AATMANIRBHAR-SANCHAR(SECURE COMMUNICATIONS)

Cookbook for the academic year 2022-23

Aatmanirbhar Sanchar:
Self-Sufficient Communications

In partial fulfilment of the Fourth Year (Semester–VIII), Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2022-2023

Submitted by

Aashish Raheja (D17B, R.No.51)
Ashwin Pansare (D17B, R.No.43)
Kartikey Verma (D17B, R.No.65)
Karan Punjabi (D17B, R.No.49)

Project Mentor

Dr. Prashant Kanade, Prof Dr Shashikant Dugad, Mrs Yugchhaya Dhote (2022-23)

Table of Contents:

Application CookBook (ASSC)

AatmaNirbhar Sanchar:

Introduction:

“Aatmanirbhar Sanchar” aims at providing the users a real time off the grid, secure and anonymous messaging service. We have fulfilled these claims by not incorporating any third-party APIs or other services. Our application is a cross-platform ephemeral anonymous messaging service providing End to End encryption while also enabling secure transmission of any kind of data files including but not limited to images, videos, and documents. The most unique feature of our product would be the use of our very own indigenous private server and database system without the inclusion of any third-party cloud servers.

Requirements:

Server-Side:

OS: Ubuntu 20.04 (Recommended, although will work on any OS supporting python and Node Js)
Network: Require a Static IP Address with at least 5 open ports.
NPM: Version 6.14.14 (Steps explained ahead)
NodeJS: Version 14.17.5 (Steps explained ahead)

Files:

All the files are hosted on github at https://github.com/Aashish-100/Chat-application.git
To download files on your local machine you can either download the zip file available at github or use the “git” command to clone the above mentioned repository.

1) If using the “zip” method just extract the compressed files and move to the next section.

2) To clone using git, first download the git module using https://git-scm.com/downloads as per your OS.
    a) Now open cmd/terminal in the directory you want to install the project in and type the following command:

```
git clone https://github.com/Aashish-100/Chat-application.git
```

Steps to get started:

A. NodeJS

a. Ubuntu Setup

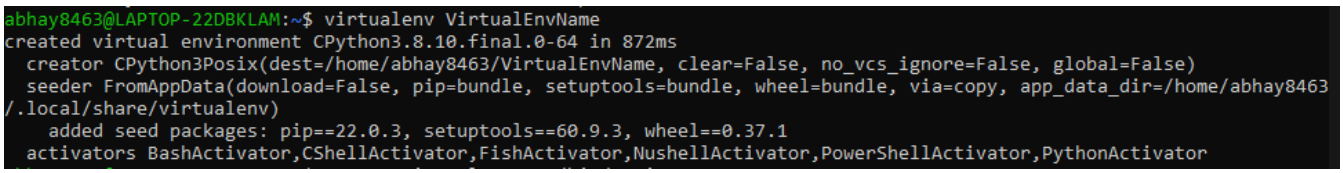
1. Setting Up Virtual Environment:

1. Installing Virtual Environment Manager: (Replace pip3 with pip if default python is version 3 and above.)

```
sudo pip3 install virtualenv
```

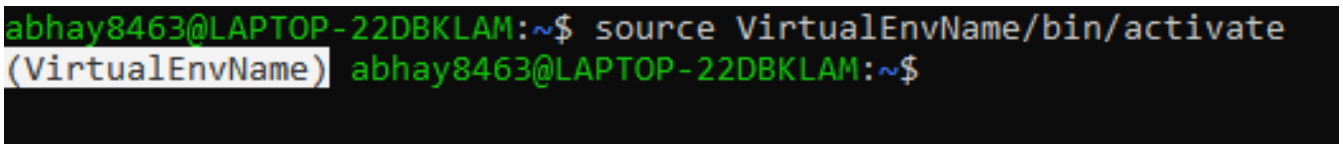
2. Creating a Virtual Environment: (You can use any name for the virtual environment instead of “VirtualEnvName”)

```
virtualenv VirtualEnvName
```



3. Activating the Virtual Environment

```
source VirtualEnvName/bin/activate
```

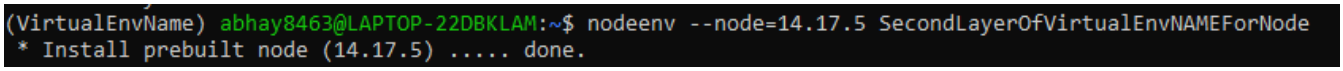


4. Installing Node manager and Virtual Env (nodeenv)

```
pip3 install nodeenv
```

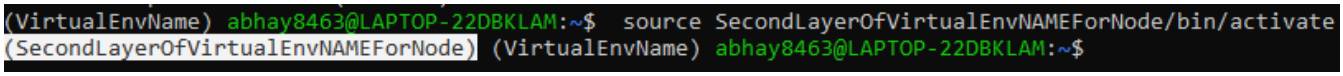
5. Creating a NodeJS Virtual Environment of version 14.17.5

```
nodeenv --node=14.17.5 SecondLayerOfVirtualEnvNAMEForNode
```



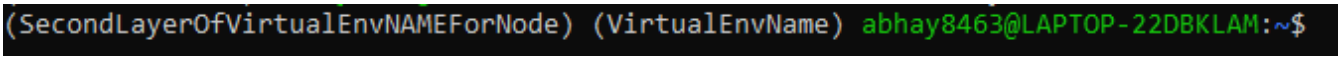
6. Activating the NodeJS Virtual Env

```
source SecondLayerOfVirtualEnvNAMEForNode/bin/activate
```



2. Checking the installations:

Before executing the following verification commands make sure both the virtual environment are active:



If not navigate to the directory you created the virtual environments in and then execute the following commands: (MAKE SURE THE ORDER IS CORRECT: First normal environment followed by node JS environment)

