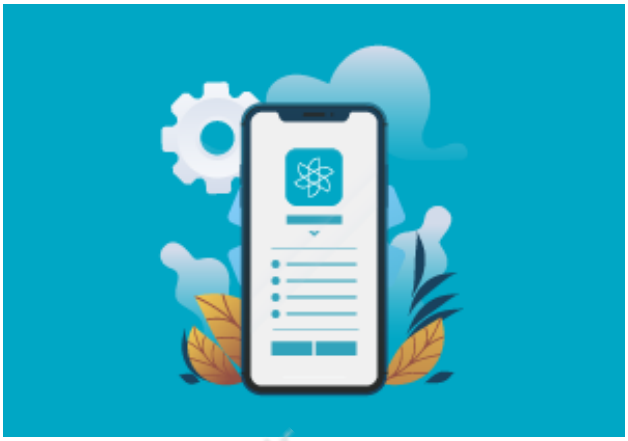
Capstone class : App Publishing and Local Environment Setup



What is our GOAL for this MODULE?

We learned to set up expo on our local environment. We also learned to generate aab or ipa files which can be published on playstore or appstore.

What did we ACHIEVE in the class TODAY?

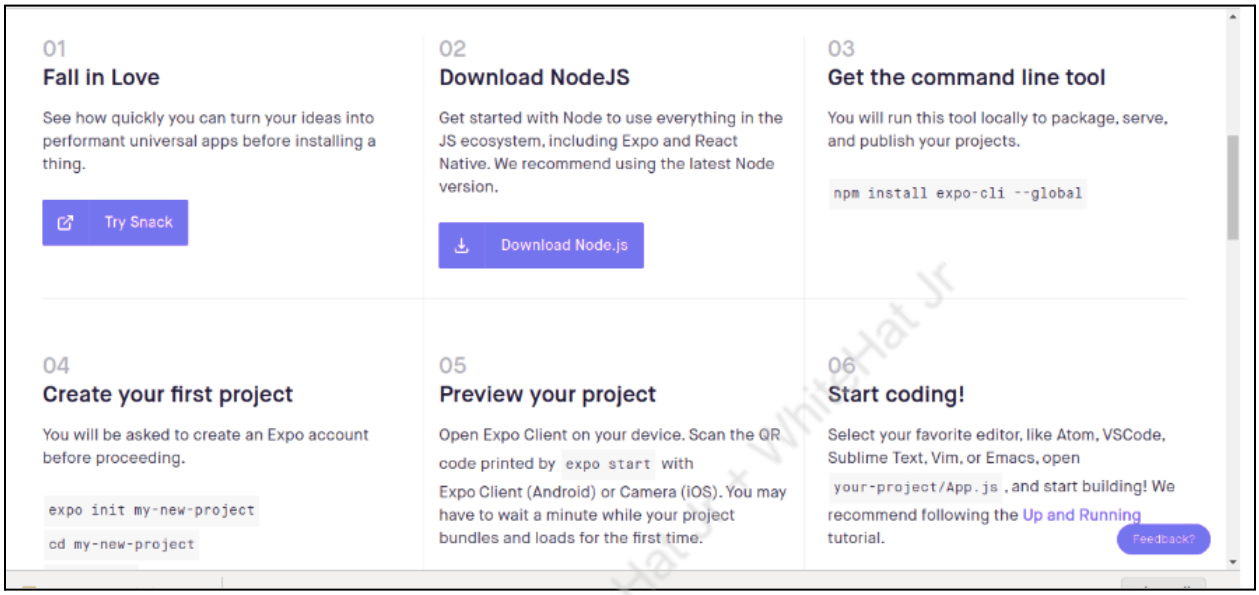
- Set up expo on the local machine.
- Generated aab or ipa files for apps to be published on playstore.
- Built a weather app to capture the weather of the current location

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Installing expo tools on the local machine
- Build aab or ipa files.

How did we DO the activities?

We will be following instructions given in Expo documentation on its website to first install expo on our local machine.



