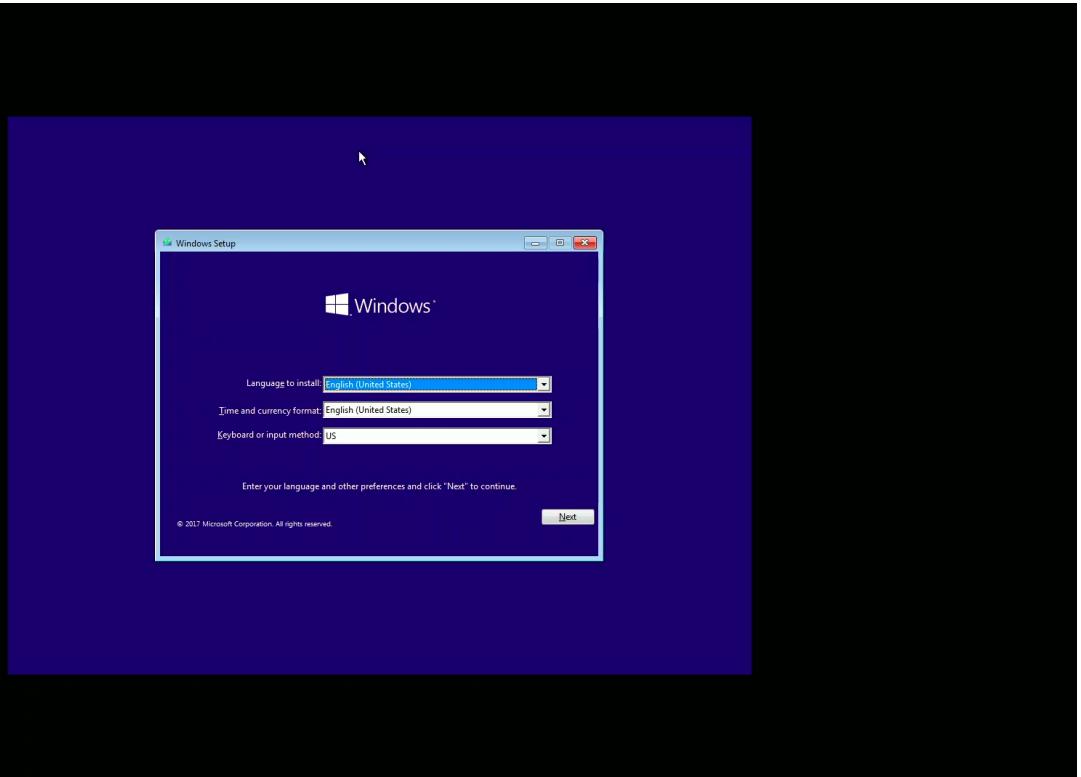


## Practical – 1

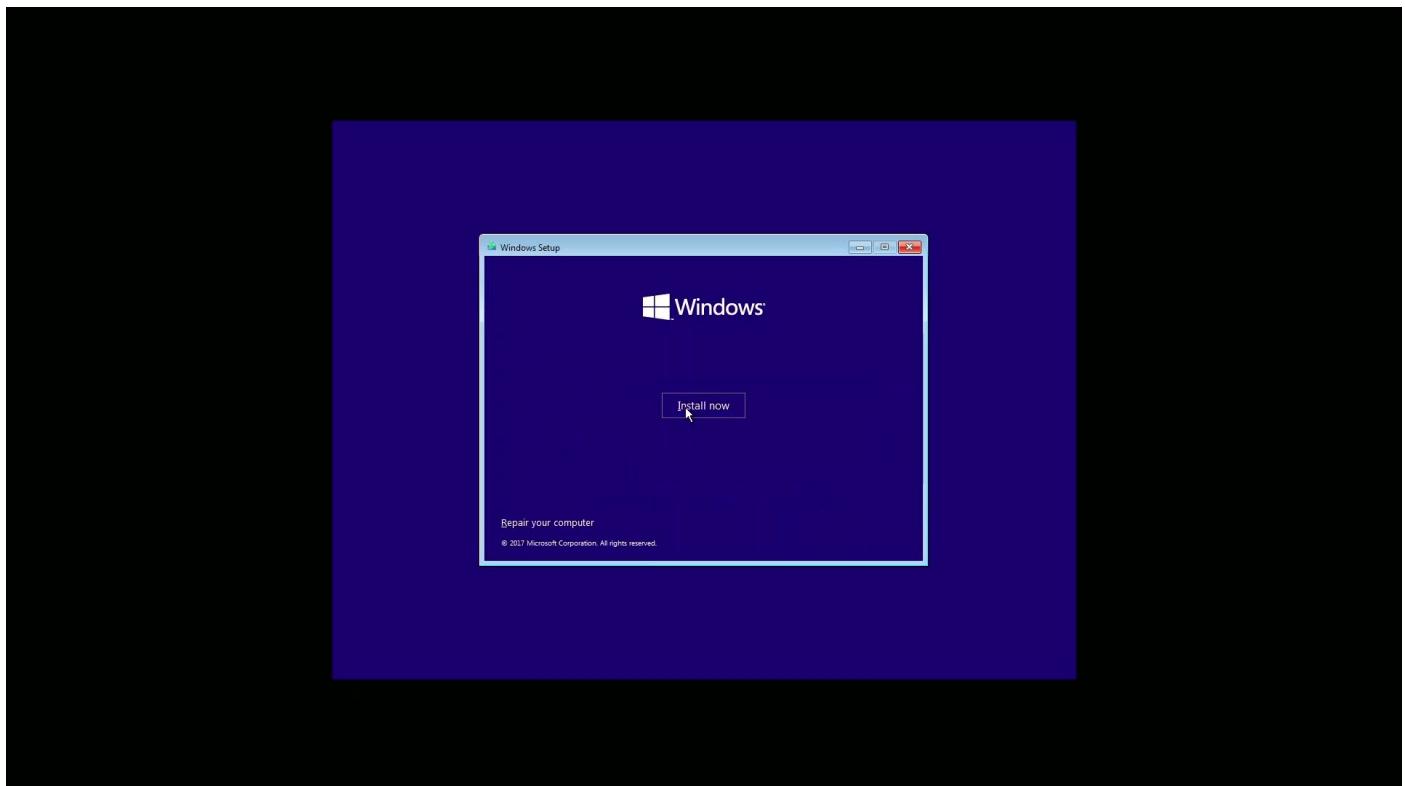
Aim: Install Multiple operating system in virtual machine.

