

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1:Splash Screen](#)

[Screen 2:Home screen](#)

[Screen 3:Events list screen](#)

[Screen 4:peek a contest](#)

[Screen 5:contest details screen](#)

[Screen 6:Settings screen](#)

[Screen 7:Credits screen](#)

[App Widget](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Project requirements](#)

[Next Steps: Required Tasks](#)

[Task 1: Implement splash screen](#)

[Task 2: Implement Home screen](#)

[Task 3: Implement contest list screen](#)

[Task 4: Implement contest detail screen](#)

[Task 5: Handle error cases](#)

[Task 6: Implement database](#)

[Task 7: Implement settings screen](#)

[Task 8: Implement app widget](#)

**GitHub Username:** Ni3Verma

# Code Time

## Description

Best way to learn coding is to do it regularly and think about new and best ways to solve a problem. There are many websites which provide coding questions for practice free of cost and they host a number of competitions. This app will show its user a list of contests being held worldwide on various websites (that user will select) and then user can see details of contest, visit the website, add event in calendar.

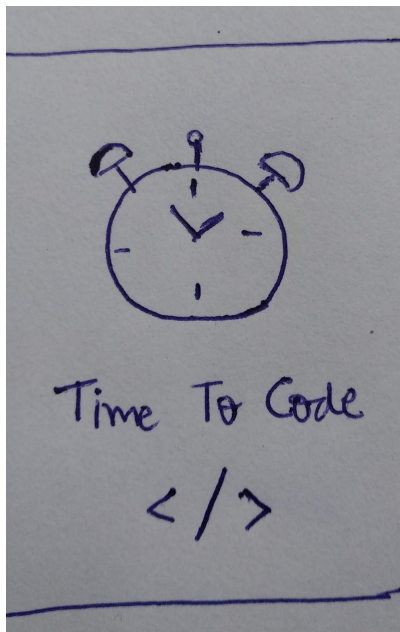
## Intended User

This app is not intended for any particular age group or gender. It is for anyone who wants to learn coding or wants to get more strong at coding or wants a strong grip on logic building and data structure.

## Features

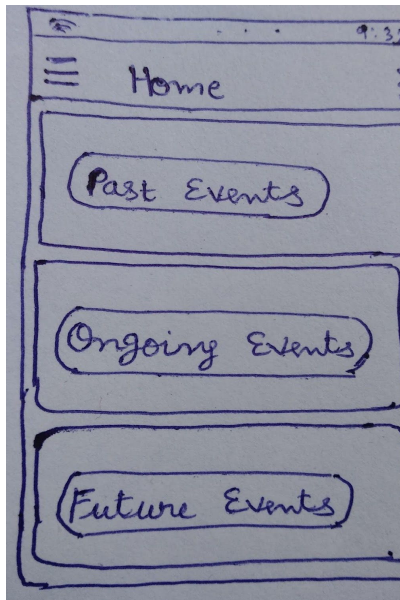
- Shows programming contests from various websites
- User can select his favourite websites
- Local room database will be used to save data so that network request is done only on demand
- User can add any event directly to calendar
- User can also share contest details with friends

## User Interface Mocks



### Screen 1: Splash Screen

Splash screen is shown to the user in the beginning. This screen will be simple and it will only show app logo and its name. Some animation will be added to make it interactive. Main task of this screen is to load data from local database or make a network request if there is no data.



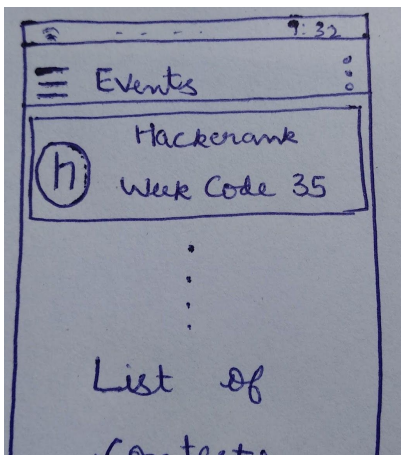
### Screen 2:Home screen

Here user will see three options:

Past events: to see the past events that were help and practice these questions.