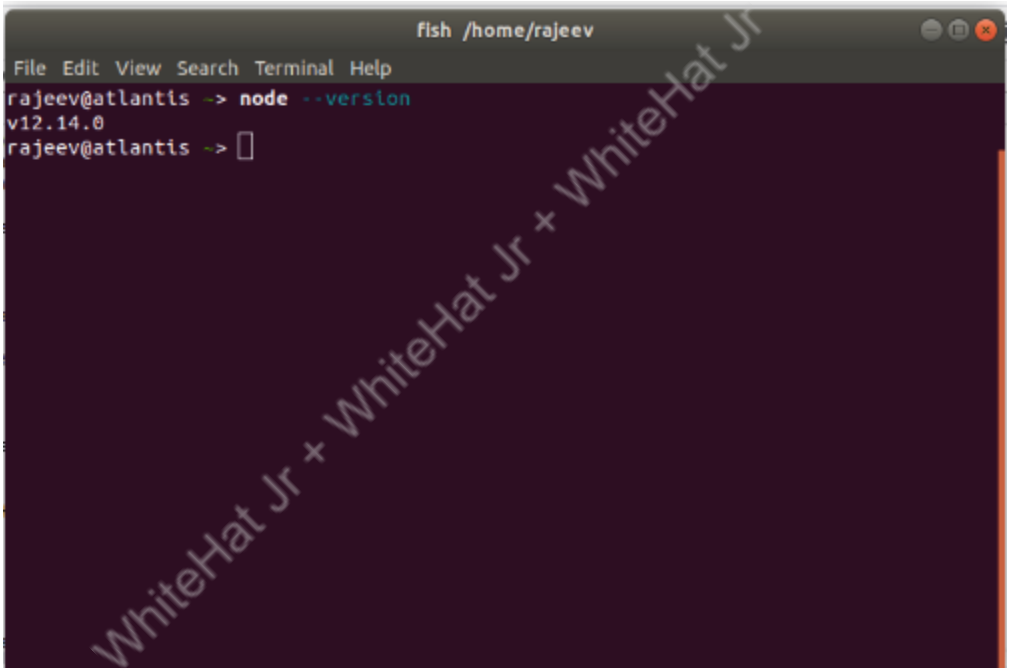
1. First, we will install node on our system. So far, we have only run javascript inside a browser. Node allows us to run javascript outside our browser as well.

    - For Windows users:
        - Download node directly from the given link.
        - Unzip the file.
        - Run the executable inside it (exe) file to install node.
        - To check if the node was installed properly, open cmd and type node --version: It should show the node version which was installed.

    - For Mac users:
        - Install homebrew first. Homebrew is a package manager for your operating system. It helps you in easily installing programs from the terminal.
        - To install homebrew, open your terminal and type:
          ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
        - *Note: You might have to add "sudo" before the command if you do not have permission to install packages on your OS.
          "sudo" stands for "do as a super user". You might have to run: sudo ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

- Now install node.
- On your terminal type: sudo brew install node.
- Check if node is installed on your system by typing in the terminal: node --version.

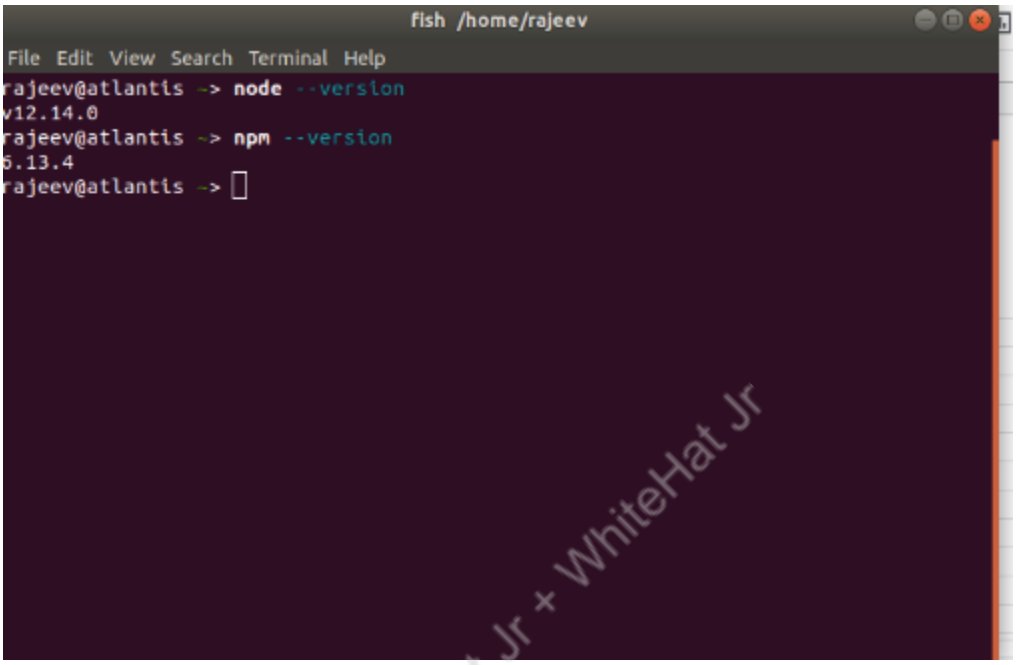
- For Ubuntu users:
  - Open your terminal and type: sudo apt install node.
  - This will install node on your system.
  - Check the node installation by typing: node --version



- When you install node, npm also gets installed. 'npm' stands for 'node package manager'.
- All the libraries that we used in snack including react, react-native, firebase, react-navigation, they all come as node packages. 'npm' helps us in installing and maintaining these packages.

*Note: You will learn more about it when we actually use 'npm'.

- You can quickly check for ‘npm’ installation using: npm --version.