Windows 10:

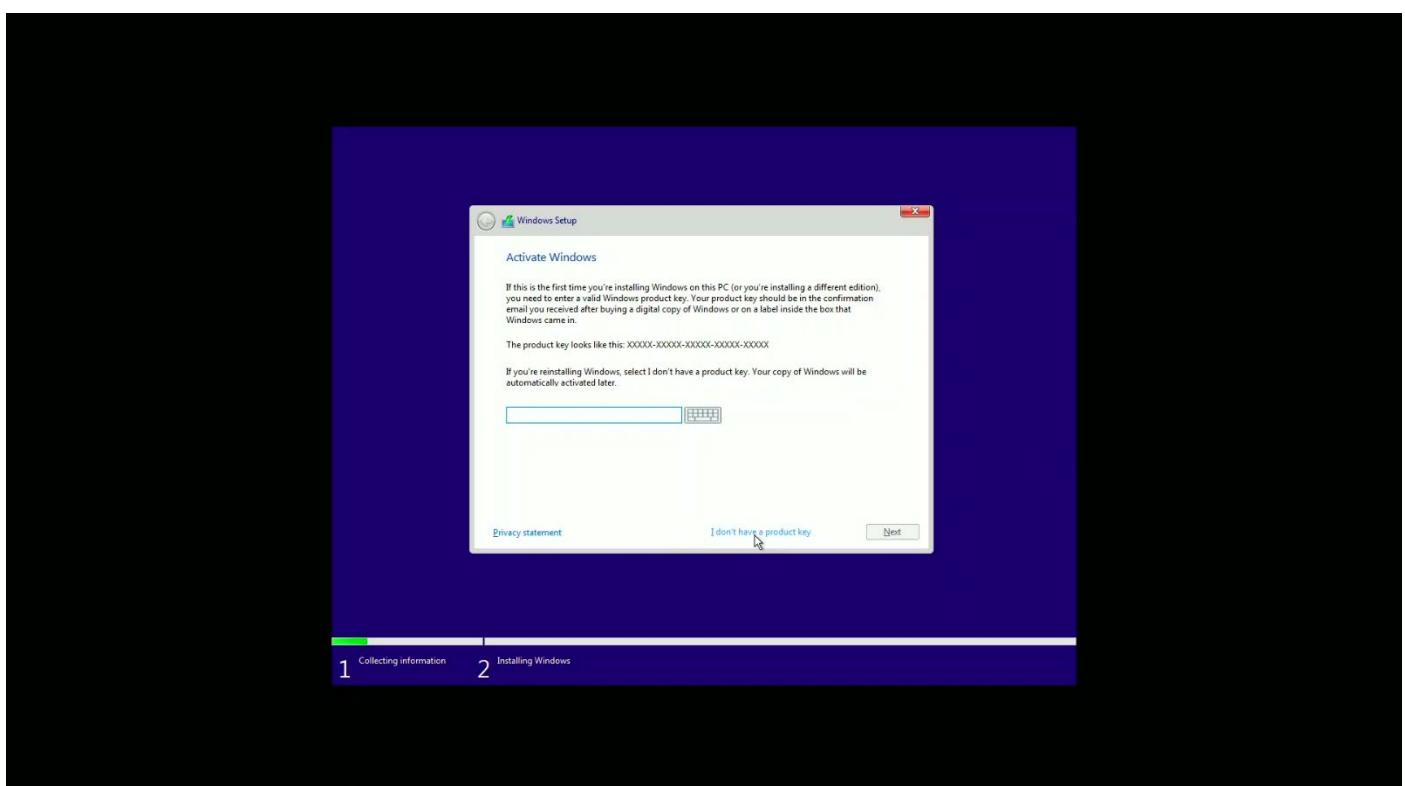
Step 1: Select Language, Time Zone and Keyboard Layout and click “Next”.



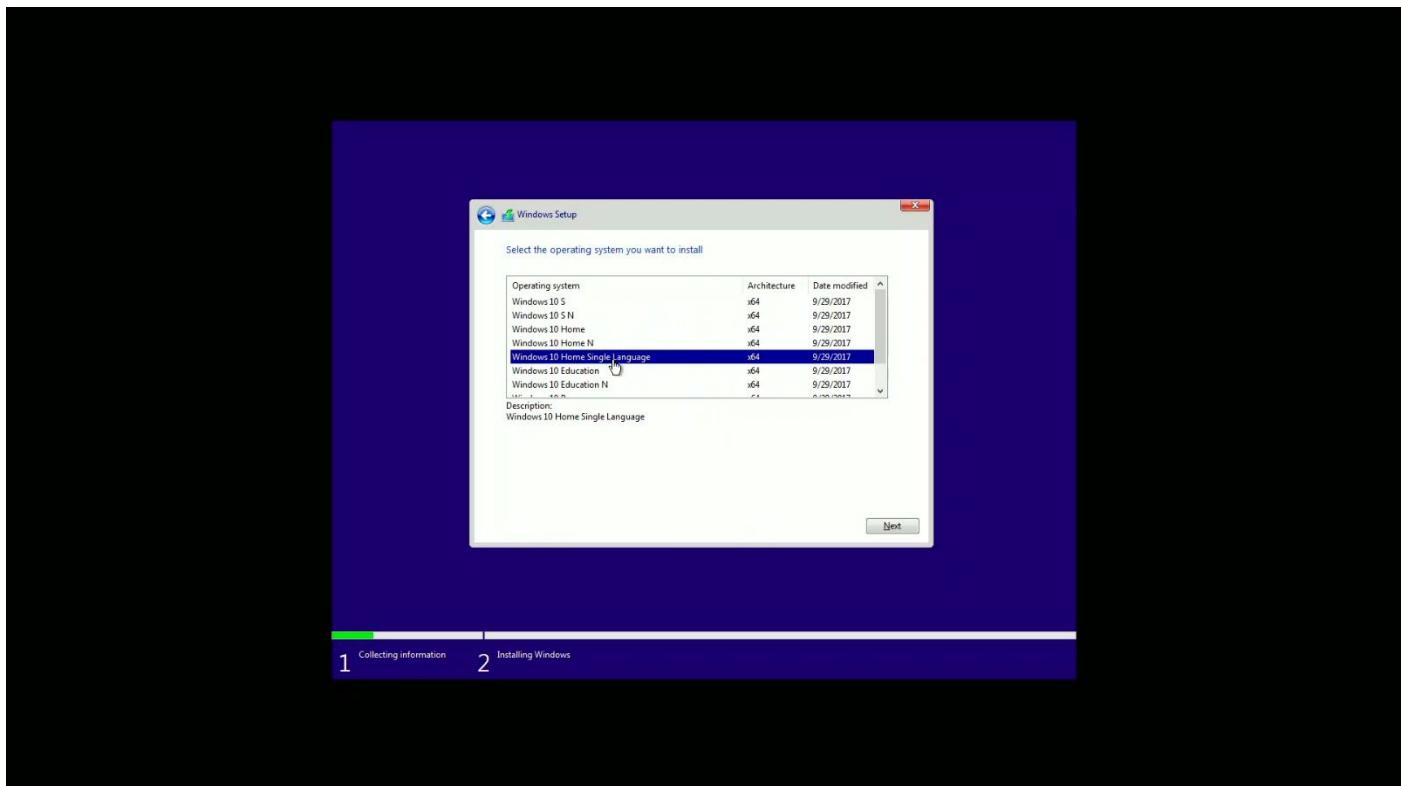
Step 2: Click “Install Now”