Ongoing events: here user will see events that are live.

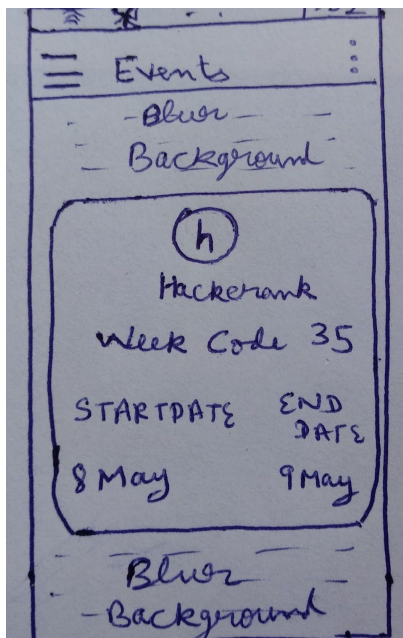
Future events: here user will be able to see all the events that will be held in the future.



### Screen 3:Events list screen

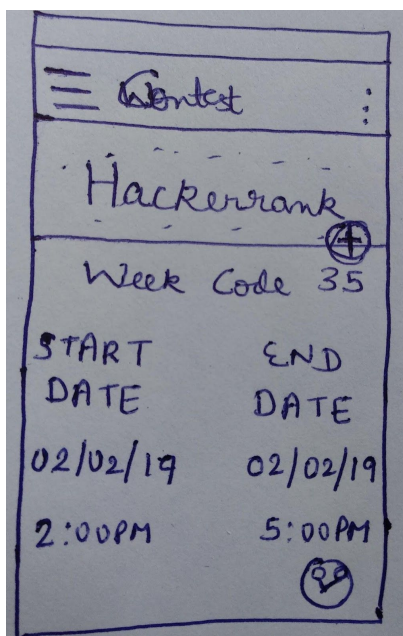
On this screen user will minimum information about the event.website logo,website name,contest name.

List of events in recyclerview will be shown. When user clicks on any list item then proper details will be shown to the user and events detail activity will be shown. However if user long clicks on the list item then he will be shown only details of the event like name, start date, end date, time with iPhone 3D touch effect or like Instagram peek and pop feature.



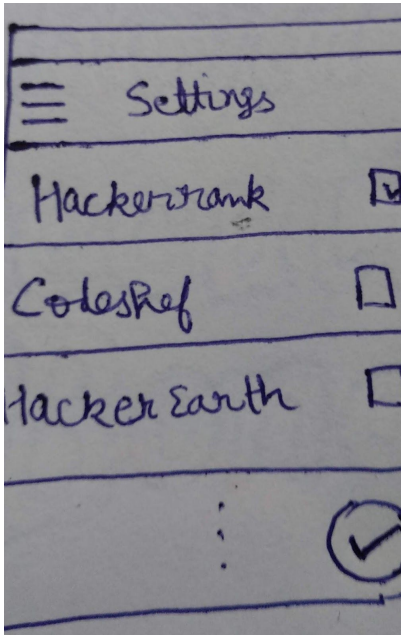
#### Screen 4: peek a contest

Here user will see minimum but enough information about a contest. This feature is added because it looks beautiful and it's easy to see quick details of many contests easily and moreover it will be challenging to implement it.



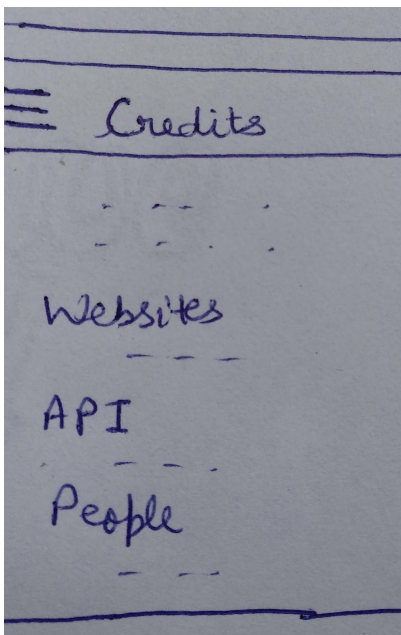
#### Screen 5: contest details screen

When user clicks on a contest then he will be taken to the full details of the contest. Here user will see all of the details of the contest and from this screen user can add it to the calendar or he can share the details with friends or visit the website and see additional summary and register for the contest.



