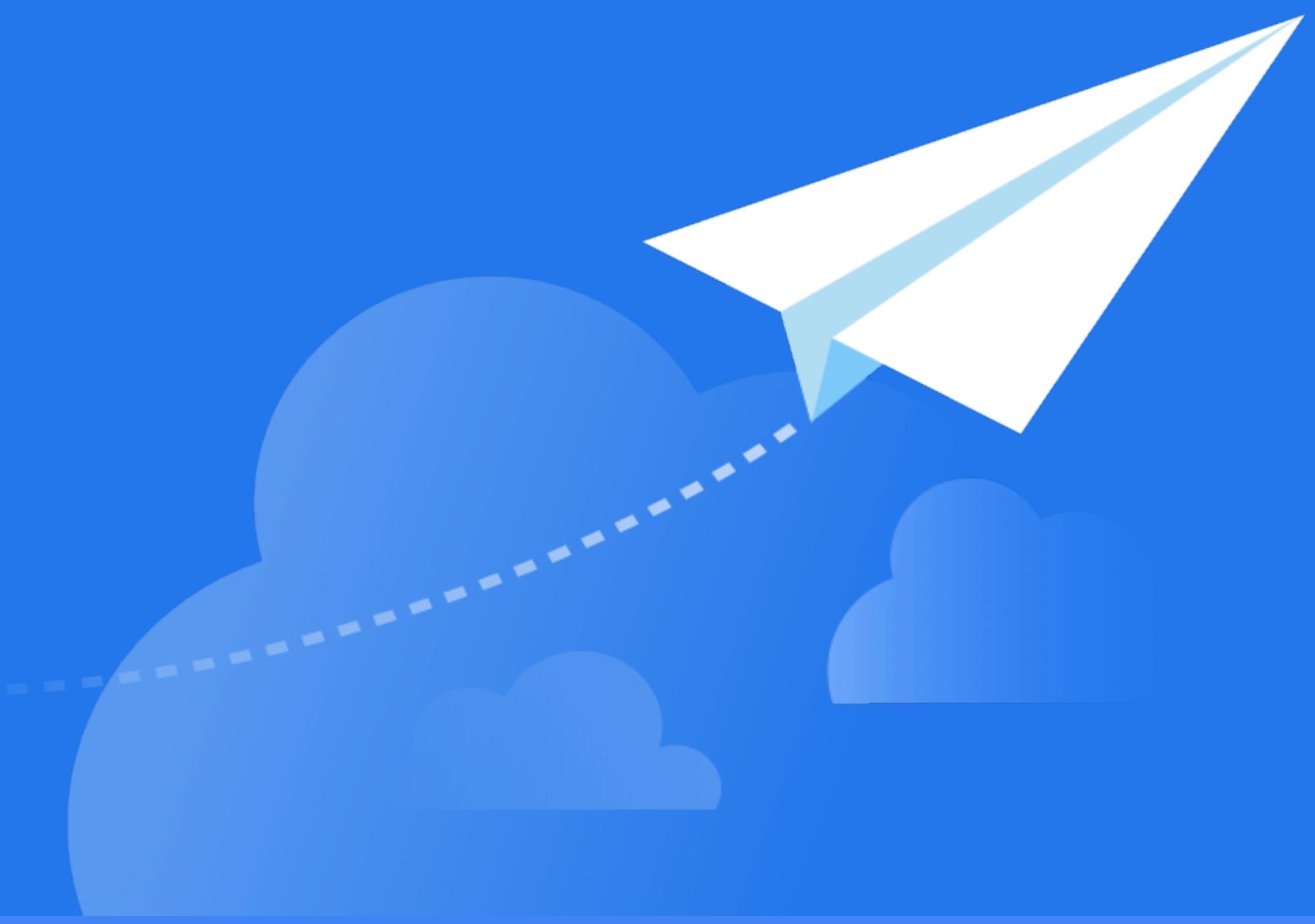


# Authentication

# Ionic Post App w/ Authentication

Step Guide

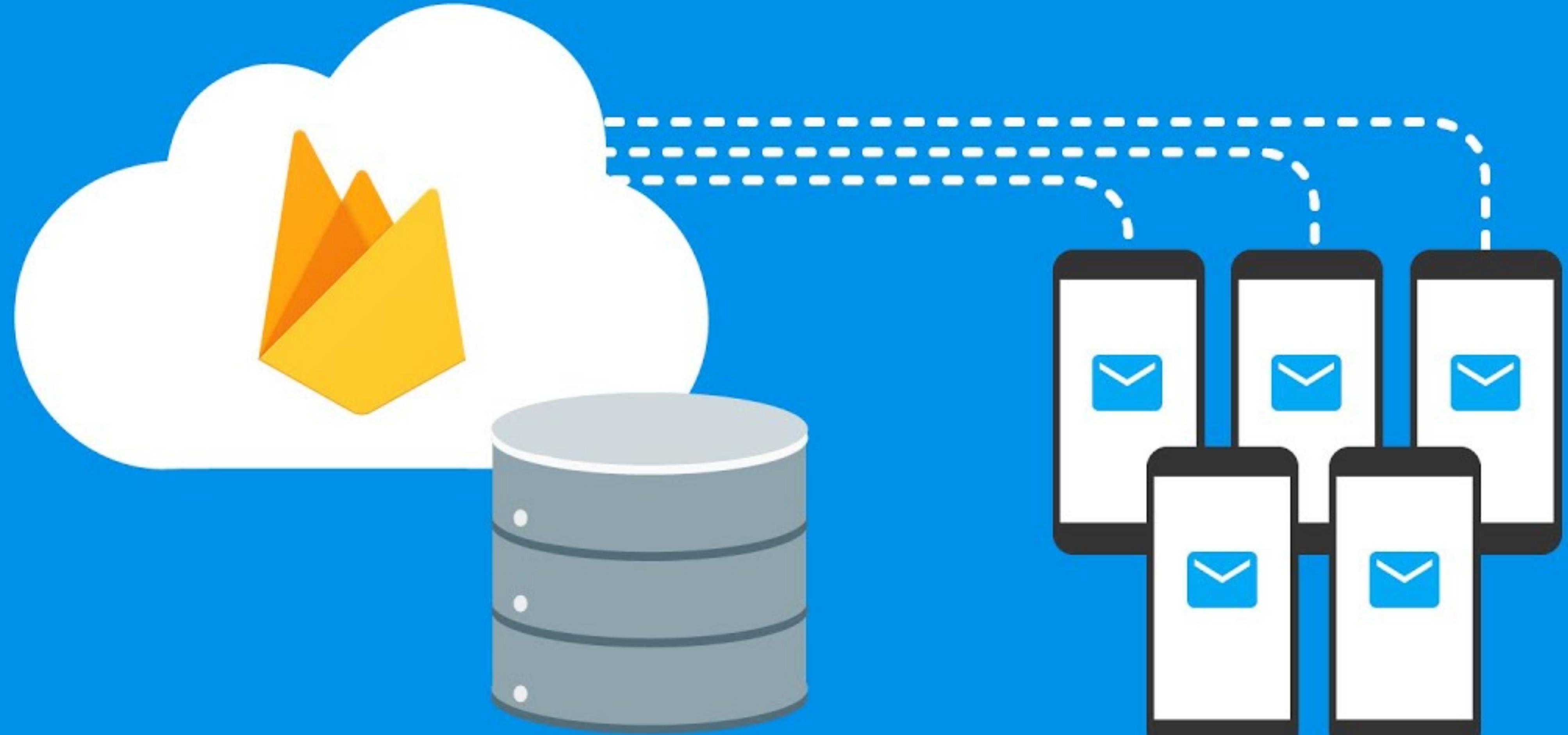
ionic-post-app-realtime-db-storage-auth



# Cloud Messaging

Send notifications across platforms

<https://firebase.google.com/products/cloud-messaging>



# Cloud Messaging

A screenshot of a web browser displaying the Capacitor Push Notifications API documentation. The URL in the address bar is <https://capacitorjs.com/docs/apis/push-notifications>. The page is part of the Ionic Capacitor Docs, version 3.