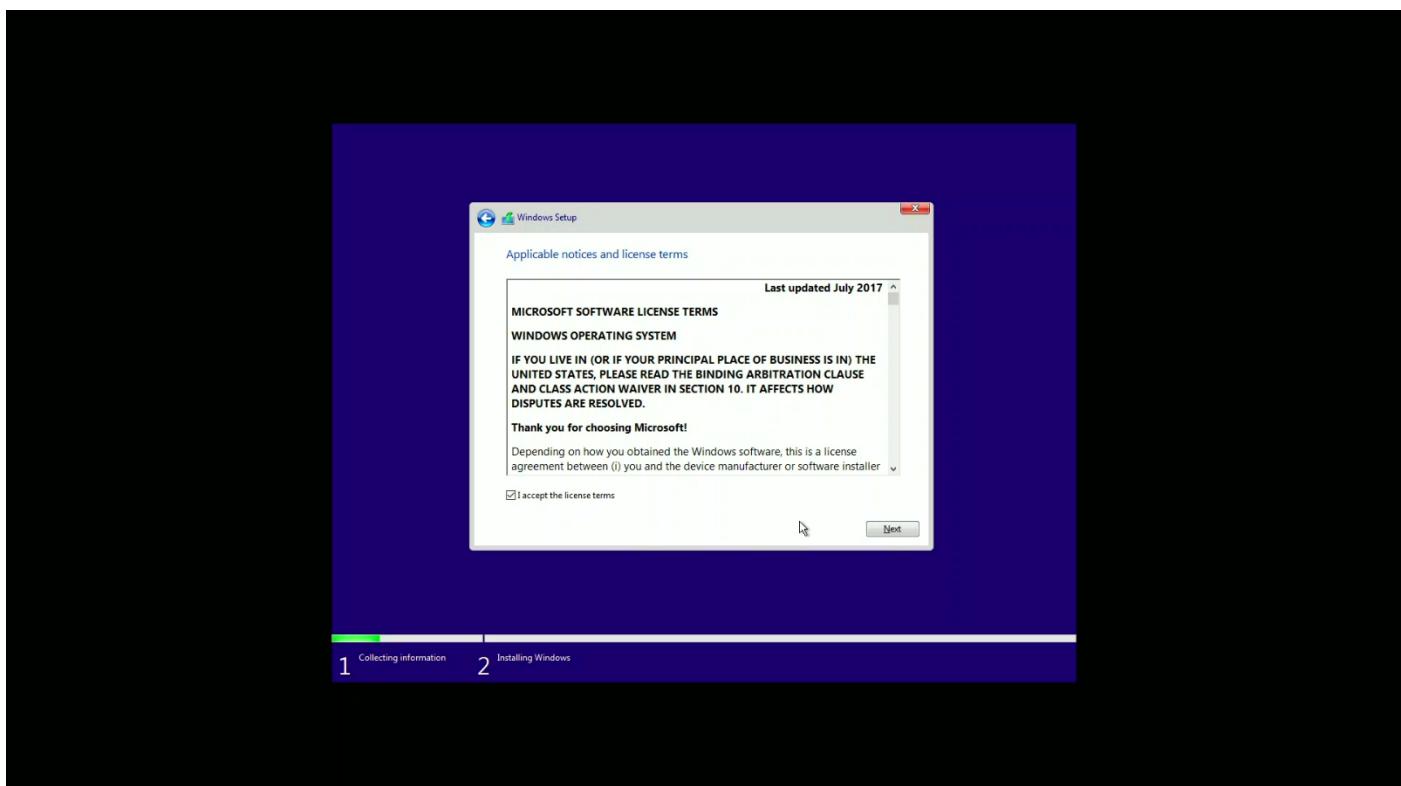
Step 3: Click “I don’t have product key”.