1. Activating the Virtual Environment

```
source VirtualEnvName/bin/activate
```

2. Activating the NodeJS Virtual Env

```
source SecondLayerOfVirtualEnvNAMEForNode/bin/activate
```

Once installed you can check the installation using the following commands:

1)

```
node -v
```

```
Output
v14.17.5
```

2)

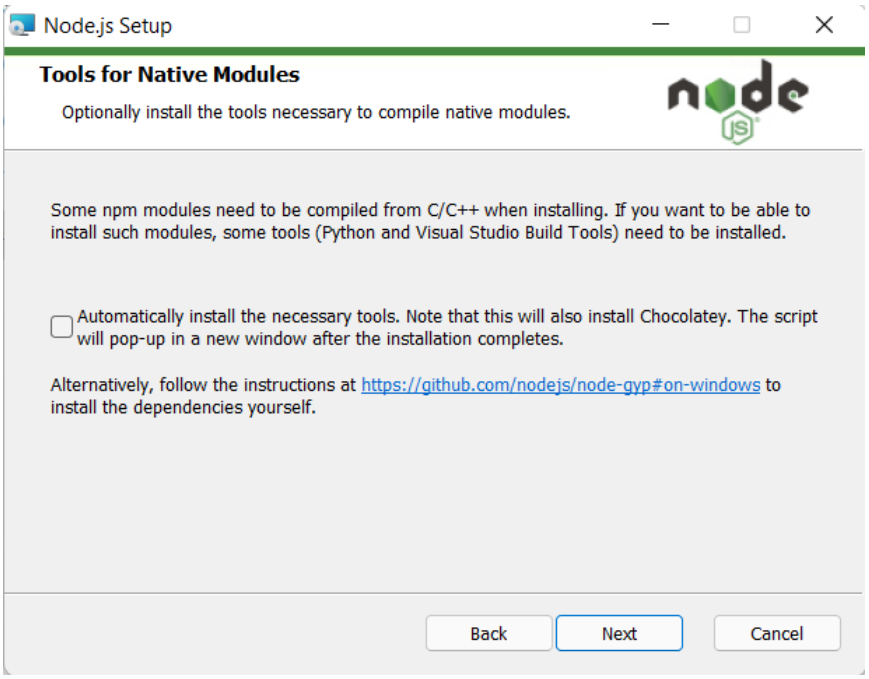
```
npm -v
```

```
Output
v6.14.14
```

b. Windows Setup

To download Node JS along with the supported NPM version visit https://nodejs.org/dist/v14.17.5/ and download the “msi” version depending on your system architecture (64 bit / 32 bit).

Open the downloaded executable and click on next until you see the following screen (Fig. n.) and untick the option as shown in the image. (You may use any directory to install the same.)



Then click on Next and let it automatically install on your PC.

Once installed you can check the installation using the following commands:

1)

```
node -v
```

Output

v14.17.5

2)

```
npm -v
```

Output

v6.14.14

B. Install the Files

Make sure you have downloaded all the files as instructed above and have opened a command prompt in the same directory.

C. Steps before running the application (ONLY UBUNTU)

➢ **If using Virtual Environments, make sure both the virtual environments are active:**

If not navigate to the directory you created the virtual environments in and then execute the following commands: (MAKE SURE THE ORDER IS CORRECT: First normal environment followed by node JS environment)

1. Activating the Virtual Environment

```
source VirtualEnvName/bin/activate
```

2. Activating the NodeJS Virtual Env

```
source SecondLayerOfVirtualEnvNAMEForNode/bin/activate
```

D. Installing and Running the application for the first time(Common for Ubuntu and Windows):

1) **Once inside the directory where the files were cloned or extracted, Install all the dependencies using the following command:**

```
npm install
```

2) **Now open the "config.js" file and edit the following fields:**

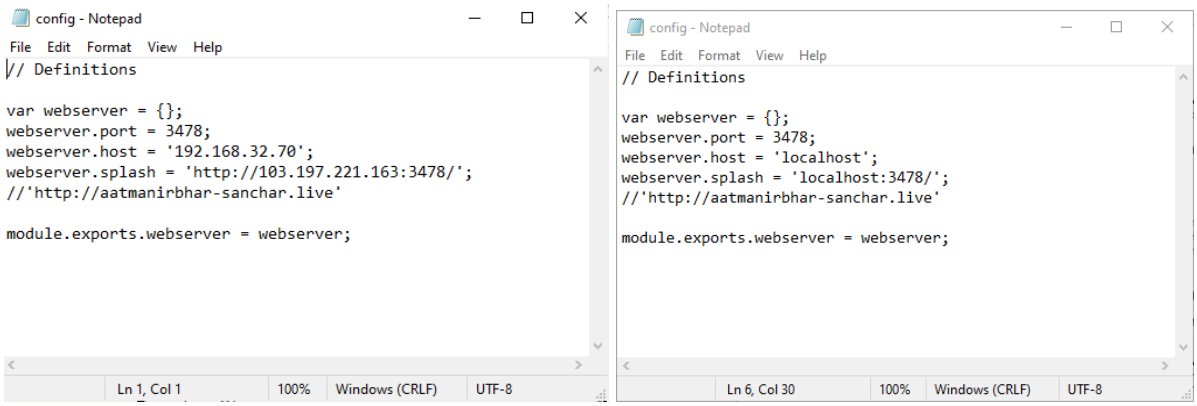
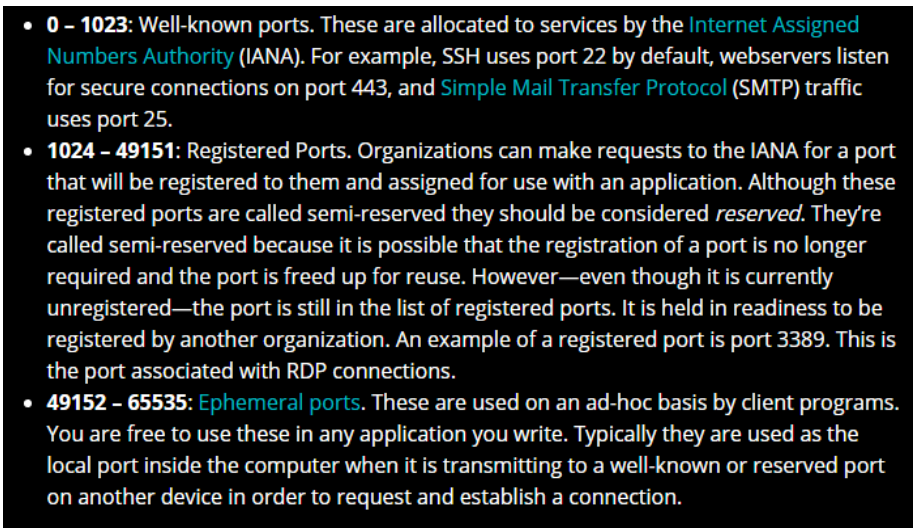