The left sidebar shows a navigation tree:

- > Overview
- ▽ APIs
  - Action Sheet
  - App
  - App Launcher
  - Browser
  - Camera
  - Clipboard
  - Device
  - Dialog
  - Filesystem
  - Geolocation
  - Haptics
  - Keyboard
  - Local Notifications

The main content area features a large heading:

# @capacitor/push-notifications

The Push Notifications API provides access to native push notifications.

## Install

```
npm install @capacitor/push-notifications
npx cap sync
```

On the right side, there is a sidebar titled "CONTENTS" listing various API reference topics:

- Install
- iOS
- Android
- Variables
- Push Notifications icon
- Push notifications appearance in foreground
- Examples
- Silent Push Notifications / Data-only Notifications
  - iOS
  - Android
- Common Issues
- Example
- API
  - register()
  - getDeliveredNotifications(..)

<https://capacitorjs.com/docs/apis/push-notifications>

The screenshot shows a web browser window with the title "Push Notifications with Firebase" at the top. The URL in the address bar is <https://capacitorjs.com/docs/guides/push-notifications-firebase>. The page is part of the Capacitor.js documentation, specifically the "Push Notifications with Firebase" guide.

The left sidebar contains a navigation menu:

- > Getting Started
- > Basics
- > Upgrade Guides
- > Cordova/PhoneGap
- ▼ Concepts
  - Ads
  - Angular
  - Automated Configuration
  - CI/CD
  - Deep Links
  - Deploying and Updating
  - Environment Specific Configurations
  - In App Purchases
  - Live Reload
  - Mocking Plugins
  - Push Notifications - Firebase**
  - React Hooks
  - Screen Orientation
  - Security

The main content area features a large heading: **Using Push Notifications with Firebase in an Ionic + Angular App**.

Below the heading, there's a section titled "Web Framework: Angular Platforms: iOS, Android".

The text explains: "One of the most common features provided by application developers to their users is push notifications. In this tutorial, we'll walk through all the steps needed to get [Firebase Cloud Messaging](#) working on iOS and Android."

Further down, it says: "For the purposes of registering and monitoring for push notifications from Firebase, we'll make use of the [Push Notification API for Capacitor](#) in an Ionic + Angular application."

A section titled "Required Dependencies" is visible at the bottom of the main content area.

The right sidebar is titled "CONTENTS" and lists several sections:

- Required Dependencies
- Prepare an Ionic Capacitor App
- Building the App & Adding Platforms
- Using the Capacitor Push Notification API
- Creating a Project for your App on Firebase
- Android
  - Integrating Firebase with the Android app
  - Download and Use the google-services.json file
- iOS
  - Prerequisites
  - Integrating Firebase with our native iOS app
  - Add the GoogleService-Info.plist file to your iOS app
  - Add the Firebase SDK via CocoaPods
  - Update the Project
  - Add Initialization Code

<https://capacitorjs.com/docs/guides/push-notifications-firebase>

ionic-push-notifications

# Exam

Individual written assignment based on a project & product you have worked on for approx. 7 weeks.  
The code must be handed in via GitHub along with a report of max 15 pages incl. illustrations.