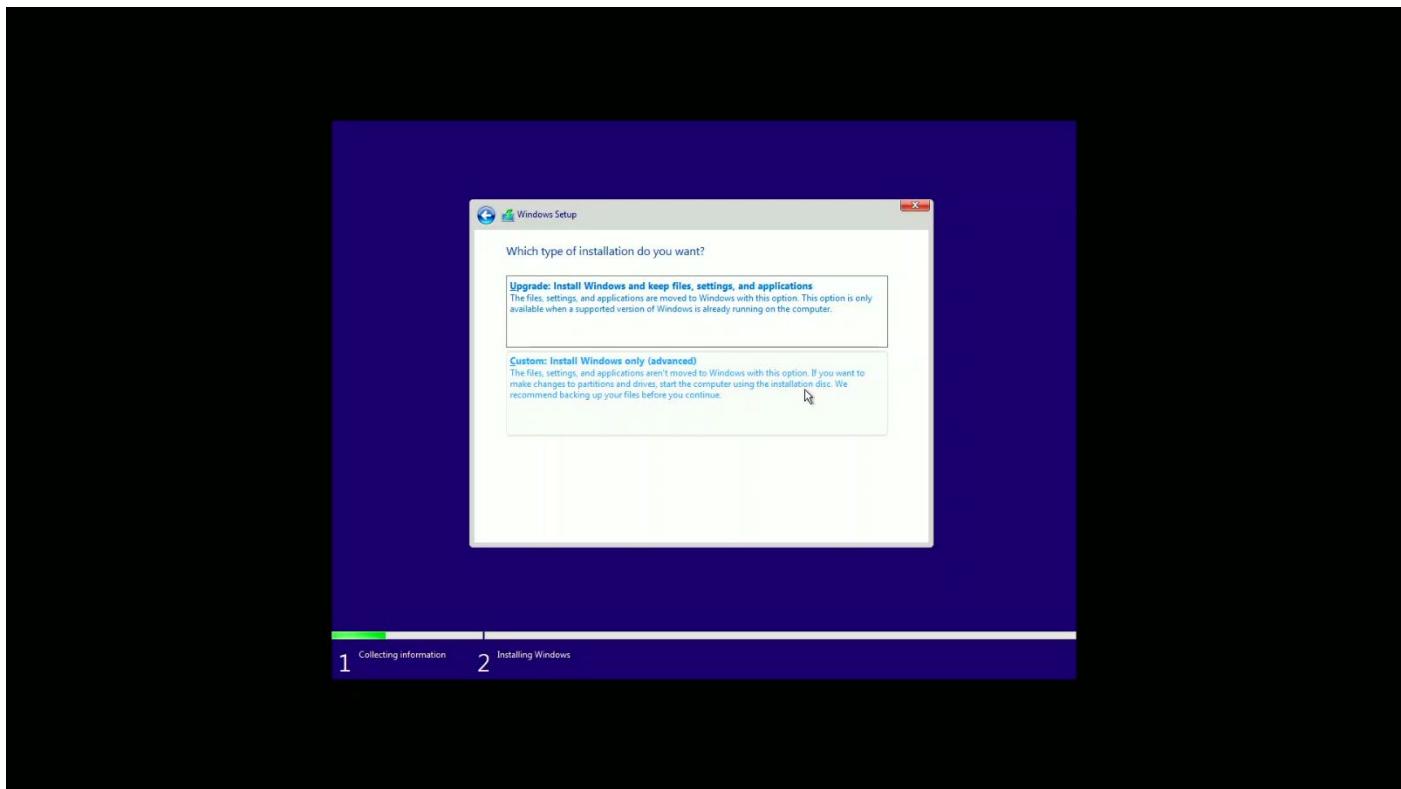
Step 4: Select Windows version and click “Next”.



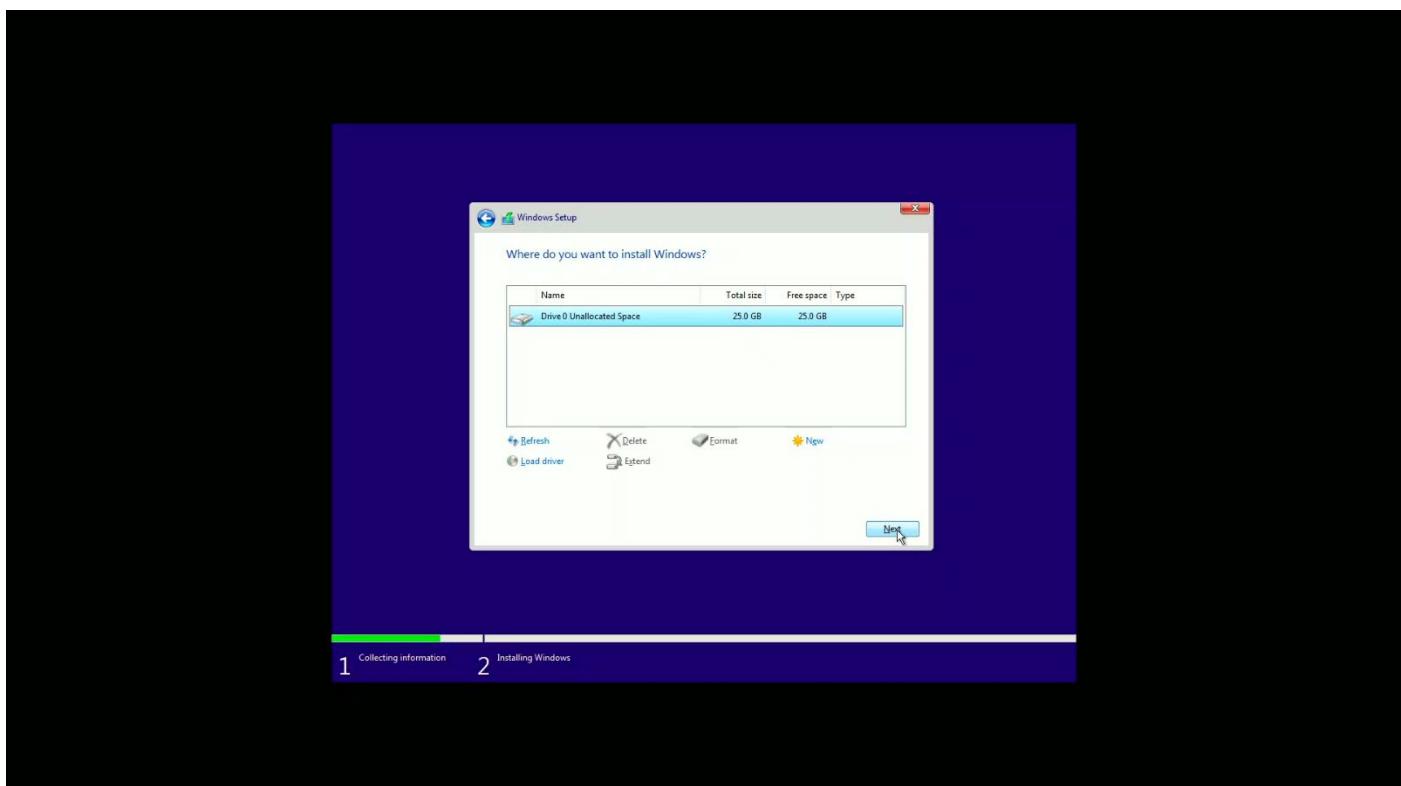
Step 5: Accept terms and conditions and click “Next”.



