Notion Clone

Sai Kalyan
Shatrugna Rao

Introduction

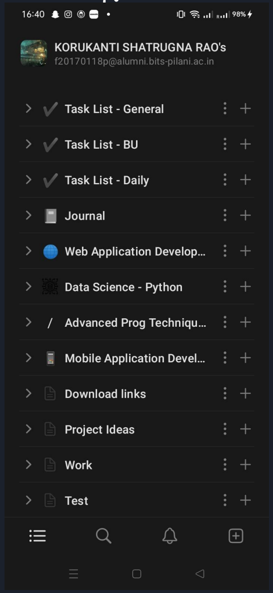
- This app is made using android Jetpack Compose framework.
- It is a note taking app, with advanced components like
  - A checkbox component
  - A table component
  - A redirect component
  - An Image component
- The motivation behind the app is to learn the latest Jetpack compose framework and utilize its feature to develop a dynamic app.

Related Work

Our app is similar to the notion app which also a note taking application but with most advanced ui features.

The diff between our app and Notion is

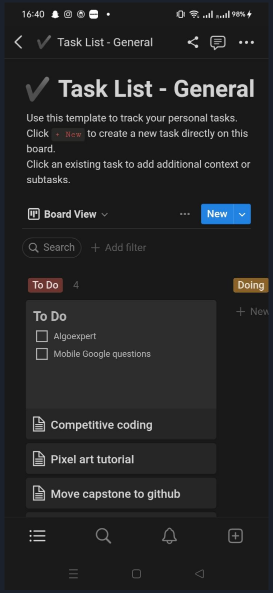
- Notion app is built on top off frontend technology like html, css, React and Capacitor.
- Our app is built using jetpack compose which lacks community unlike react community.
- Our app is lacking the ui and animations which could be done easily using a frontend tech.
- But our app is built using native android, which boost in performance when compared to



16:40 98%

KORUKANTI SHATRUGNA RAO's
f20170118p@alumni.bits-pilani.ac.in

- Task List - General
- Task List - BU
- Task List - Daily
- Journal
- Web Application Develop...
- Data Science - Python
- / Advanced Prog Techniqu...
- Mobile Application Devel...
- Download links
- Project Ideas
- Work
- Test



16:40 98%

Task List - General

Task List - General

Use this template to track your personal tasks.
Click + New to create a new task directly on this board.
Click an existing task to add additional context or subtasks.

Board View ··· New

Search + Add filter

To Do 4 Doing

+ New

To Do
□ Algoexpert
□ Mobile Google questions

Competitive coding

Pixel art tutorial

Move capstone to github

Jetpack Compose

Jetpack Compose is a modern UI toolkit recently launched by Google which is used for building native Android UI. It simplifies and accelerates the UI development with less code, Kotlin APIs, and powerful tools.

- **Declarative:** It is fully declarative so that you can describe your UI components by calling some predefined functions.
- **Increase development speed:** Previously developers have to work on the XML file and Kotlin file. But with the help of jetpack compose this becomes easy and developers only have to work on the Kotlin files that’s why it will help developers in increasing development speed.

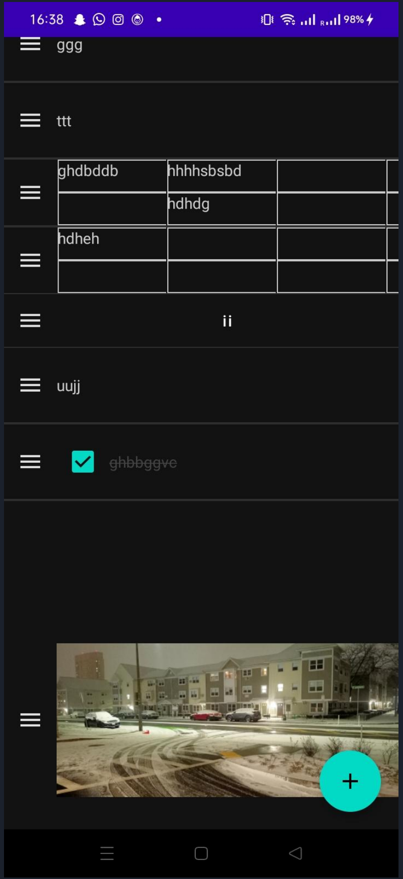
Drawbacks

- Does not have a huge community.
- Styling is lacking flexibility

Implemented Requirements

The Following features were implemented :