Fig. **Config.js** [when hosted on **external** ip(**LEFT**)] |||||| [when **locally** hosted (**RIGHT**)]

a) webserver.port = “3478”
This field defines the port on which the web-application will be running. (If using a static IP make sure you only use ports which are INBOUND open.
To check if your port is open visit: https://www.yougetsignal.com/tools/open-ports/
NOTE: For the public static IP address, your Network admin will have to ask the Internet Service Provider who will provide a public static IP address and the private IP address can then be allocated by the network admin for the local network. If using a public static IP address, MAKE SURE the port you are using is a publicly INBOUND open port. Consult your Network admin and ask him to open a port for you if not already open)

NOTE: It will only show whether a port is open or not, once the server has started.) (Talk with your network admin or ISP if the ports are not available.)

NOTE: Don’t use important ports for this (Recommended use range 1024-49151). REFER: https://www.cloudsavvyit.com/8844/why-are-some-ports-risky-and-how-do-you-secure-them/ for more information.



b) webserver.host = “localhost”
Replace the word “localhost” with your internal/private IP address [use ipconfig(windows) or ifconfig(ubuntu) to see your internal Ip address] of the computer if hosting on a public static IP address else keep the word localhost.

c) webserver.splash = “localhost:3478”
This acts as a default web url to go to if any error occurs.
This will be replaced by your external IP address along with the port number if hosted on an external public IP address.
For Eg: webserver.splash = 'http://103.197.221.163:3478/'

3) Now save the edited “config.js”.

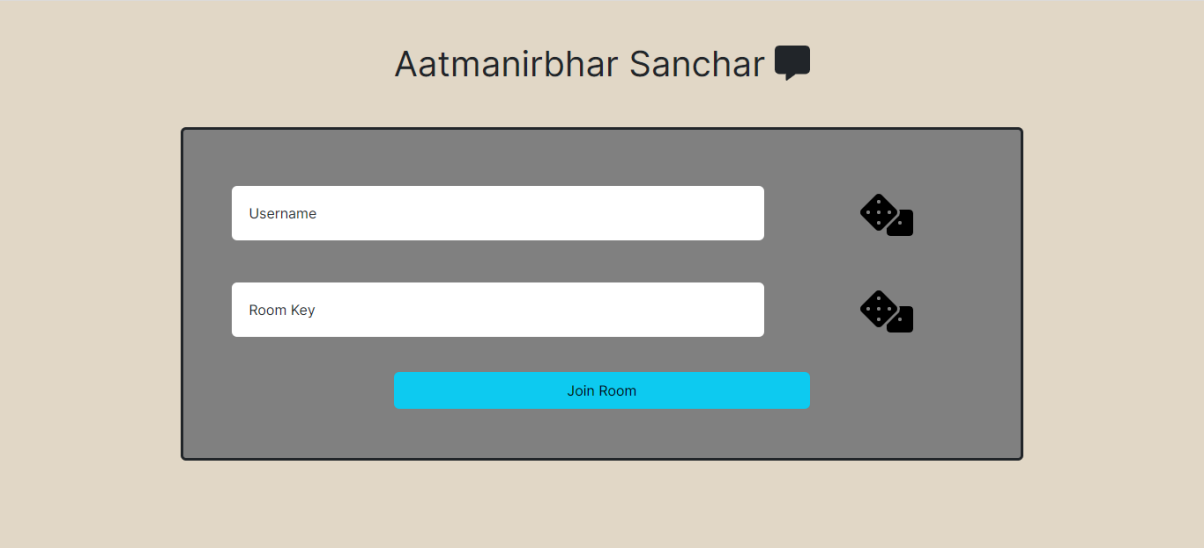
4) Now, we launch the actual app:
NOTE: Make sure you are inside the project folder while executing the following command.

```
npm start
```