# Course Overview

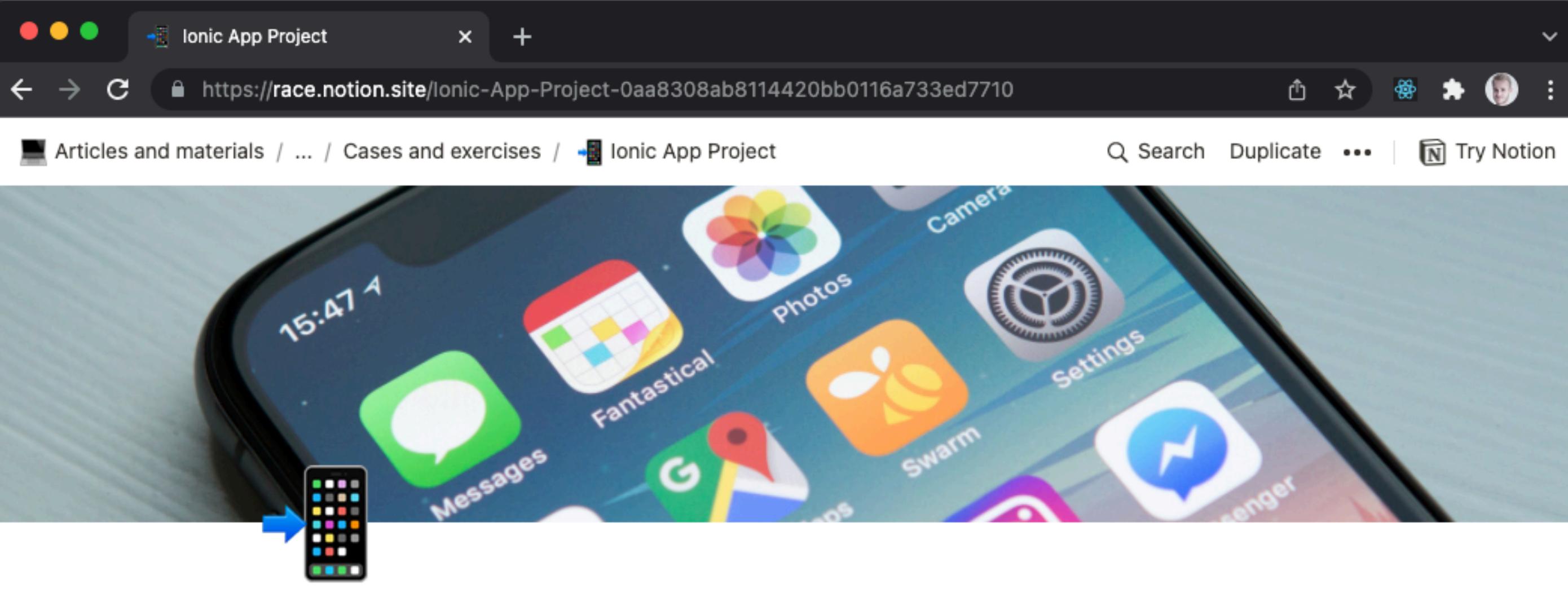
Date	Name	Course	Teacher
03/02/2022	1. Introduction to Mobile App Dev	MAD	RACE
10/02/2022	2. Web Native, Tools & UI Components	MAD	RACE
17/02/2022	3. Native Device Features	MAD	RACE
24/02/2022	4. Storage & Native APIs	MAD	RACE
03/03/2022	5. Firebase & App Dev	MAD	RACE
10/03/2022	6. Ionic App Project	MAD	RACE
17/03/2022	7. Ionic App Project	MAD	RACE
21/03/2022	8. Introduction to Mobile App Dev with Unity	MAD	KATO
28/03/2022	9. Unity	MAD	KATO
04/04/2022	10. Unity	MAD	KATO
25/04/2022	11. Unity	MAD	KATO
02/05/2022	12. Unity	MAD	KATO
09/05/2022	13. Course Wrap Up	MAD	KATO RACE

# Two Mobile App Project

During the Mobile App Dev Course, you will be introduced to two approaches of Mobile App Development (Ionic React and Unity), finalised by a project.

The projects forms the basis of your exam in Mobile App Dev and must be handed in, in order to pass the course and participate in the exam at the end of the semester.

Comparing the different approaches, will be a part of the final individual written exam. Read more about the exam in the curriculum.



# Ionic App Project

Develop a Mobile App based on themes, concept and terms from the first part of the Mobile App Dev Course.