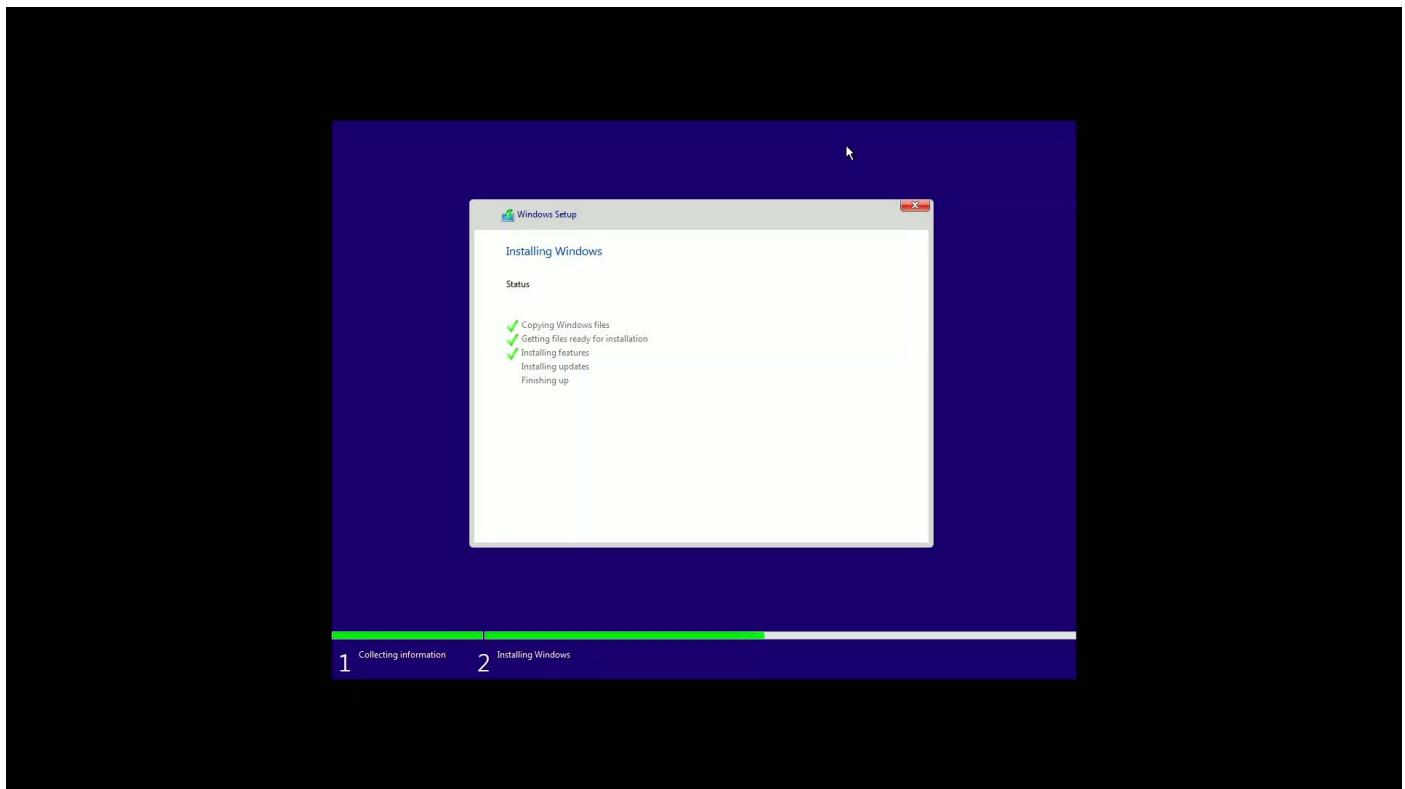
Step 6: Click “Custom: Install Windows only”.



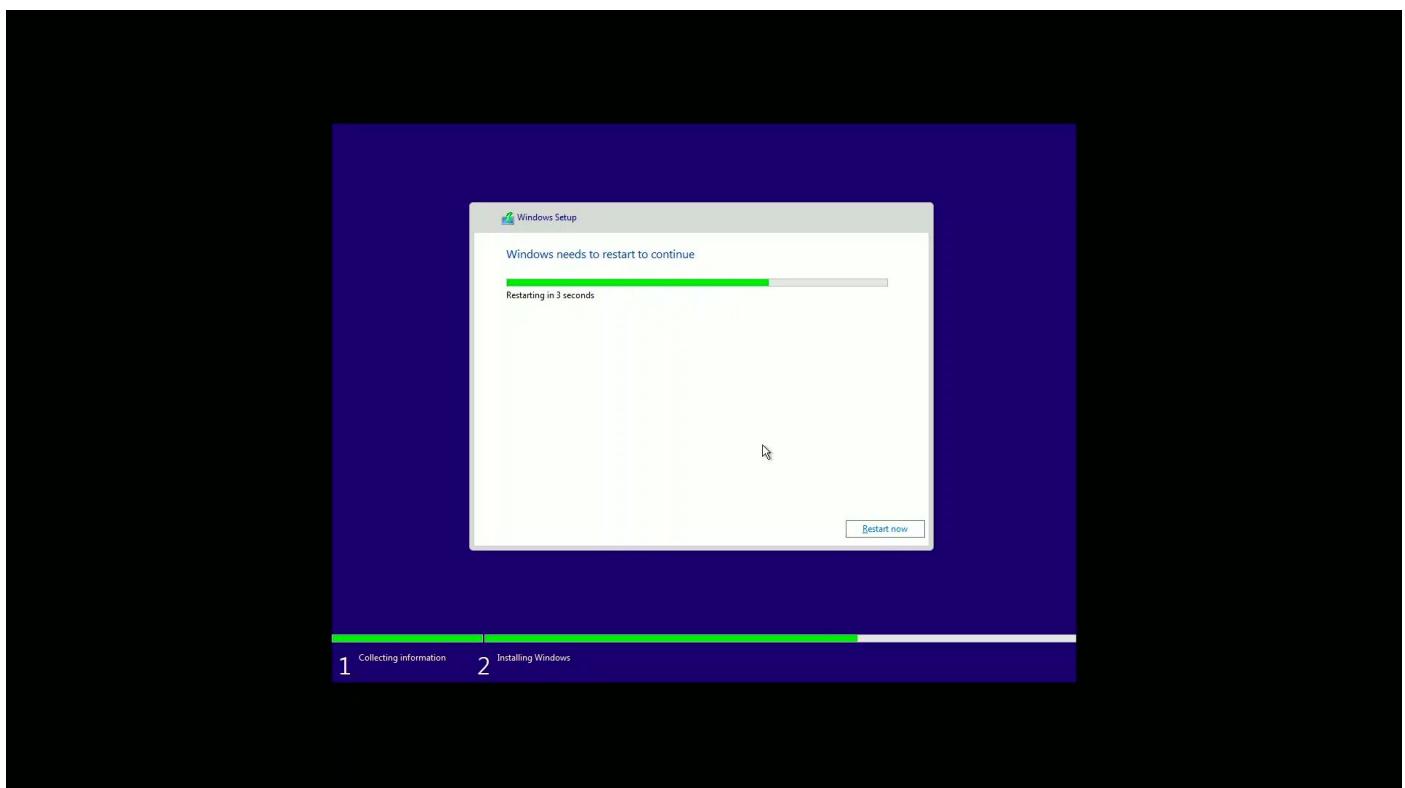
Step 7: Select the disk and click “Next”.