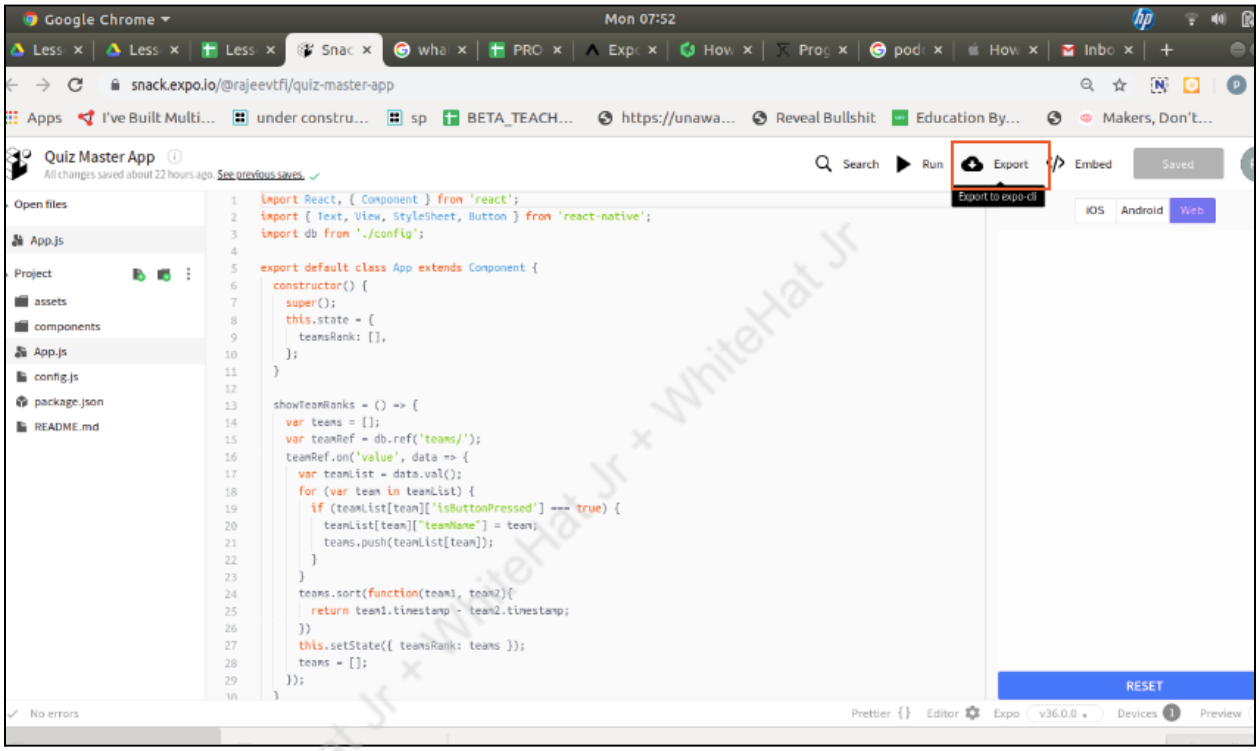
2. Now we will be using npm to install the expo command-line tool.
    - Expo command-line tool or ‘expo-cli’ comes with many libraries and tools already installed which help us in quickly getting started with building react native apps.

    - To install ‘expo-cli’, on your terminal type: npm install expo-cli --global
        - The "global" tag installs expo with a global scope. This means you can use expo anywhere on your system. Without a global tag, the expo will be installed only in the folder in which you are running the command.

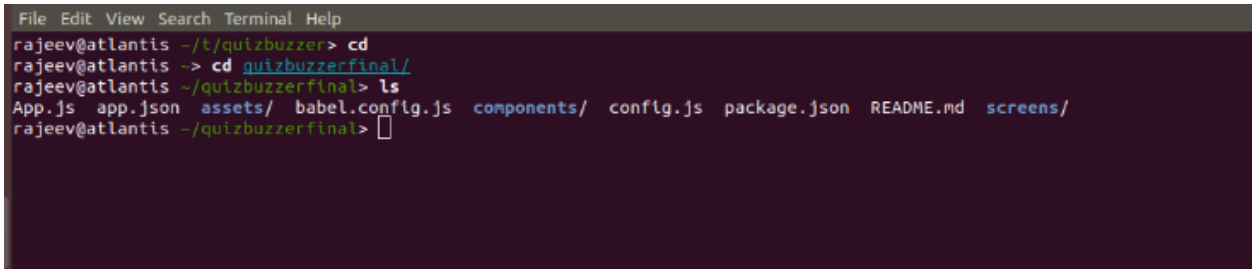
We have ‘expo’ installed on our system now.

Let's first convert our quiz buzzer app into aab or ipa so that it can be installed on the phones and tested or published on playstore or app store.