The Mobile App is developed in groups of 2-3 students. The project is based on a self-chosen theme and must meet the below technical requirements, demonstrating your learned skills and competencies from the course.

There are no requirements in addition to research and project management but feel free to use known project tools. However the main purpose is to practice programming and development of mobile apps with React, Ionic React and Capacitor JS.

The project is finalised by handing in a link to a GitHub repository including a README.md file with a short description of your app.

[Ionic App Project Description](#)

# Next Steps

Practice your skills and competencies  
with an Ionic App Project

Deploying a PWA

Deploying a Desktop App

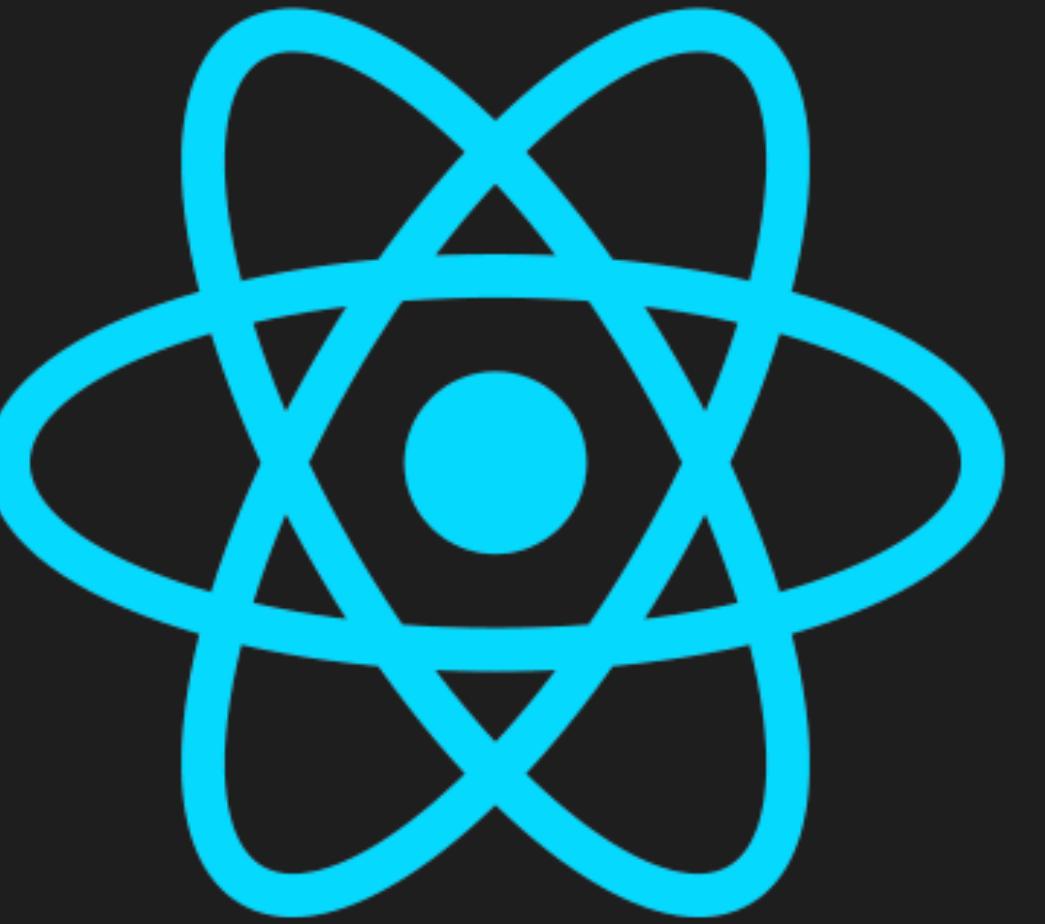
Publish an App through Google Play  
and/or App Store

