How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: MahaHajaj

News App

Description

This App keep you updated with trending and last news and articles from many sources , and allows you to bookmark articles to save them in your bookmark list .

Java language will be used for development , and this app will be using stable versions of all libraries, Gradle and Android Studio .
App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.

Intended User

People interested in tracking the latest news from many sources .

Features

List the main features of your app. For example:
- Display last news .
- Bookmark news .
- Order the news .
- Choose the news sources .

User Interface Mocks

The user interface Mocks in this link :
https://ninjastorage.blob.core.windows.net/html/Z7K6CDx/fc8fa194-24ac-48e7-ad59-32f674ce11ec.html

widget :

widget contain list of bookmark news .

Key Considerations

How will your app handle data persistence?

This App will use google-news API to get the news , And save the bookmark by using Firebase Realtime Database .

Describe any edge or corner cases in the UX.

If the user sign up the app will navigate to main activity .

App includes support for accessibility. That includes content descriptions, and navigation using a D-pad .

Describe any libraries you'll be using and share your reasoning for including them.

Picasso to handle the loading and caching of images.
ButterKnife to bind the view .
Retrofit for network request .
Firebase for store bookmark in database , And for authentication

Describe how you will implement Google Play Services or other external services.

Firebase authentication to login and sign up , Firebase realtime database to store bookmark , and google news API to get news from the web .

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Create an empty new project in android studio , and add the libraries to build.gradle file and sync .

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:
- Build UI for MainActivity .
- Build UI for SignUp Activity .
- Build UI for bookmark Activity .
- Build UI for LogIn Activity .
- Build UI for Setting Activity .

Task 3: Main Activity

Describe the next task. List the subtasks :
- Create layout to view the news .
- Use AsyncTask to check for active internet connection .
- Get the news from API to display in main activity by retrofit , Retrofit will parse the API data on the **background thread**. Then return the result back to UI thread via *onResponse or onFailure* methods .
- Create menu to move to logIn/ signup activity , bookmark activity , and setting activity

Task 4: Implement Google Play Services

Describe the next task. List the subtasks :
- implement firebase authentication to sign up and login .
- implement firebase realtime database to store the bookmark news to view in the bookmark activity .
- implement google news API to get to data from the web and connection with internet by retrofit .

Task 5: Setting Activity

Describe the next task. List the subtasks :
- Create layout for setting activity .
- Create orderby preferance .
- Create select source preferance .

Task 6: Widget

Describe the next task. List the subtasks :
- Create layout for widget .
- Get the bookmark news from firebase realtime database to display in widget list .

Add as many tasks as you need to complete your app.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File →
Download as PDF]
  - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:
- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"