1. Navigate to the link for the Quiz Buzzer App, you created (Teacher Activity 3) and export the files.



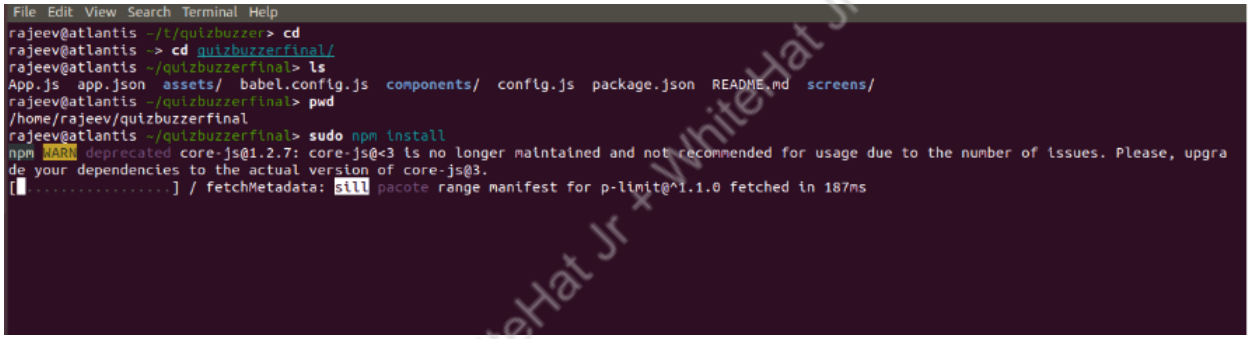
2. Extract the downloaded files into a folder of your choice. Now, open your terminal again and navigate to the folder where you extracted the files.



3. To navigate to different folders on your computer using the terminal use cd command.
    - 'cd' stands for 'change directory'.

4. Now type ‘ls’ to list all the files inside the folder. You should be able to see all the files and folders you had created on expo snack.

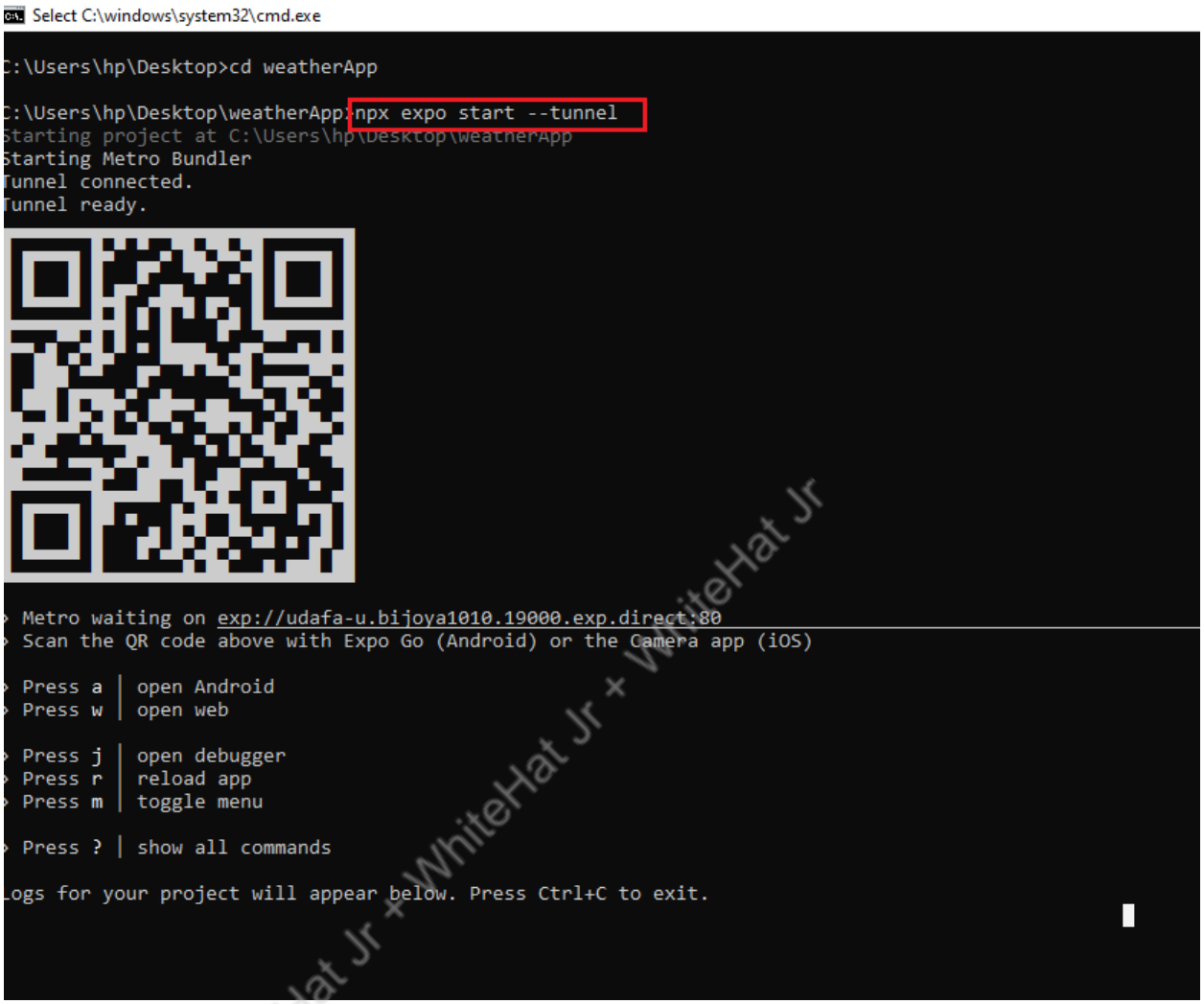
5. Now we need to install all the packages we had used in our app on our local machine.
    - React, react-native, react-navigation, firebase, etc. packages are listed inside ‘package.json’.
    - To install all the packages in ‘package.json’, we just need to type: npx expo install.
    - ‘npm’ will look into the ‘package.json’ file and install all the packages needed for our project.