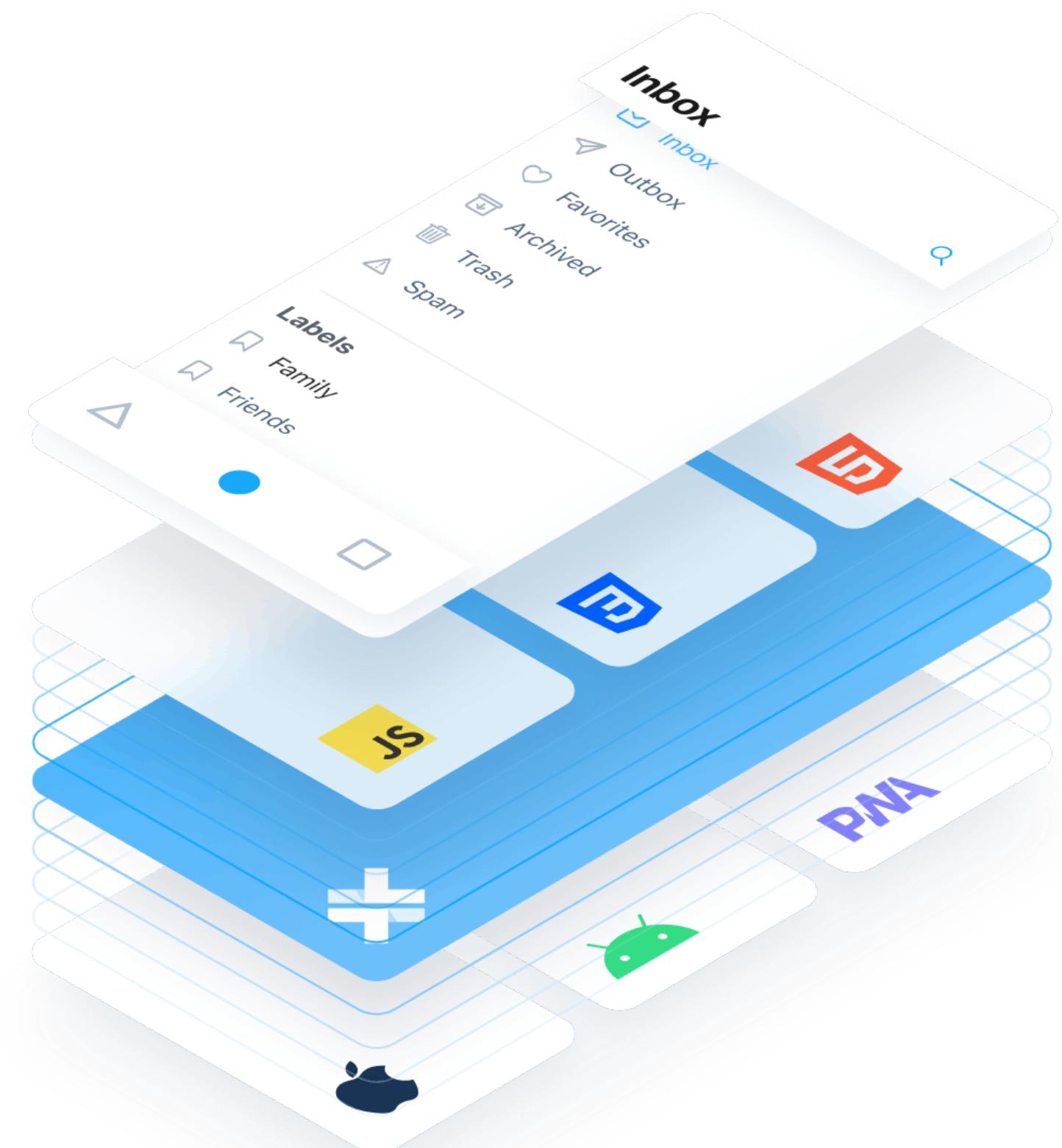