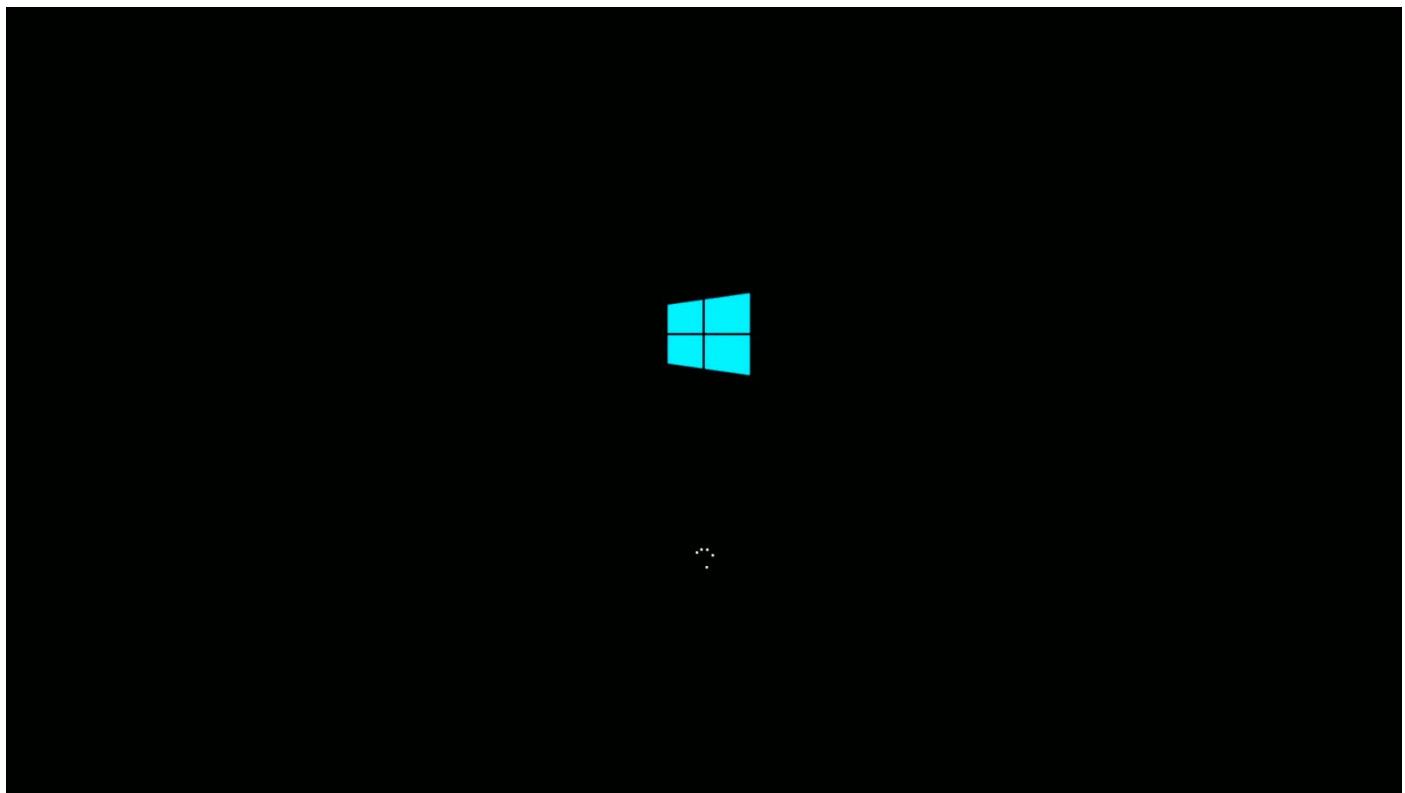
Step 8: Wait for the installation to get completed.



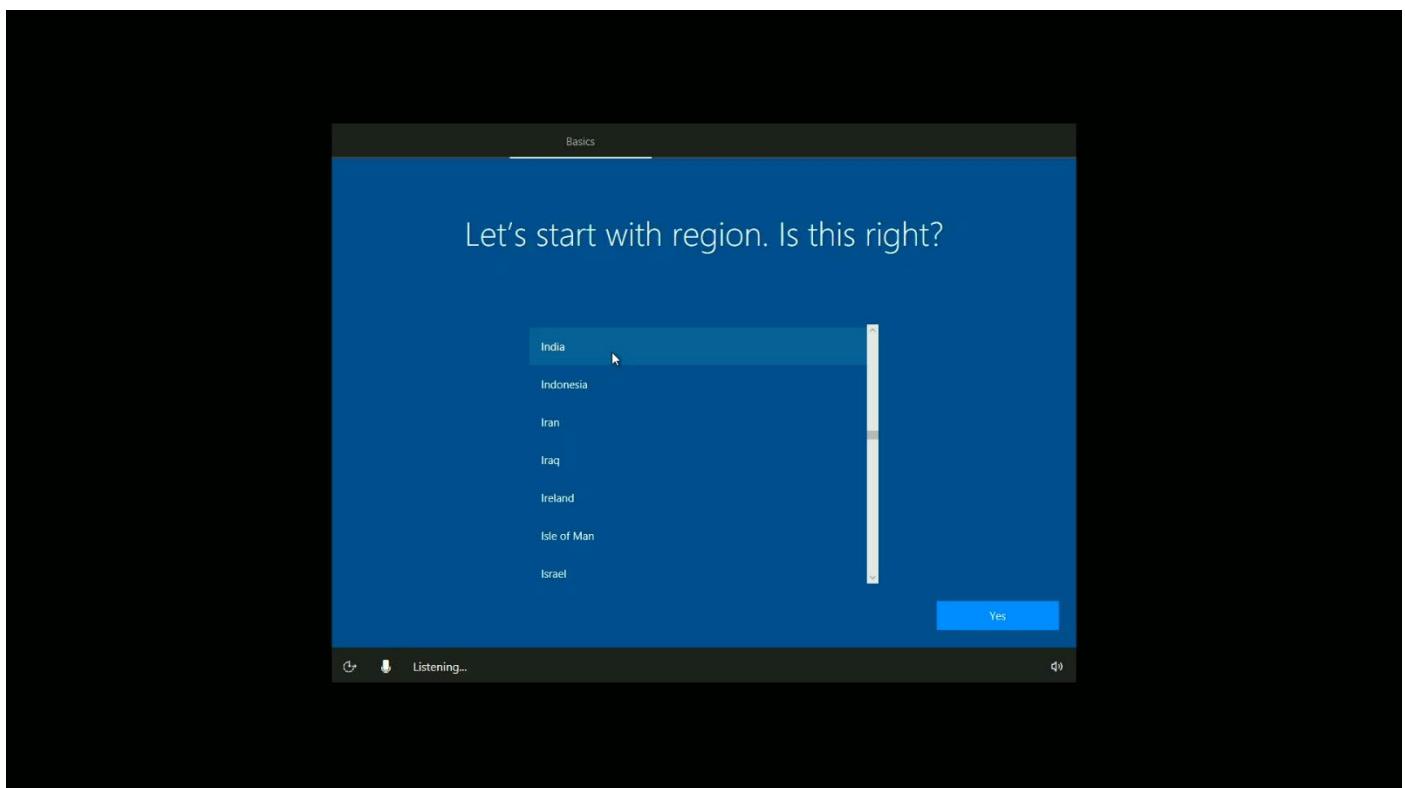
Step 9: Click “Restart Now”.