- CRUD operation on most of the components.
- Dynamic creation of components, enhancing reusability
- Room DB api implemented to store the app data.
- Recursive page creation.
- Serialization and deserialization of ui data.



ggg

ttt

ghdbddb	hhhhsbsbd		
	hdhdg		

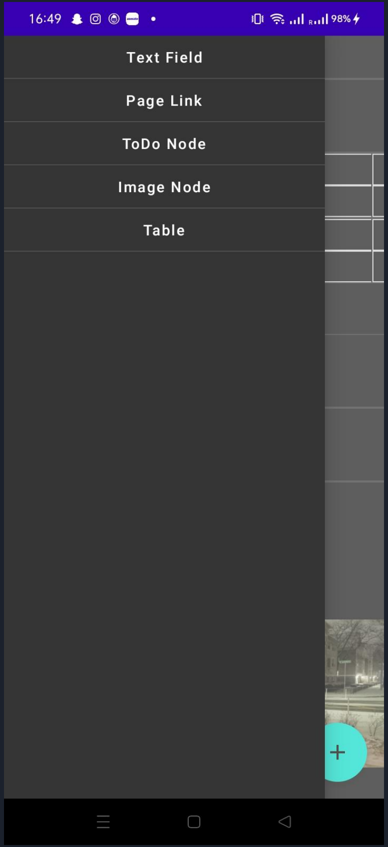
hdheh			

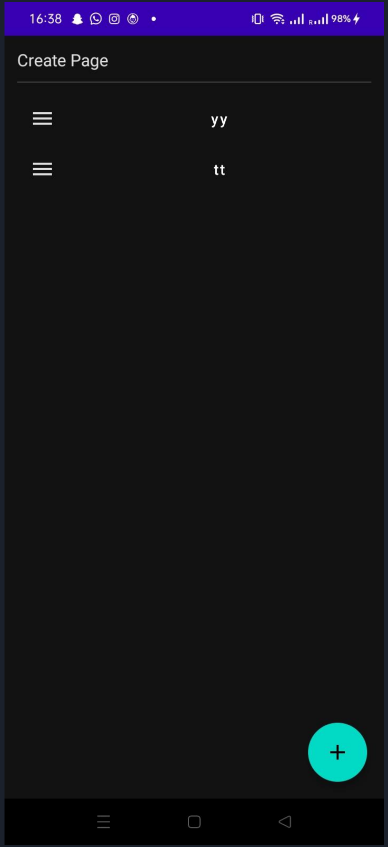
ii

uujj

☑ ~~ghbbggve~~

App Screenshot

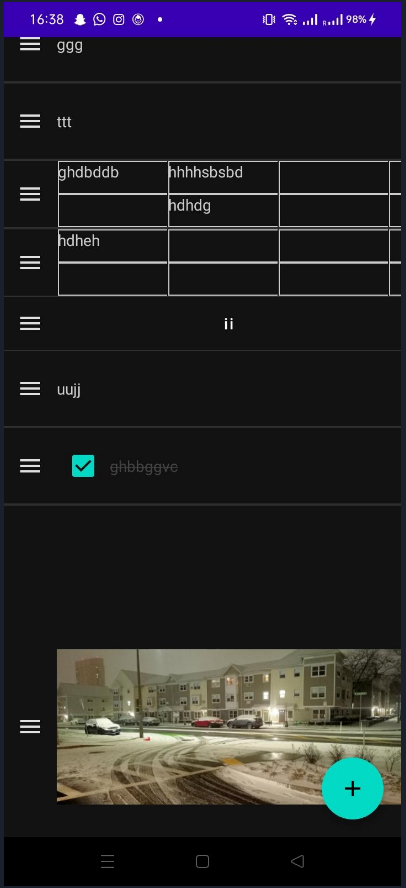




Create Page

yy

tt



ggg

ttt

ghdbddb	hhhhsbsbd		
	hdhdg		
hdheh			

ii

uujj

☑ ~~ghbbggve~~

DEMO

Project Structure

Components - contains kotlin classes with composable render function and state variables. They implement Node interface. It handles deserialization and serialization of room DB

- NodeImage - responsible for adding image
- NodePageLink - responsible for redirecting
- NodeTable - responsible for table
- NodeTextField - responsible for basictextfield
- NodeToDo - responsible for checkbox textfield

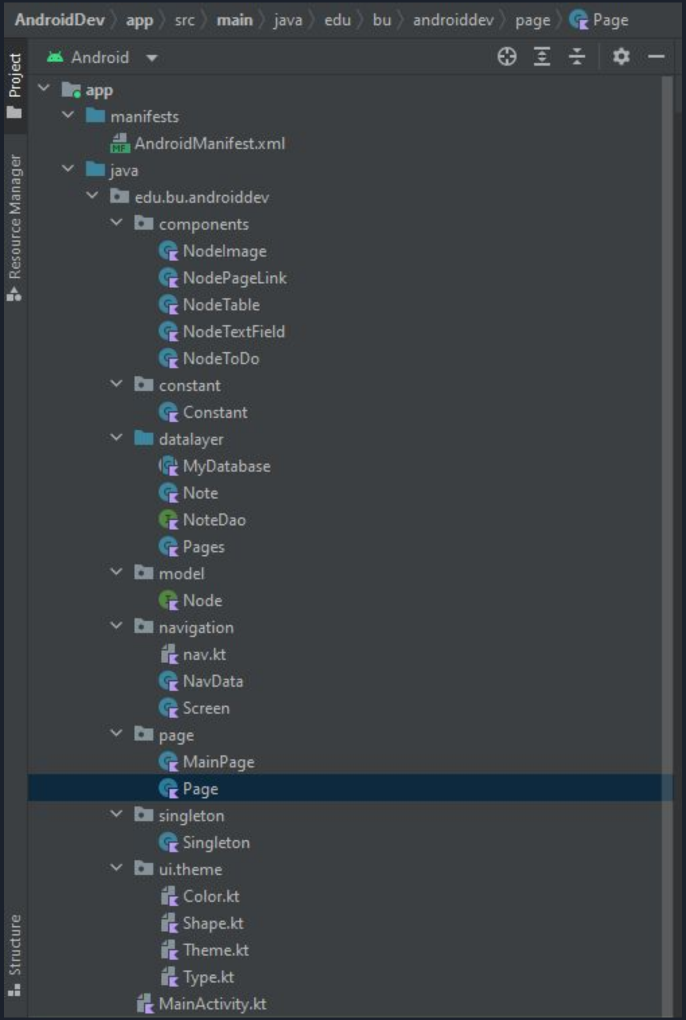
Constant - contain constants variables

Datalayer - contains code for instantiation of Room DB and related classes

- MyDatabase - initiates room db object
- Note - Data model class to store pages
- Pages - Data model class to store components data
- NoteDao - A Dao layer which contains operations for both Note and Pages data models.

Model - contain model classes or interfaces

- Node - Interface for the components. Contains functions for serialization and deserialization.



AndroidDev 〉 app 〉 src 〉 main 〉 java 〉 edu 〉 bu 〉 androiddev 〉 page 〉 Page

```
app
  manifests
    AndroidManifest.xml
  java
    edu.bu.androiddev
      components
        NodeImage
        NodePageLink
        NodeTable
        NodeTextField
        NodeToDo
      constant
        Constant
      datalayer
        MyDatabase
        Note
        NoteDao
        Pages
      model
        Node
      navigation
        nav.kt
        NavData
        Screen
      page
        MainPage
        Page
      singleton
        Singleton
      ui.theme
        Color.kt
        Shape.kt
        Theme.kt
        Type.kt
      MainActivity.kt
```

Project Structure

Navigation - contains navigation codes.