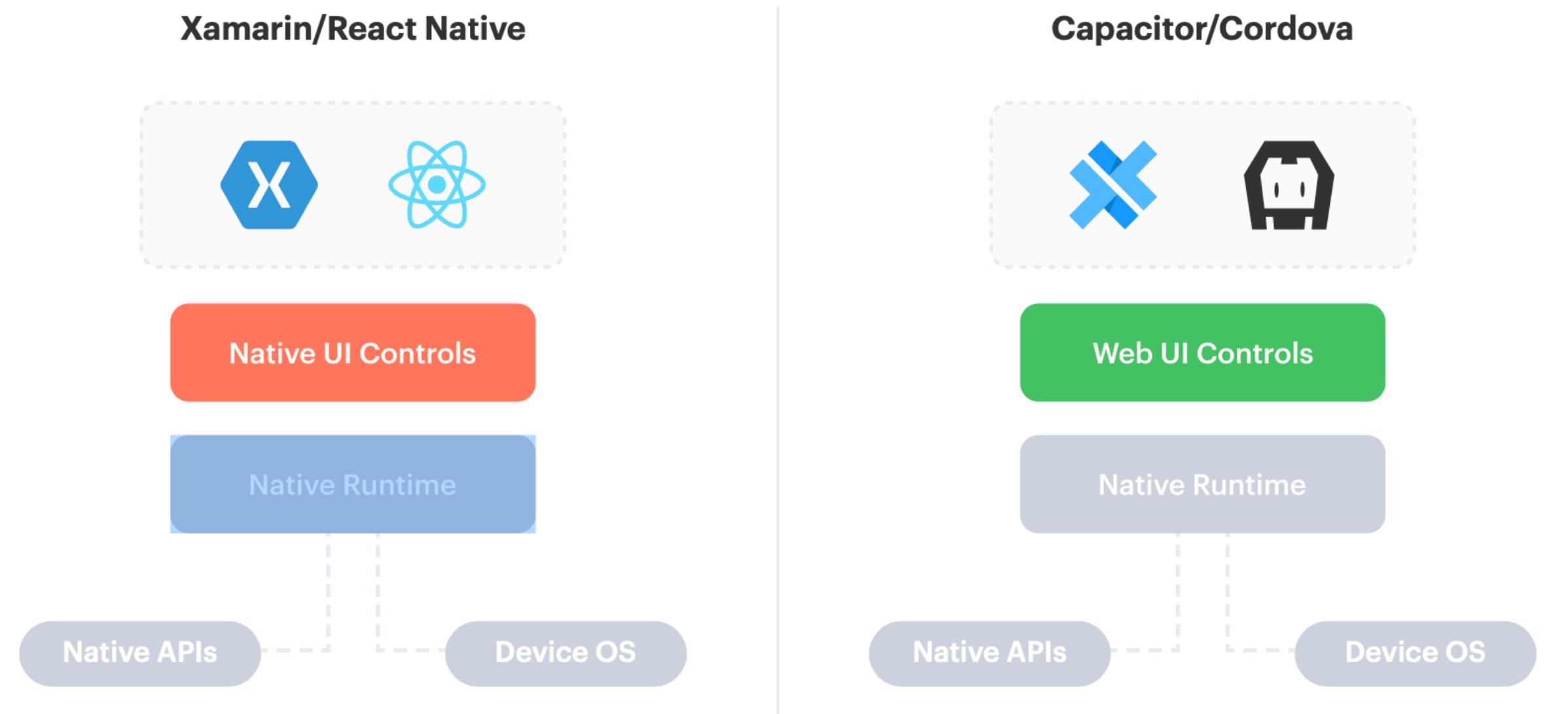
AppFlow

