

Cross-Platform App Development

Final Exam

Academic Integrity

- This is an individual assessment.
- Permitted activities: Usage of Internet to search for syntax only; usage of course materials
- Not permitted:
 - Communication with others (both inside and outside the class)
 - Discussion of solution or approaches with others; sharing/using a “reference” from someone
 - Searching the internet for full or partial solutions
 - Sharing of resources, including links, computers, accounts

Evaluation Objectives

1. Programming skills using ReactNative and JavaScript, modular code design and organization
2. App UI design capabilities
3. Ability to select most suitable data structures for storing values
4. Problem solving abilities, logical and analytical skills

Submission Checklist:

- Remove the node_modules folder from your project
- Upload the **zip of your entire project folder** named as **YOURFIRSTNAME_Videos** (such as **John_Videos**) in the drop box.
- **A screen-recording** of your app execution which should demonstrate all the functionalities and/or errors if any. You must show the CRUD operations reflected in **database** in your video.

Task:

Using React Native, you will develop an app that retrieves trending videos from Wired Magazine’s channel on DailyMotion.com. The user must be able to:

- View a list of videos published on Wired Magazine’s channel
- Click on a video to see more details about the video
- Add the video their favorites list

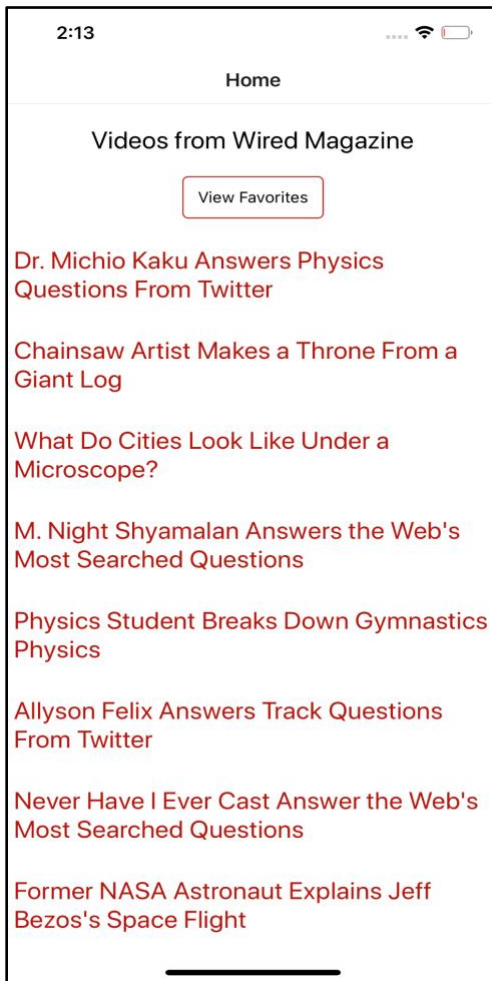
Creating Your Project

- Projects must be created using the Expo CLI.
- When creating your project, name the project: **YourFirstName-Videos**

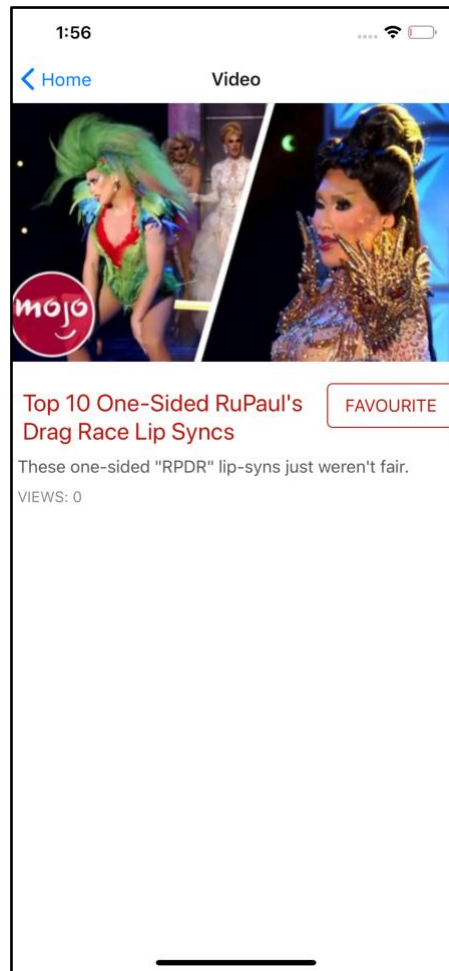
Sample UI:

- The user interface of the application should be similar to (not exactly same) as the following screenshots.
- However, the app must be reasonably pretty and must be obvious to the user where everything is. Styling must be performed using React Native’s CSS styling.

