

# NeevaBrowser

---

## What is Included ?

---

### 1. Demo Video

- Shows functionality of opening URL when user types in edittext
- Regular search by typing in edittext – open google search
- Forward/back/home button functionality
- I have 5 hard-coded tabs only for this demo
- These tabs can be switched by tapping on tab thumbnail on top or swiping left or right
- Pull to refresh experience

### 2. Project code ZIP folder with README included in it.

## How to load project in Android studio

---

- Step 1 : Download the NeevaBrowser.zip folder
- Step 2 : Extract Zip folder
- Step 3 : Open Android Studio
- Step 4 : select option “Open an Existing Project”
- Step 5: Select NeevaBrowser > build.gradle file This may take couple of minutes to load and index the project

## How to run the application on Device/Simulator

---

Prerequisite : Project is loaded in Android studio Debug android device is connected to your computer - Step 1 : In top right corner of Android studio click on “Run app”

## How to run APK without Android studio setup

---

Prerequisite : You need to have ADB installed on your computer - Step 1 : download APK on your computer and navigate to the folder in your terminal - Step 2 : run the following command

```
adb install -r -f app-debug.apk
```

## Points to Highlight

---

1. Project completed in Kotlin & uses Jetpack libraries
2. Using MVVM architecture
3. Demonstrated Fragment/Activity relation
4. Used LiveData to automatically refresh the UX when data gets modified
5. Usage of DataBinding and ViewBinding
6. Good code review practice with inline comments

## What is next ?

---

1. Better UX – Design improvements
2. Application can be designed for different layouts for “portrait” & “Landscape” modes
3. Internationalization to support multiple language
4. Error handling for better user experience (wifi not connected and handle other scenarios)
5. Provide user tab experience (ex. google chrome browser) to pick selected tab
6. Other features can be added – bookmark, brows history social sharing etc
7. Add multiple logs throughout so it can become production ready
8. Unit tests, Instrumentation tests, Automation tests.