6. Now that all our packages are installed, quickly test if our application is installed correctly.

    - You can type: npx expo start --tunnel
    - This will start your project. It will generate a QR code. You can scan the QR code on an expo client installed on your phone to open the app.

*Note: Your computer and your phone must be connected to the same network for this to work.



- We have the app installed on our local machine now. We can now open any of our files - App.js, Homescreen.js, Buzzerscreen.js etc. - on any of our favourite editors.
- We can change the code inside and the app will change live on our phone!
- This means that you can code and test your app on the phone at the same time. It will also show warnings and errors.
- Now, try opening the AppHeader.js file in an editor and change the name of your app to see it updated on your phone.

How to build aab / ipa files from this project:

1. Before building the aab or ipa, we need to add a unique identifier for playstore and

appstore to remember our app with.

- This is done using reverse web domain name inside app.json file- since each user's web domain of each user will be different and unique. You can use any dummy domain name for now.

2. Now press Ctrl + C (Command + C for Mac) to stop the metro bundler you ran using npx expo start --tunnel.

3. In the same folder, run the following commands:

    1. Install the latest EAS CLI. EAS CLI is the command-line app that you will use to interact with EAS services from your computer.

        npm install -g eas-cli

    2. Login to your expo account using the following command-

        eas login

    3. Configure your project for android or iOS-

        eas build:configure

    4. Build your project -

        eas build --platform android

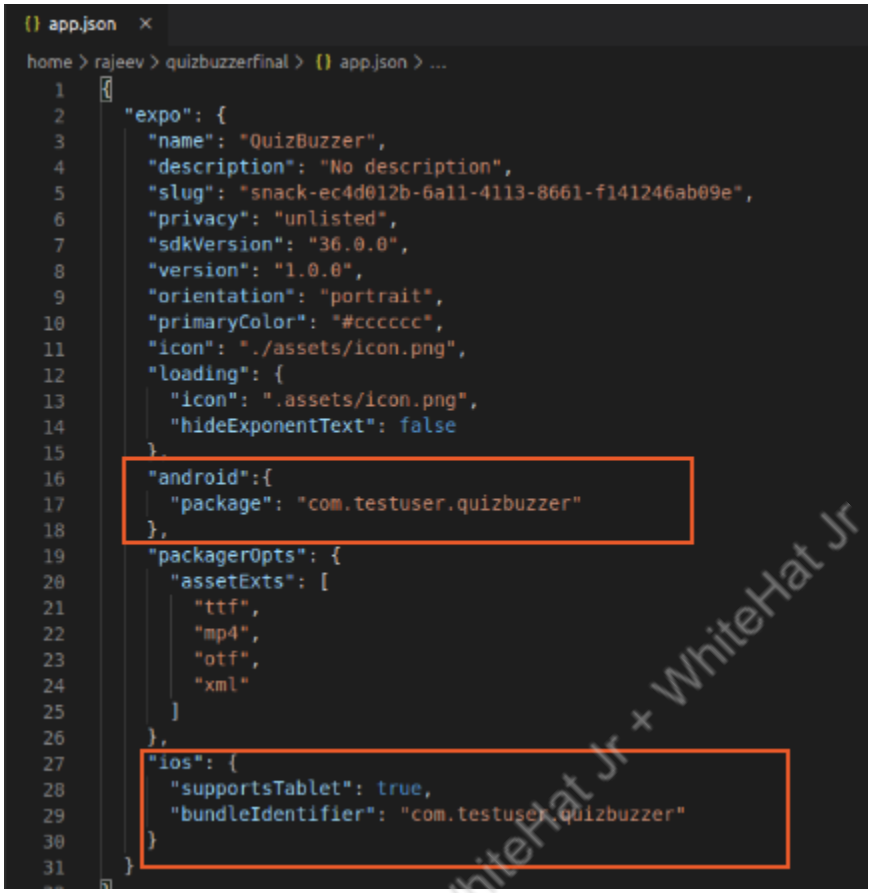
        or,

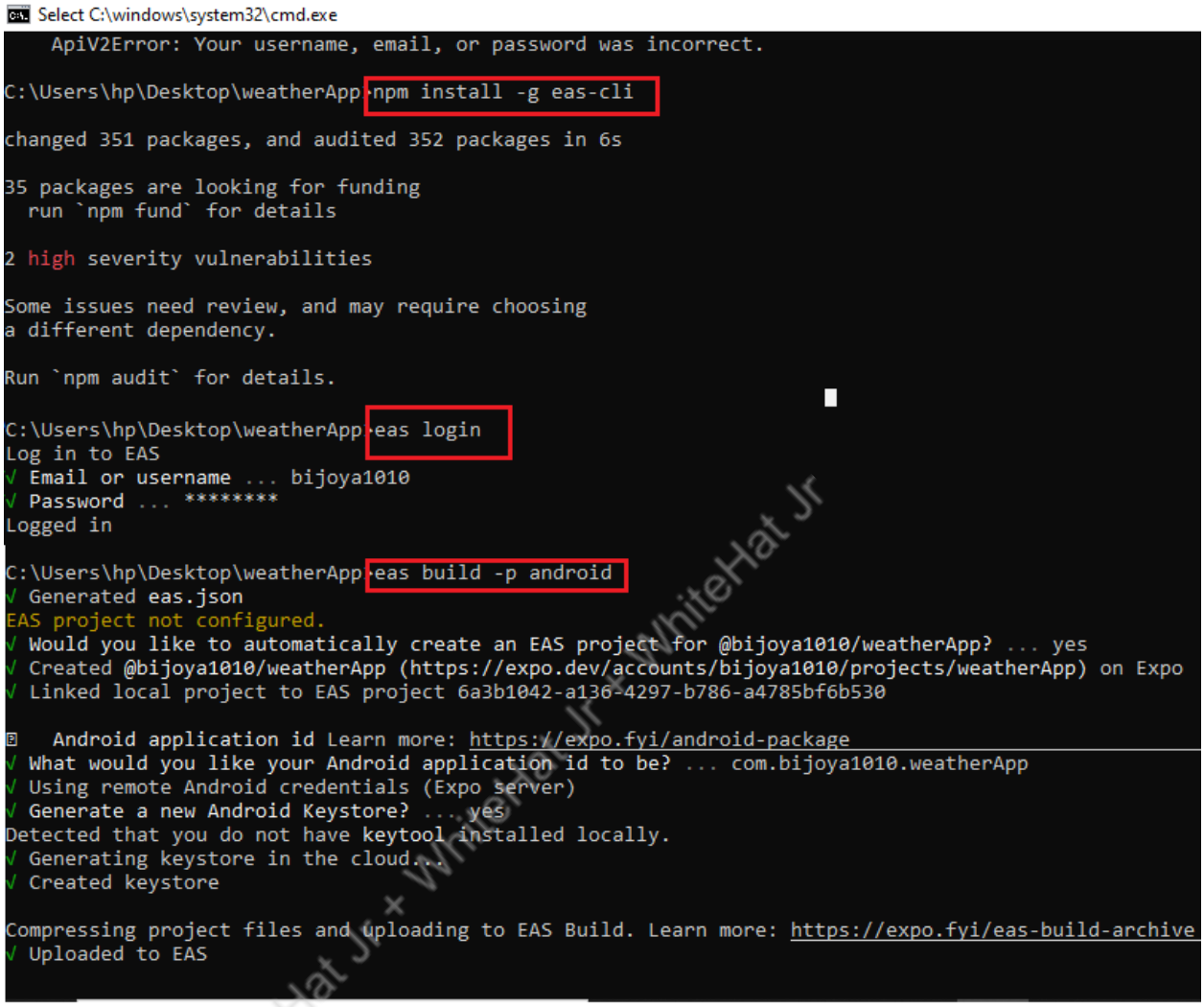
        eas build --platform ios

*Note 1: There might be an error like "unable to resolve react-native-gesture handler."
This means that the above library did not get correctly installed.
Run - npx expo install react-native-gesture-handler
This will install the above package. And then you can run build commands again.