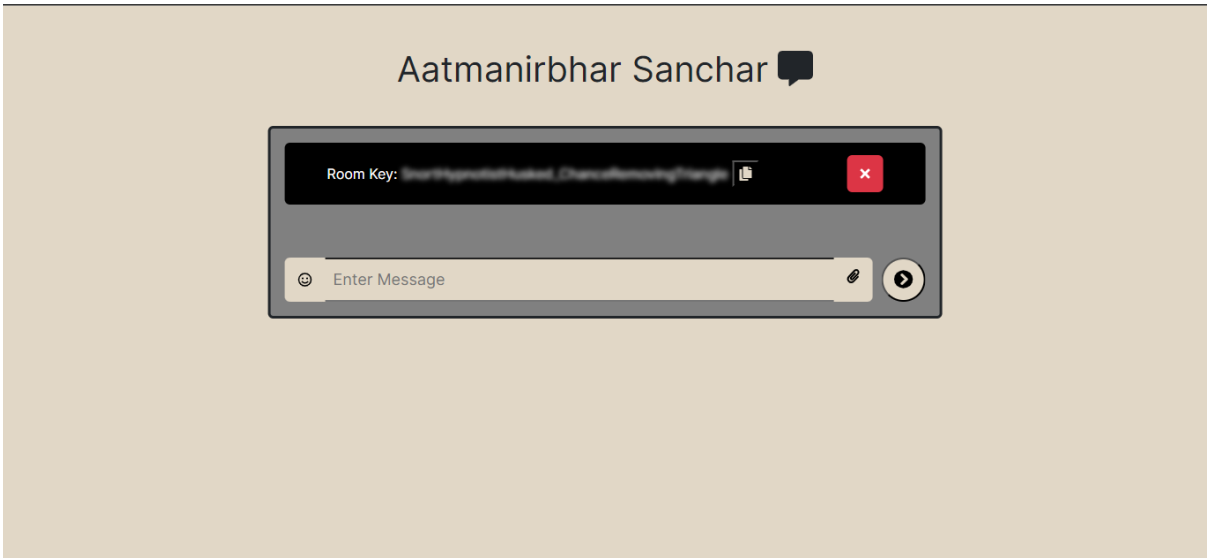
5) **Now you may visit your respective hosted web address.**

   a) **Local host:**
   Visit http://localhost:5000/

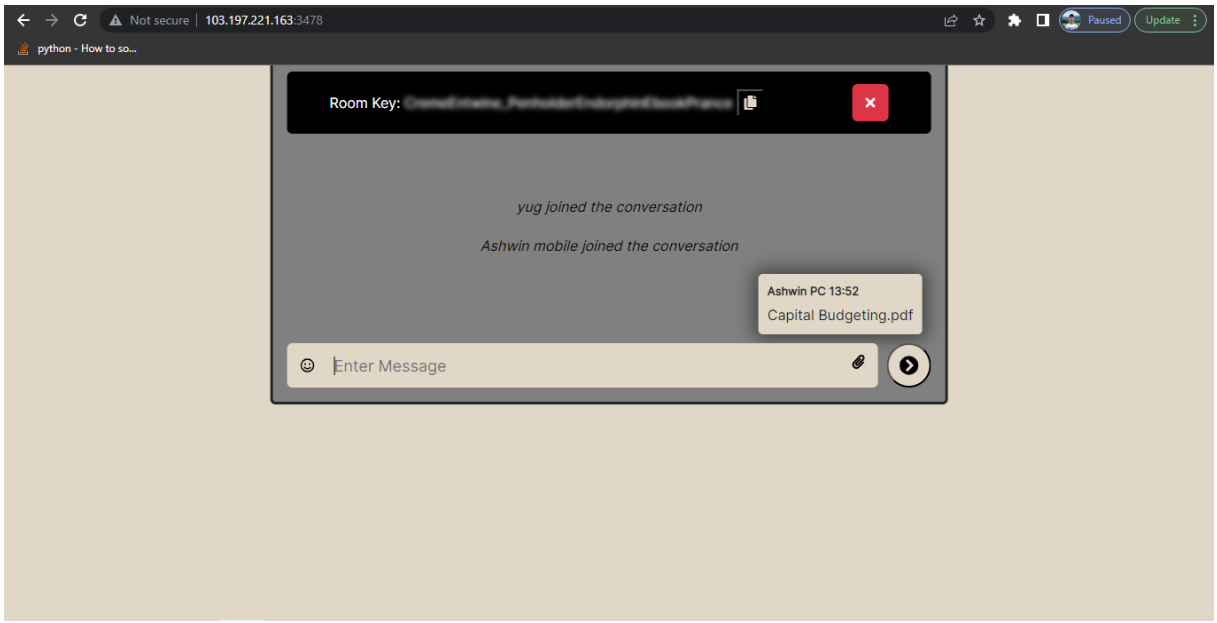
   b) **Publicly hosted on external IP address**
   You may now visit the "externalIpaddress:portNumber" from any internet connected device.
   For eg: http://103.197.221.163:3478/

