

LevelUP

React Native

Timeline

Day	Goal
Day 1, 2, 3, 4	Learn React Native + GitHub + Writing Test Cases
Day 5	Interview + Demo
Day 6, 7, 8	Assignment
Day 9	Technical Interview + Demo + Code Review

Learning Resources

Getting started and Set up

- React Native Website : <https://reactjs.org/>
- Website Docs : <https://reactjs.org/docs/getting-started.html>
- Set Up: <https://reactnative.dev/docs/environment-setup>

Components

State, Props, Views, Styling, FlexBox, TextInput, images, buttons, Routers, Modal, ActivityIndicator, Picker, StatusBar, Switch, text, alert, AsyncStorage, axios, HTTP, Events, Animations, TouchableHighlight, TouchableOpacity, refreshControl, Hooks.

Components and APIs

- **Mandatory** - React Native Resources(Tutorials point):
https://www.tutorialspoint.com/react_native/react_native_overview.htm

- Core-Components :
<https://reactnative.dev/docs/components-and-apis>
- APIs : <https://reactnative.dev/docs/alert>

Video Tutorials

- **Mandatory** - Programming with mosh(Youtube) :
<https://www.youtube.com/watch?v=0-S5a0eXPoc>
- **Mandatory** - Net ninja Tutorials (Youtube): (Project Mandatory -
Todo App)
<https://www.youtube.com/watch?v=ur6l5m2nTvk&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ>
- Optional - Academind (Youtube Crash-Course):
<https://www.youtube.com/watch?v=VozPNrt-LfE>

Folder Structure

<https://github.com/anhquan291/e-commerce-app-react-native> - Github link

▼ REACT-NATIVE-CLI-APP

▼ src

▼ api

JS APIServices.js

JS APIStrings.js

▼ assets

> fonts

> images

{...} cityList.json

{...} countries.json

{...} currencyList.json

▼ components

> GridView

▼ Header

JS HeaderComponent.js

JS HeaderStyle.js

> ListView

> Title

JS Block.js

JS CustomLoader.js

JS Footer.js

JS Loader.js

JS NotificationCard.js

JS Question.js

▼ Redux

> Reducer

> Store

▼ routes

JS auth.js

Routes.js

▼ screens

> AboutUs

> Call

> Chat

Assignment

Overview

You will be building a part of the Palette Mobile application. The way the track is structured, you are required to work with your teammate and deliver a complete end to end application.


About the app - Palette is an built for the education sector. The goal is to bring students closer to the adults in their life so that there is a holistic growth in the student. Part of the application involves a To-Do module , which is what you will be building.

You will be assigned a teammate who will be working along with you to build the APIs on the backend. You will be evaluated based on how you collaborate with your teammate and not how you work alone.

UI Design

Prototype -

<https://www.figma.com/proto/foKBPtSPIHxGTNgMIHzzMQ/LevelUp---Palette?node-id=37%3A3340&scaling=scale-down-width&page-id=5%3A2&startin-g-point-node-id=37%3A3340>

If you are unable to properly view the design prototype on figma download this pdf -  LevelUp July - Mobile Designs.pdf

Design File (Use this for inspecting the components) -

<https://www.figma.com/file/foKBPtSPIHxGTNgMIHzzMQ/LevelUp---Palette?node-id=5%3A2>

Milestones

Milestone 1

- Complete course on NodeJS with at least 2 mini projects (you can build whatever is in the course)
- Theoretical Interview on Day 4 - your understanding of the concepts will be assessed. This will contribute to your final grade.

Milestone 2

- Complete the assignment attached to this LevelUP track
- Demo
- Code Review
- Technical Interview
 - Technical understanding
 - Soft Skills

Evaluation Criteria

- Assignment's shall be graded on a scale of 1 to 100

Merely covering the requirements would not be enough. There would be a heavy focus on code quality and robustness of code in general. The evaluation criteria as mentioned below would be adhered to at completion of given time.

Criteria	Percentage
Output matching the requirement (Completion, Desired output)	40%
Creativity (A direction in taking a step more than what is asked for) and SE Process(Git, Raising the red flag for requirement clarification or any other blocker etc.)	10%
Pixel Perfect Implementation and effective layouts	15%
Code Quality + Git 1. Code structure (Quality, Best practices) 2. Git workflow usage	25%
Collaboration	10%

NOTE:

- The following shall strictly not be entertained:
 - Deviations in the timeline
 - Taking others help (Other than requirement clarification)

- Plagiarism of any kind
- Make regular git commits. Each individual commit will be verified.

Folder Structure - Best Practices