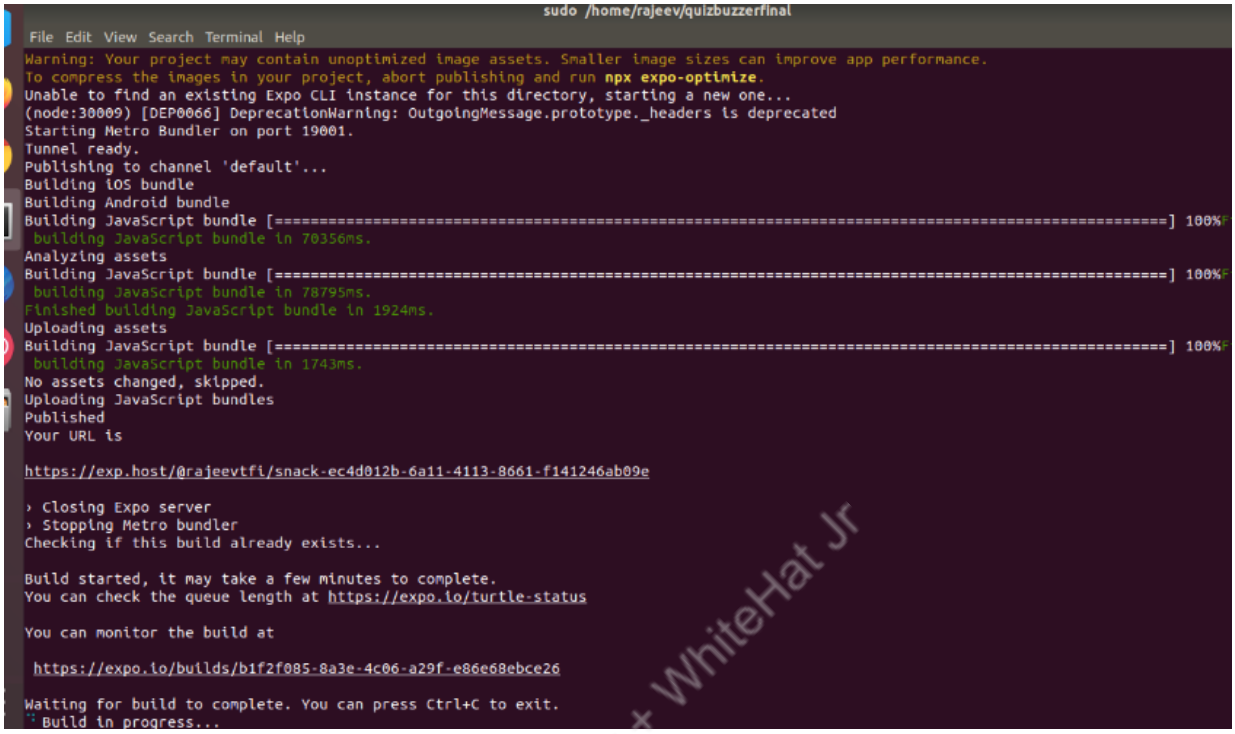
*Note 2: For ios build, you will need an apple id and password for your paid developer account. It will authenticate the developer account. The student will have to create a paid developer account for this purpose.

*Note 3: Expo builds aab on a shared server machine. The build will fail if one of the expo's server machines is not available for building.





4. The build command takes a while. You can visit the build link given in the terminal to see the progress. Once the build is finished, you can download the aab file directly from there.



5. After some time, you can see the link of the aab file. You can click on it to download it and publish it on the playstore.

   aab stands for **Android App bundle.**
   aab is a publishing format that includes all the resources and compiled code for an application. (aab file can published on playstore, but cannot be directly installed on your phone)

Now that you have expo installed, you can also write react native code on your local machine and test them on your phone.

Let's try to build a simple project.

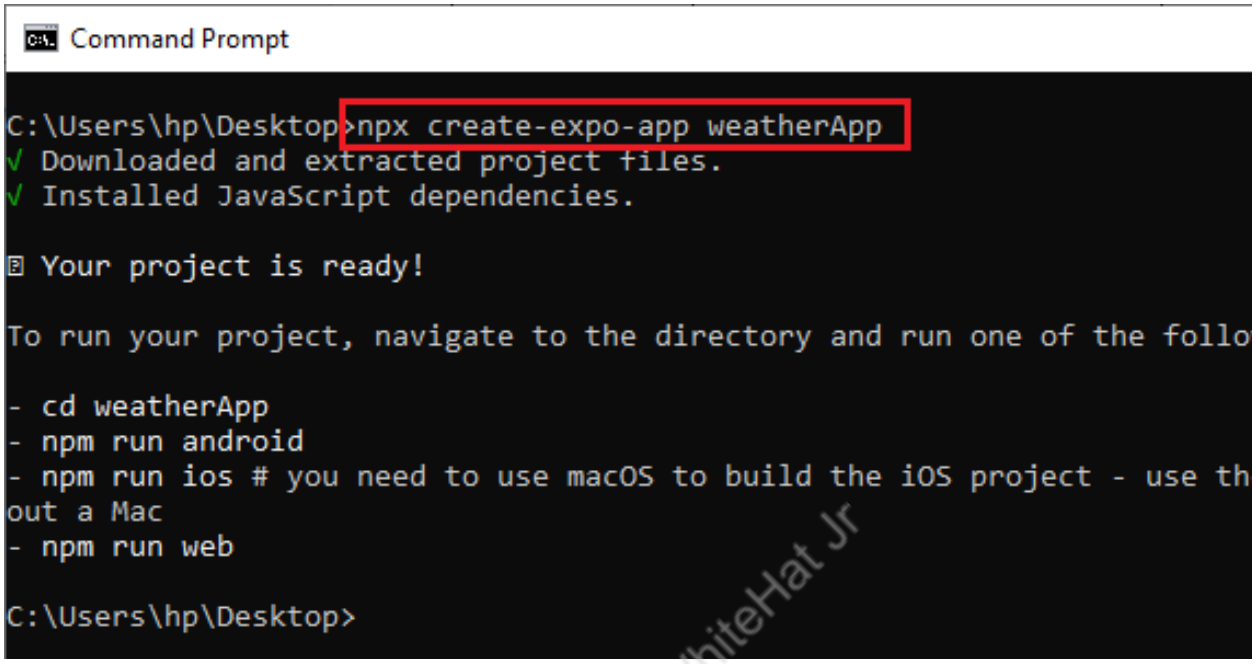
1. Navigate to your home directory from your terminal and type
   npx create-expo-app <yournewproject name>