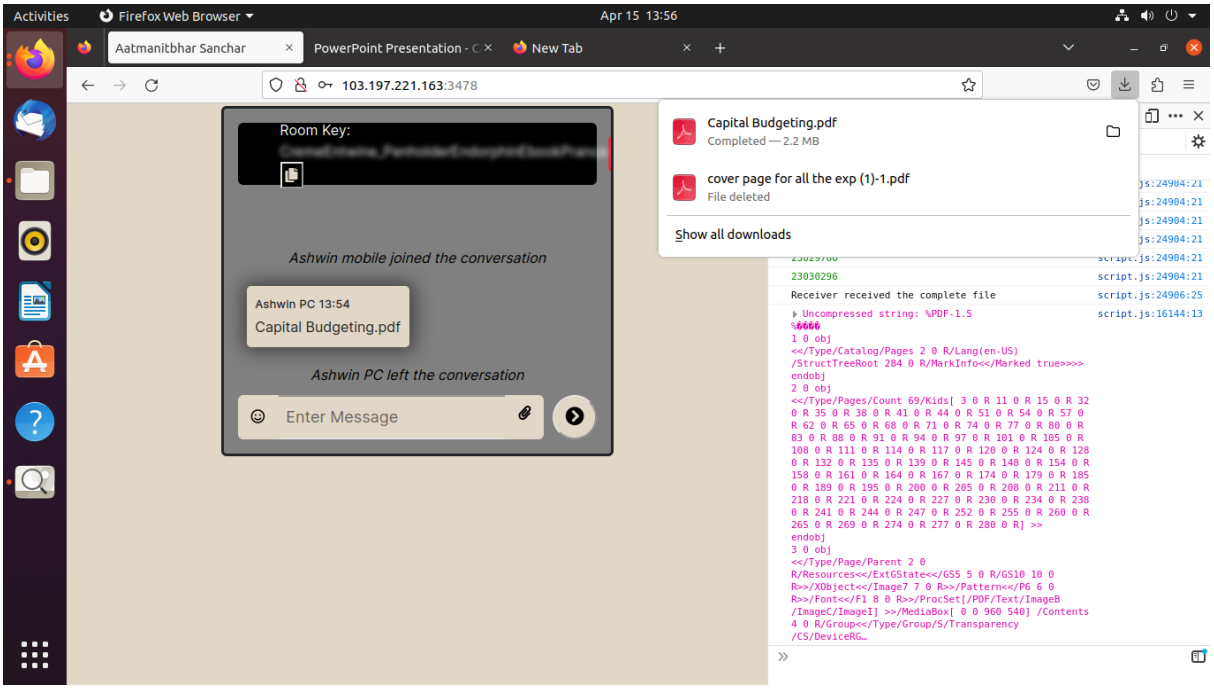


6) **To test if the installation is successful visit the hosted site and join a room.** You will see the following page after joining a room:



7) **File Transfer:**
   We support all kinds of file transfer (Video, Audio, Images and any other form)





E. Running the application later on(After the initial setup):

a. Make sure you are inside the project folder.

b. Open a terminal and run the following commands:
(NOTE IF ON UBUNTU: Make sure both the virtual Environments are running using the steps mentioned before).

```
npm start
```

```
student@VESIT307-2:~/D17B-tifr/AatmaSanchar/AatmaSanchar-final/Aatmanirbhar-Sanc
har-main/api-2.4/api-2.4$ npm start

> @ start /home/student/D17B-tifr/AatmaSanchar/AatmaSanchar-final/Aatmanirbhar-S
anchar-main/api-2.4/api-2.4
> node server.js

listening on *:3478
```

c. Now that the server is up and running the chat application can be accessed at "http://localhost:3478"
d. NOTE: if you used a static ip address with open INBOUND ports it can be accessed from anywhere at "*http://your.public.ip:PortNumber*"

F. Automating server start on boot-up (Ubuntu):

All the steps for starting the server can be automated to run on machine start up

a. **Creating a bash script file:**

1. Open a terminal and navigate to the directory where you want to create the shell script.
2. Create a new file using the touch command:
   touch start_server.sh
3. Open the file using a text editor such as nano or vim:
   nano start_server.sh
4. Add the following lines to the file:

```
#!/bin/bash

# Activate the Python virtual environment
source PythonVirtualenv/bin/activate

# Activate the Node JS virtual environment
source NodeVirtualenv/bin/activate

# Navigate to the project directory
cd /home/username/Desktop/Chat-application

# Start the Node JS server using PM2
pm2 start pm2server.json
```

5. Save the file using Ctrl+X and exit the text editor.
6. Make the file executable by running the following command:
   chmod +x start_server.sh

b. **Use the crontab to run this script on machine startup**

1. Open the crontab editor by running the command crontab -e in the terminal.
2. Add the following line at the end of the file:
   @reboot /home/username/Desktop/Chat-application/start_server.sh

3. Save the file using Ctrl+X and exit the editor.

This will start the application server automatically on machine startup without the need of typing manual commands.

G. Understanding the Backend

To understand in detail how the backend (communication and encryption) works please refer to our Research paper.

Android Studio Setup.

Android Studio Windows Setup:

1. Download “Android Studio Arctic Fox (2020.3.1) Patch 3” Setup EXE from https://redirector.gvt1.com/edgedl/android/studio/install/2020.3.1.25/android-studio-2020.3.1.25-windows.exe
2. If the above link doesn't work visit https://developer.android.com/studio/archive to download the setup.
3. Once the site is loaded, scroll down and click on “I agree to the terms button” and then proceed to search and download the “Android Studio Arctic Fox (2020.3.1) Patch 3” EXE file.

Running the Setup

1. Run the downloaded EXE file and proceed with all the default options. You may change the install directory if you wish.

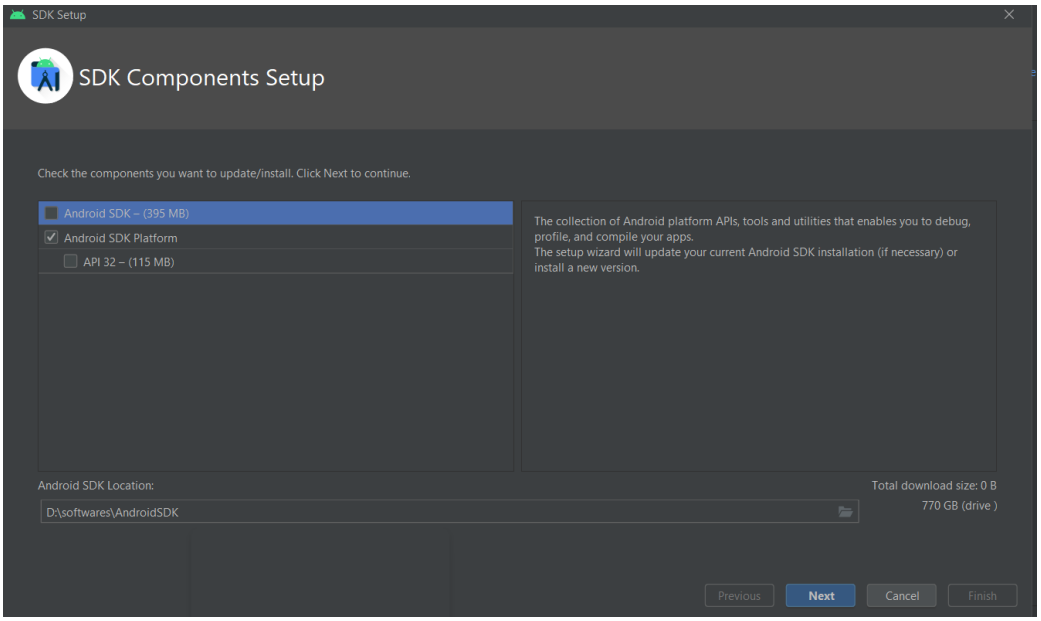
Android Studio Ubuntu Setup:

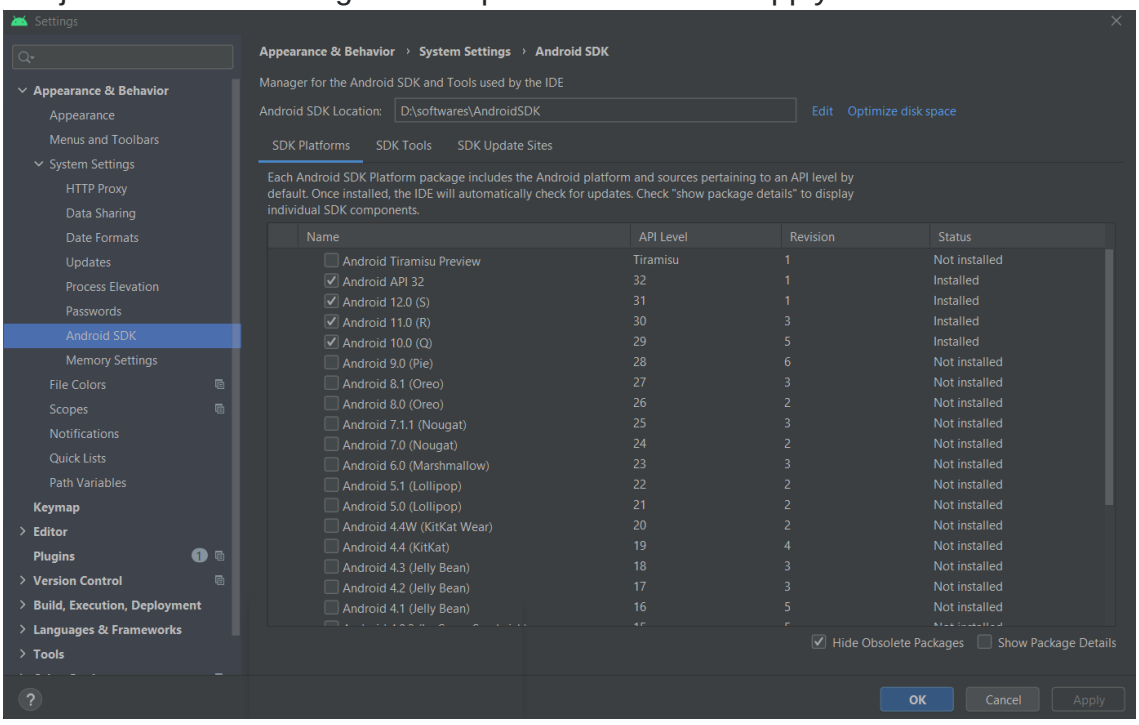
1. Download “Android Studio Arctic Fox (2020.3.1) Patch 3” Setup TAR.gz from https://redirector.gvt1.com/edgedl/android/studio/ide-zips/2020.3.1.25/android-studio-2020.3.1.25-linux.tar.gz
2. If the above link doesn't work visit https://developer.android.com/studio/archive to download the setup.
3. Once the site is loaded, scroll down and click on “I agree to the terms button” and then proceed to search and download the “Android Studio Arctic Fox (2020.3.1) Patch 3” TAR.gz file.

Running the Setup

1. Extract the downloaded Tar.gz using “tar xf filename.tar.gzandroid-studio-2020.3.1.25-linux.tar.gz” . Now, navigate inside the directory till you reach the “bin” folder.
2. Once inside the bin folder, right click and open a terminal at the current path and write “**./studio.sh**”.
3. This will start android studio and proceed with the default steps.

Download the Android SDK

1. (NOTE: You may skip to the next step if you already have Android SDK installed) Once the project is fully indexed and opened. Go to File > Settings > Appearance & Behavior > System Settings > Android SDK
2. If a SDK does not exist click on edit > Select a new SDK directory > Click apply.
   a. 

3. Once installed, make sure the following sub SDK components are installed as well, if not just tick the following shown options and click on apply.
   a. 

Open the Aatmanirbhar Sanchar Project

1. Download the Aatmanirbhar Samakraman Project from https://github.com/JayJhaveri1906/Aatmanirbhar-Sanchar-App.git . You may git clone or Download and extract the zip file.
2. Once downloaded, open android studio and select Open Project.
3. Now, browse to the downloaded Sanchar project and open it.

Building & Installing the Sanchar Android App.