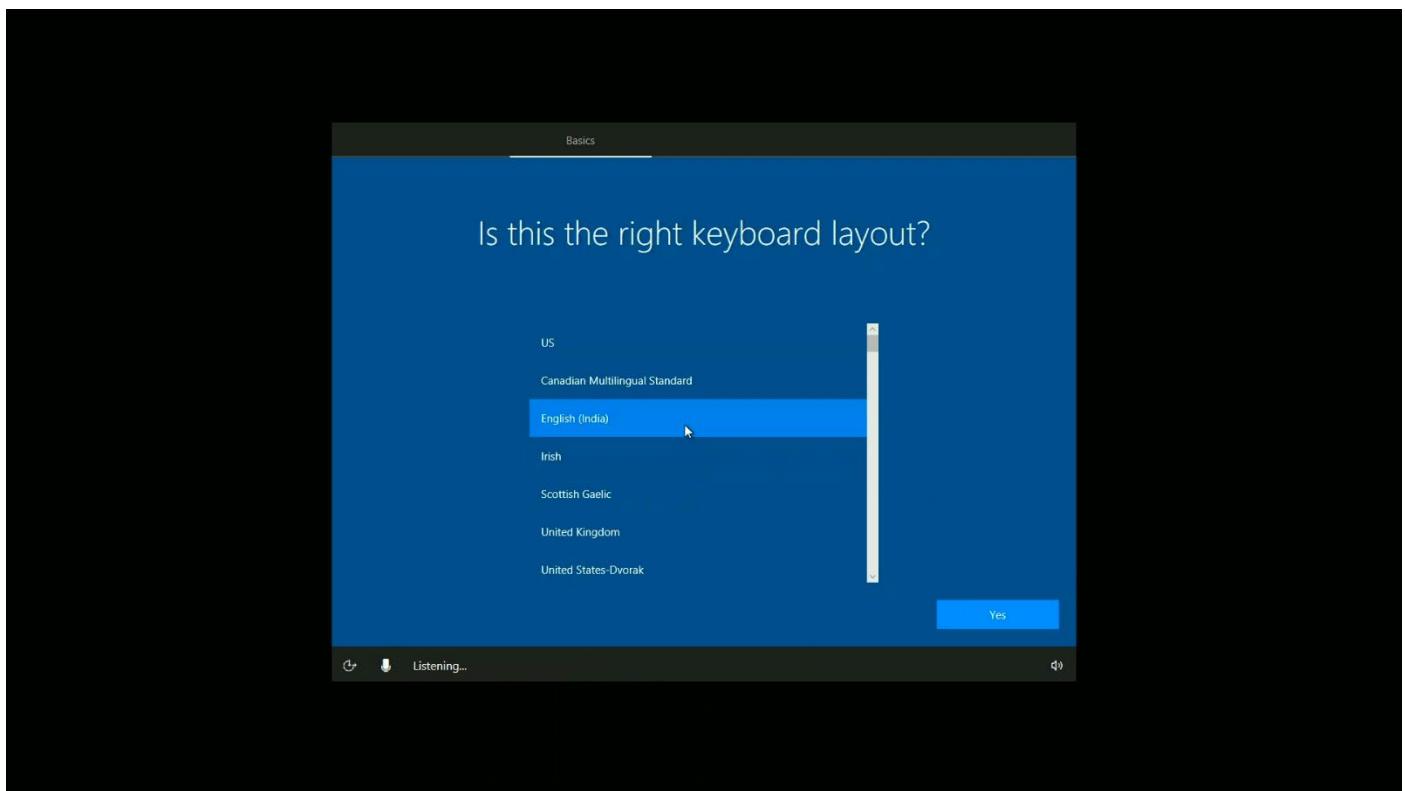
Step 10: System Reboots.