- Nav - initialises navcontroller
- navData - stores info of the url path

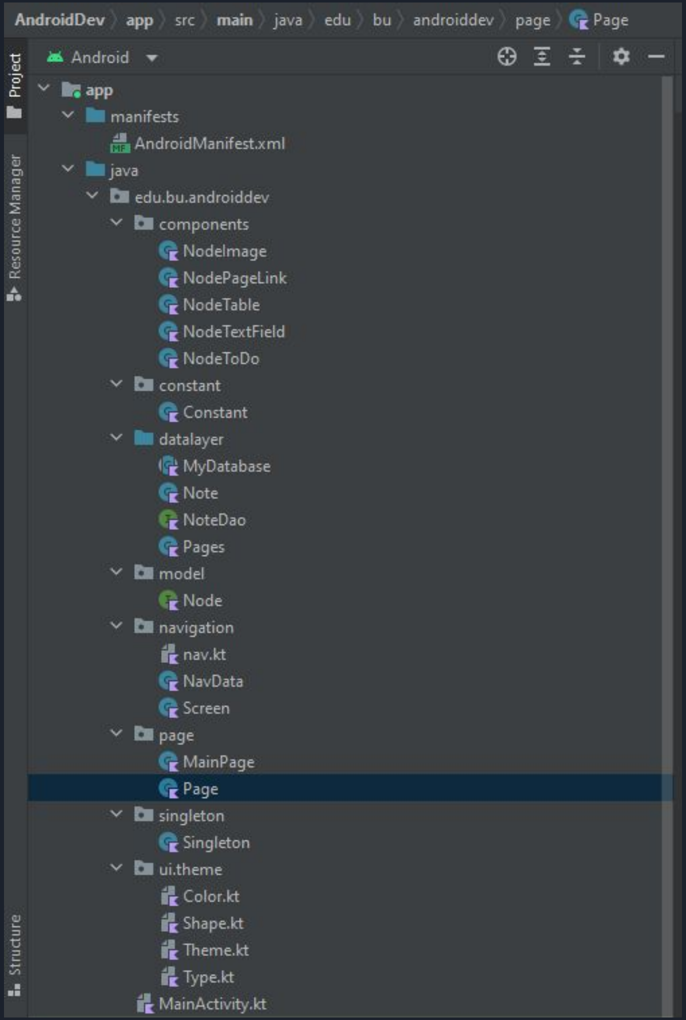
Page - contain ui which spans across the entire screen and are mapped to the navigation

- MainPage - First viewable screen to contain only redirect component
- Page - composable function which can hold all the components.

Singleton - contain statically declared objects so as to avoid redeclaring them

Ui.theme - default package

MainActivity - the app starts from here and it has only one activity.



AndroidDev 〉 app 〉 src 〉 main 〉 java 〉 edu 〉 bu 〉 androiddev 〉 page 〉 Page
Project
Resource Manager
Structure
Android

```
app
  manifests
    AndroidManifest.xml
  java
    edu.bu.androiddev
      components
        NodeImage
        NodePageLink
        NodeTable
        NodeTextField
        NodeToDo
      constant
        Constant
      datalayer
        MyDatabase
        Note
        NoteDao
        Pages
      model
        Node
      navigation
        nav.kt
        NavData
        Screen
      page
        MainPage
        Page
      singleton
        Singleton
      ui.theme
        Color.kt
        Shape.kt
        Theme.kt
        Type.kt
      MainActivity.kt
```

Permission

External storage access for importing images.

Future Scope

- Would like to improve on UI
- Drag capability
- Firebase Integration
- Notification System
- Complex components like kanban board