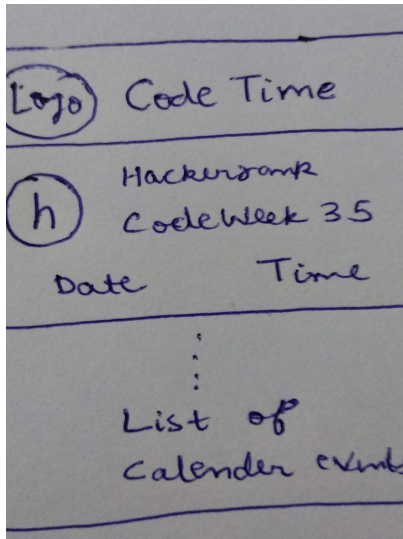
### Screen 6:Settings screen

From here user can select the websites from which they want to see the contests. This will be a preference activity. User can select/deselect any website and data will be updated accordingly.



### Screen 7:Credits screen

This is a hardcoded screen with no interaction with user.here i will credit all of the people/websites/libraries/REST API.



## App Widget

App widget will show all of the events that user have added to the calendar in a list.

Each list item will consist of event name, date and time of contest.

## Key Considerations

How will your app handle data persistence?

Room database will be implemented to persist data. every database query will be run in background thread. database will be synced everyday so that user can see latest data. A button will also be provided from where user can force refresh the database.

Describe any edge or corner cases in the UX.

Implementing peek and pop feature is gonna be a challenging task. A snapshot of current screen should be taken then it will be blurred and then applied as background to get a blur effect.

Navigation drawer will also be implemented so fragments should be dealt properly. Back button and fragment history need to be tweaked in a careful manner.

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

**Picasso** will be used to load any image resource as it is fast, use caching techniques, and is most popular. (version **2.71828**)

**Volley** will be used to make network request. (version **1.1.1**)

**Room** persistent library will be used to persist data. (version **1.1.1**)

### Project requirements

- 1) App will be written solely in java language and it will use stable version all libraries, gradle and android studio.
- 2) All string resources will be saved in the strings.xml file
- 3) App enables RTL layout switching
- 4) App provides support for accessibility such as content description for image
- 5) Live data and view model will be used while implementing room db.
- 6) App uses intent service to pull data from API

## Next Steps: Required Tasks

### Task 1: Implement splash screen

This screen is easy to make. It will be a simple screen with little animation. In the background network and disk requests will be made to get and store data.

### Task 2: Implement Home screen

- Build UI for home screen
- Add animation to each section
- Construct home screen adapter
- Refresh fab to get latest data from REST API

### Task 3: Implement contest list screen

- Add recyclerview

- Construct adapter
- Add onclick and on long click for different actions

#### **Task 4: Implement contest detail screen**

- Build attractive UI
- Show all details
- Remind me button
- Visit website button
- Share with friends button

#### **Task 5: Handle error cases**

- Check internet connectivity
- Rotation of screen during sync
- Handle Invalid network requests and responses
- Memory leaks
- Handle database requests on background thread
- Make proper intents to share contest details or add to calendar

#### **Task 6: Implement database**

Room database is easy to implement. Entities and DAO will be made to store and query data. database will be refreshed everyday at 2 am.

#### **Task 7: Implement settings screen**

Implementation of this screen will be done by using preference class. No separate layout will be made, rather checkbox preference will be used to store user preferences.

#### **Task 8: Implement app widget**

App widget will be simple and to the point. It will show user all of the events that he wants to take part in and have added to the calendar. list of these events will be shown. Whenever any new event is added to the calendar then this widget will be updated.