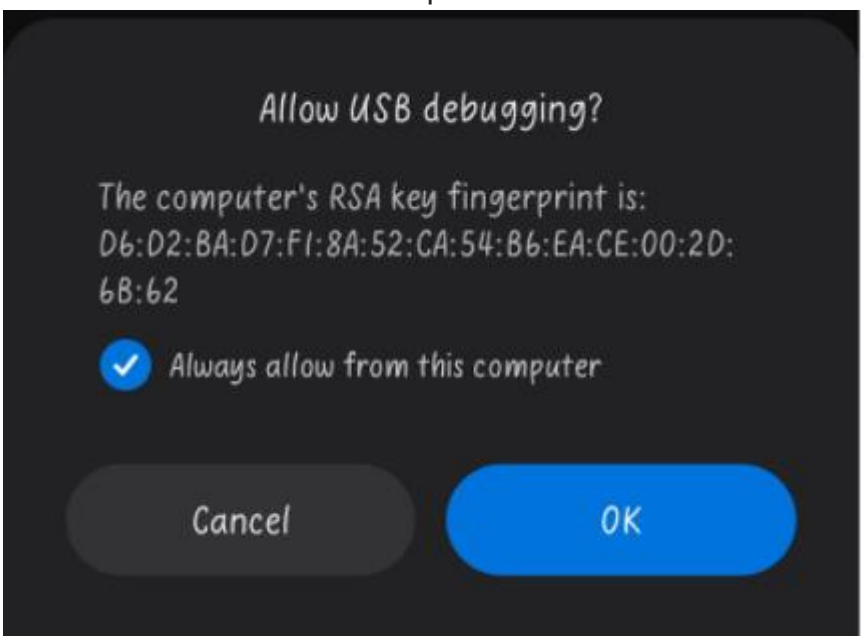
Building the APK

1. Follow these steps to change the server URLs: Navigate to app/src/main/java/com/example/AatmaSanchar/MainActivity.java:
    a. At line number 29: change the url variable value to your default public server address along with the port number attached.

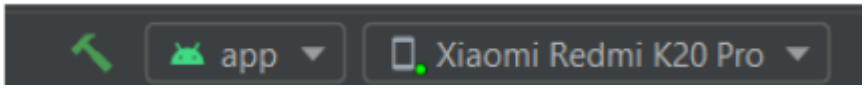
```
25  public class MainActivity extends AppCompatActivity {
26
27      private Button button;
28      private Bitmap mActionCallIcon, mActionCallLightIcon,
29      public String url = "http://158.144.55.73:3478";
```

2. Before building:
    a. Now on your Android device.
    b. Open the Settings app.
    c. Select System.
    d. Then click About phone.
    e. click Build number 7 times.
    f. Return to the previous screen to find Developer options near the bottom.
    g. Scroll down and enable USB debugging.
    h. The above steps from 1-6 are one time steps.
    i. Connect an android device to a laptop via USB.

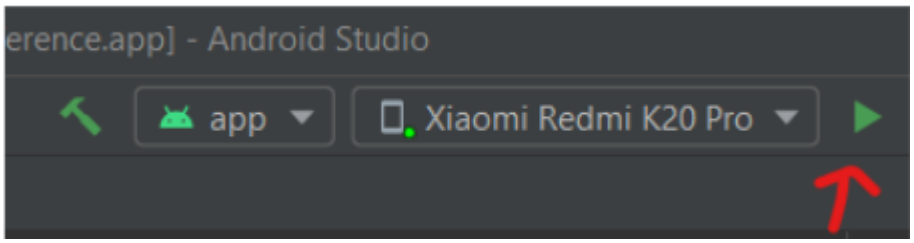
j. After this a pop-up message will appear stating to trust the computer, click on “OK” to continue with the operation.



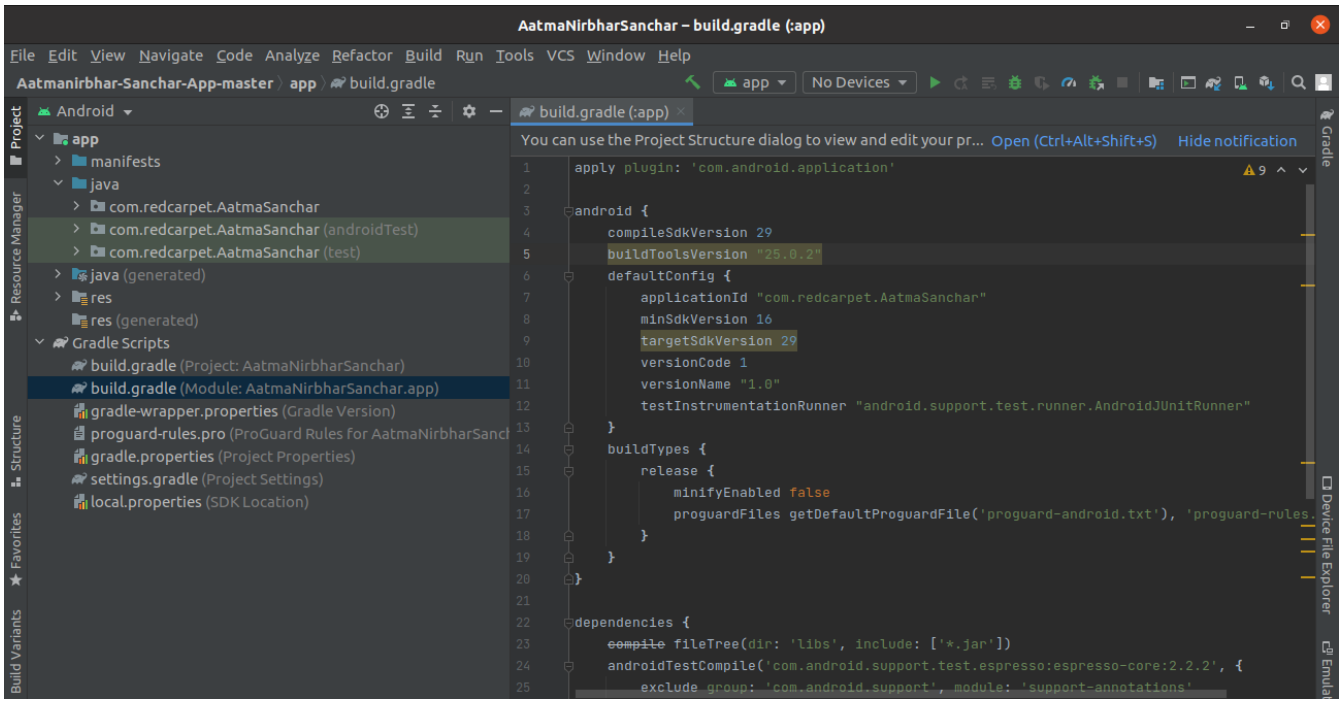
k. Now android studio will show your mobile name like this:



3. Now you may click the Play button at the top of android studio to run the application on your Android device.

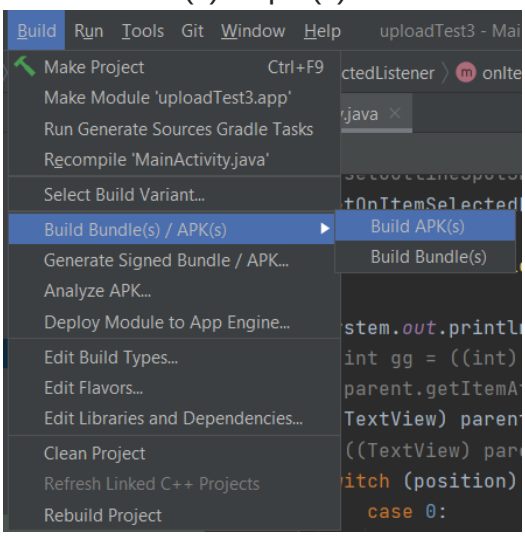


4. If the following error occurs:
   The minCompileSdk (<version>) specified in a dependency's AAR metadata (META-INF/com/android/build/gradle/aar-metadata.properties) is greater than this module's compileSdkVersion

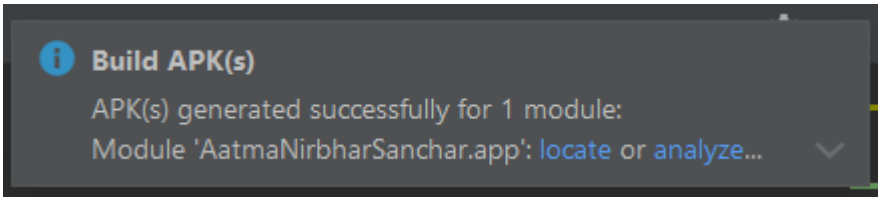


Set both compileSdkVersion and targetSdkVersion to <version> in your build.gradle(app) file.

5. Now to build an APK, select the build button at the top of the screen. Navigate to Build Bundle(s) / Apk (s) > Build APK (s).



6. A notification will pop-up in some time stating that the APK is ready and you can click “Locate it” and share the generated APK file using ***Aatmanirbhar Sanchar*** on your phone ;)



(NOTE: If you miss the notification you can click the “event log” button at the bottom right of your screen to see the notification history)

Running the Sanchar Application

Home Page

1. Press the “Open aatmanirbhar sanchar” button to launch the chat application. **NOTE:** The rest of the chat application works exactly the same way as described before in the web App version.

References

[1] Cohn-Gordon, K., Cremers, C., Dowling, B. et al. A Formal Security Analysis of the Signal Messaging Protocol. J Cryptol 33, 1914–1983 (2020). https://doi.org/10.1007/s00145-020-09360-1

[2] Raman Singh, Hitesh Tewari et. al "Blockchain-Enabled End-to-End Encryption for Instant Messaging Applications" https://arxiv.org/abs/2104.08494

[3] Botha, J.G., Van 't Wout, M.C. and Leenen, L. 2019. A comparison of chat applications in terms of security and privacy. 18th European Conference on Cyber Warfare and Security, University of Coimbra, Portugal, 4-5 July 2019

[4] Sabah, Noor & Mohamad, Jamal & Dhannoon, Ban N.. (2017). Developing an End-to-End Secure Chat Application. 17.

[5] M. B. Kılıç , "Encryption Methods and Comparison of Popular Chat Applications", Advances in Artificial Intelligence Research, vol. 1, no. 2, pp. 52-59, Sep. 2021

[6] Whatsapp Whitepaper. (2021). Whatsapp. https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf

[7] *Specifications >> The X3DH Key Agreement Protocol*. (2016). Signal Messenger. https://signal.org/docs/specifications/x3dh/

[8] Rijndael, "*AES-256 bit Whitepaper. (2020)",* https://www.cs.miami.edu/home/burt/learning/Csc688.012/rijndael/rijndael_doc_V2.pdf

[9] Double Ratchet and Diffie Hellman algorithms, https://signal.org/docs/specifications/doubleratchet/

[10] HMAC, http://pssic.free.fr/Extra%20Reading/SEC+/SEC+/hmac-cb.pdf

[11] Encryption and Cyber Security for Mobile Electronic Communication Devices. (2018, December 20). Federal Bureau of Investigation. https://www.fbi.gov/news/testimony/encryption-and-cyber-security-for-mobile-electronic-communication-devices

[12] Kseniia Ermoshina, Francesca Musiani, Harry Halpin. "End-to-End Encrypted Messaging Protocols: An Overview". Internet Science. INSCI 2016. Lecture Notes in Computer Science(), vol 9934. Springer, Cham.

[13] Discussion on projects using XMPP-based Instant Messaging on https://xmpp.org/uses/instant-messaging/

[14] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," in Cryptography and Coding: 6th IMA International Conference Cirencester, UK, December 17–19, 1997 Proceedings, 1997.