2. You will be asked to choose a template. Choose a blank template.

3. 'npm' will seek permission to install all the expo libraries that come packaged with expo. Give permission.

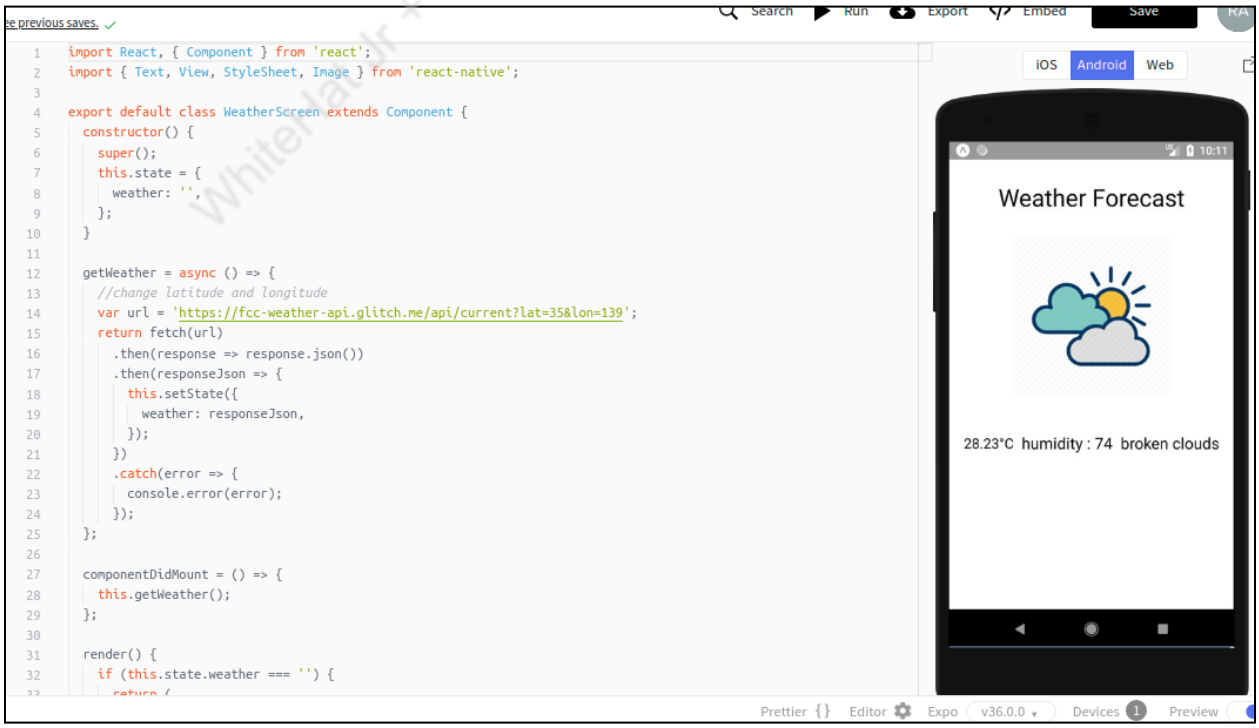
4. It will take some time for the project to build.

*Note: you can safely ignore the warnings for now.



Let's now build a simple weather forecasting app which gets weather and temperature data from a weather API and displays it on a home screen.

1. You can open your App.js file using any editor and start writing your code.



2. Get Json data from the API.

3. Change the state of the weather using the data.

4. Use the weather state and display it on the App User Interface

5. Now navigate to the folder in your terminal and run -
   cd <yourproject folder name>
   npx expo start --tunnel

6. You can scan your code and run the app on the Expo client app on your phone to test it.

7. You can also build aab or ipa files using the expo tools you just learned.

What's NEXT?

In the next class, we will work on another case study to create an app which solves a practical problem.

EXTEND YOUR KNOWLEDGE

1. Expo documentation for EAS build: https://docs.expo.dev/build/setup/