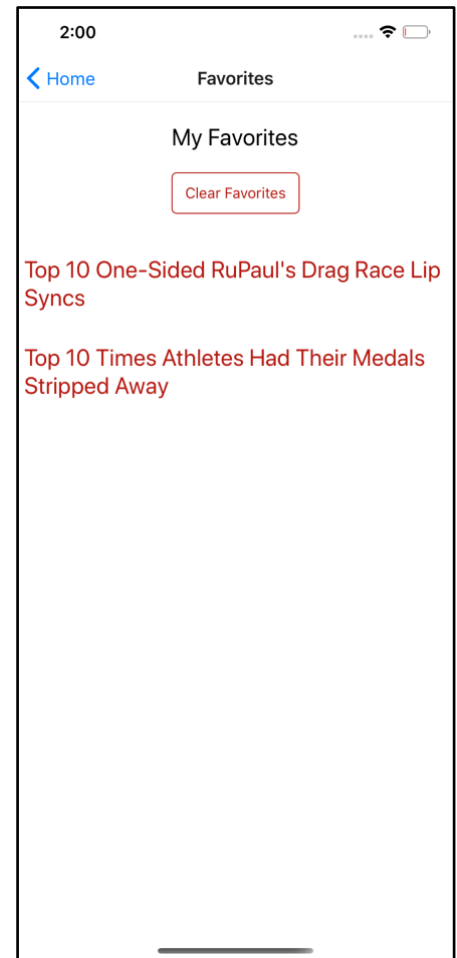
- **Note:** UI *doesn't* have to be exact same for your work.



Screen 1 - Home Screen



Screen 2 - Video Details Screen



Screen 3 – Favourite List Screen

Screen 1 - Home Screen

- When user loads the application, the initial screen displays a list of videos from the DailyMotion.com API. The list of video must be retrieved when the screen loads.
- Retrieving a list of videos from Wired Magazine's channel is: <https://api.dailymotion.com/user/x1audmk/videos?limit=20>
- The API response contains information about the video's DailyMotion id and title

Screen 2 - Video Details Screen

- Tapping on a video's name navigates the user to the Video Details screen.
- The screen displays the video thumbnail, title, description, and total number of views.
- The screen contains a FAVOURITE button.
- Tapping on this button adds the video to the user's favorites list in the database.

Screen 3 – Favourite List Screen

- The Home Page contains a button called VIEW FAVORITES
- Tapping on this button navigates the user to the Favorites Screen
- The screen shows the title of favorited videos fetched from database.
- Pressing the CLEAR FAVOURITES button will remove all videos from the favorites list in database.
- Tapping on the title of the video will navigate the user to the corresponding Video Details screen.
- If there are no favorites in the list, then:
 - display a message that says: “No favorites found”.
 - disable the CLEAR FAVORITES button

Detailed Technical Requirements

1. The app MUST be built with React Native. Your app must make appropriate use of props, state, and component communication (parent to child or child to parent).
2. The app must be architected using component-based design. Where appropriate, components should be designed to be reusable throughout the application.
3. Lists of data must be displayed using React’s <FlatList> component.
4. Within your components, you must use the elements provided by React Native. Specifically, you should use <View>,<Button>, <Text>, etc. You are not permitted to use HTML or other 3rd party elements.
5. The look and the feel of the app does not have to be exactly the same as the screenshots shown Figures 1 to 3. However, the app must be reasonably pretty and must be obvious to the user where everything is. **Styling must be performed using React Native’s CSS styling.**
6. Navigation must be implemented using Stack Navigators from the **react-navigation** library:
<https://reactnavigation.org/>
7. Your application must make appropriate use of the following API endpoints:
 - Retrieving a list of videos from Wired Magazine’s channel is: <https://api.dailymotion.com/user/x1audmk/videos?limit=20>
 - The API response contains information about the video’s DailyMotion id and title.
 - To retrieve details about a single video, use the following link:
https://api.dailymotion.com/video/#####?fields=thumbnail_240_url,description,views_total,title,created_time

You must replace ##### with the daily motion ID of the video you want details about. An example of a DailyMotion id is: **x8jzezv**

The API response contains information about the video’s name, description, number of views, and link to the video’s thumbnail (height = 320px)

8. Data persistence must be implemented using Firebase Firestore. Your application is only permitted to add, remove, or retrieve **favorited videos** from Firestore. You are **not permitted** to use Firestore to pass data between components.
9. You are **NOT** allowed to AsyncStorage or any other storage provider.
10. When favoriting a video, you may assume the user will only ever add a video to their favorites list once. You are **not required** to implement logic to check if a video has been previously favorited, duplicate entries in the favorites list, or disabling the Favorite button.
11. When performing equality operations, you must use `===`. Marks will be deducted for usage with `==`.
12. When declaring variables, you must use `let/const` syntax. Marks will be deducted for inappropriate (and general) usage of `var`.
13. You may use either traditional `function()` syntax or arrow `()=>{}` syntax. However, you must select a single style and use it consistently throughout your application. For example, if you choose to declare your functions using traditional `function()` syntax, then all your functions must be declared in this style. Marks deducted for inconsistent syntax.
14. All components must be defined using function-based components. In other words, your component definitions must look like:

```
const MyComponent = (props) => { return(); }
```

or

```
function NumberPicker() { return(); }
```

Rubrics:

Each section will be graded based on coding style, architecture, design, technical correctness per requirement.

Criteria	Points
Home Screen	20
Video Details Screen	40
Favourite List Screen	40
Total	100