Games with Ionic and Phaser: @ionic-phaser/react - npm

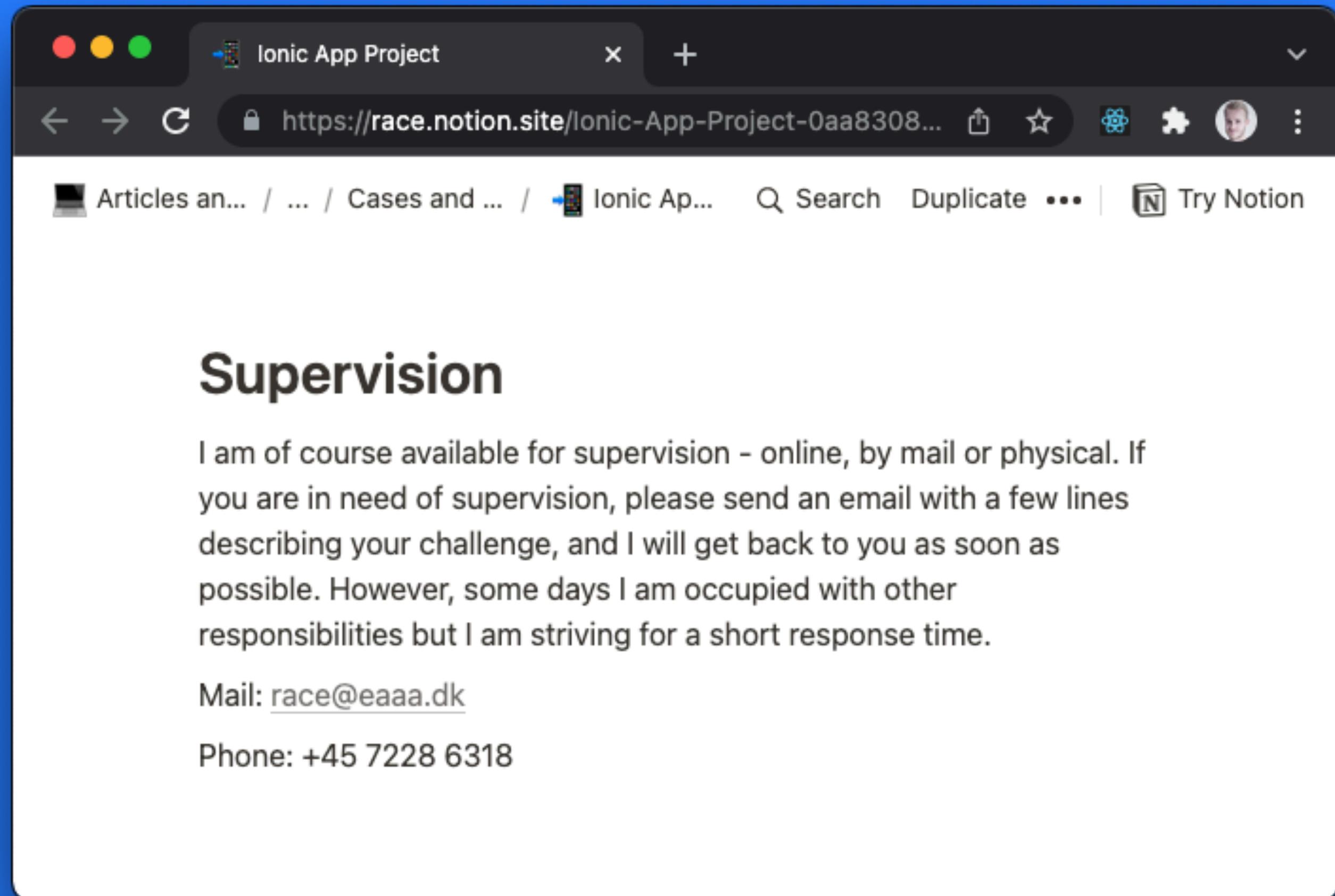
React Native

... Unity





# Supervision 🤝



A screenshot of a web browser window titled "Ionic App Project". The address bar shows the URL <https://race.notion.site/Ionic-App-Project-0aa8308...>. The page content is a Notion page titled "Supervision". The text on the page reads:

I am of course available for supervision - online, by mail or physical. If you are in need of supervision, please send an email with a few lines describing your challenge, and I will get back to you as soon as possible. However, some days I am occupied with other responsibilities but I am striving for a short response time.

Mail: [race@eaaa.dk](mailto:race@eaaa.dk)

Phone: +45 7228 6318