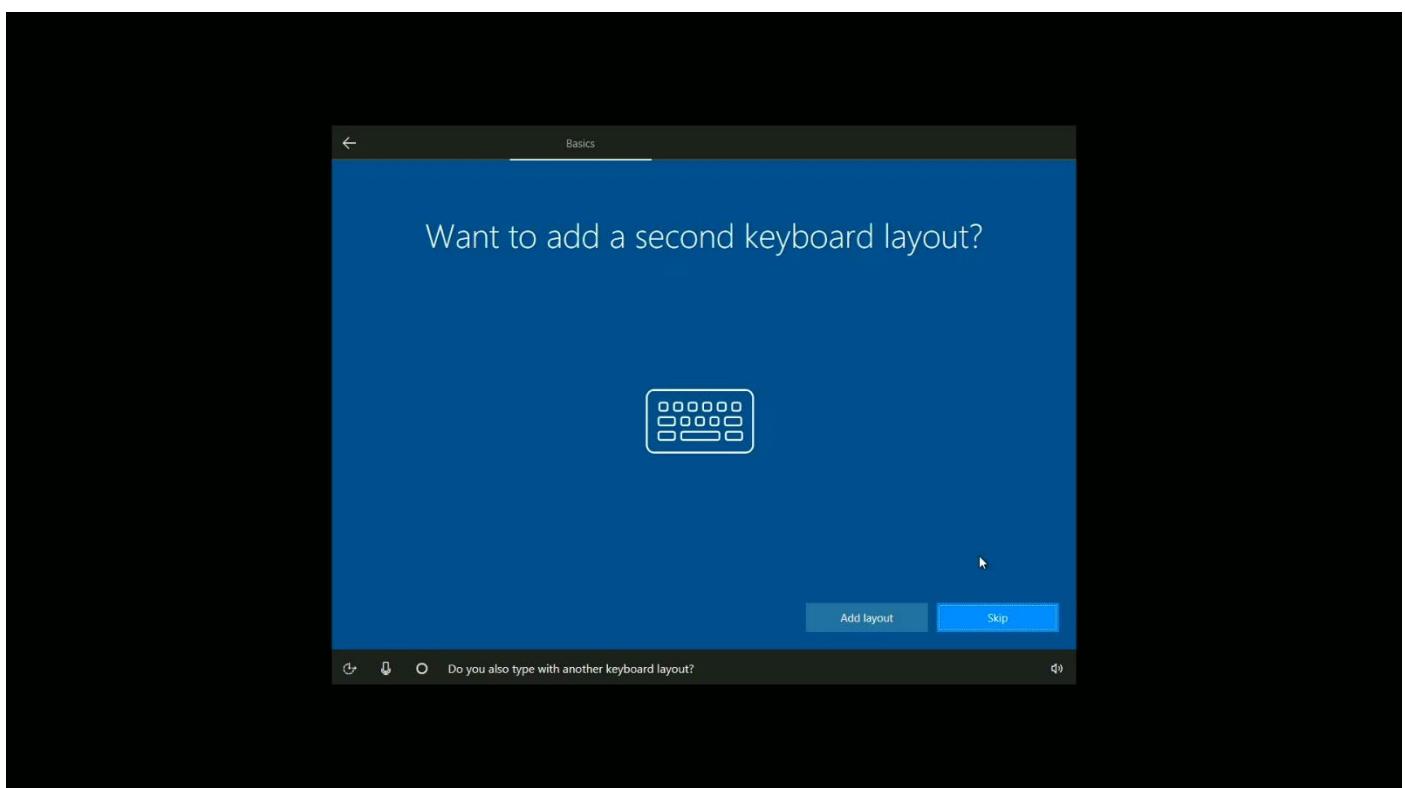
Step 11: Select your region and click “Yes”.



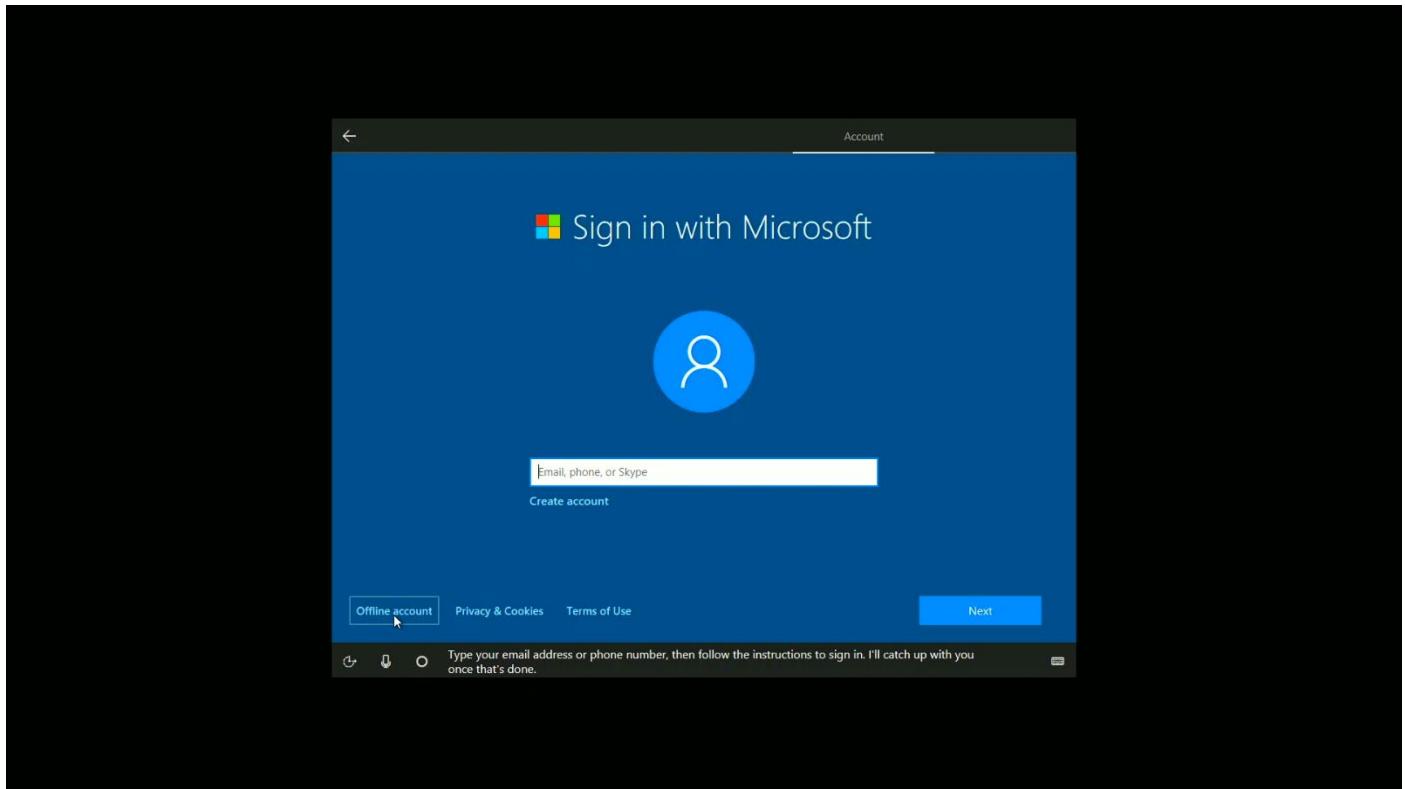
Step 12: Select Keyboard Layout and click “Yes”.



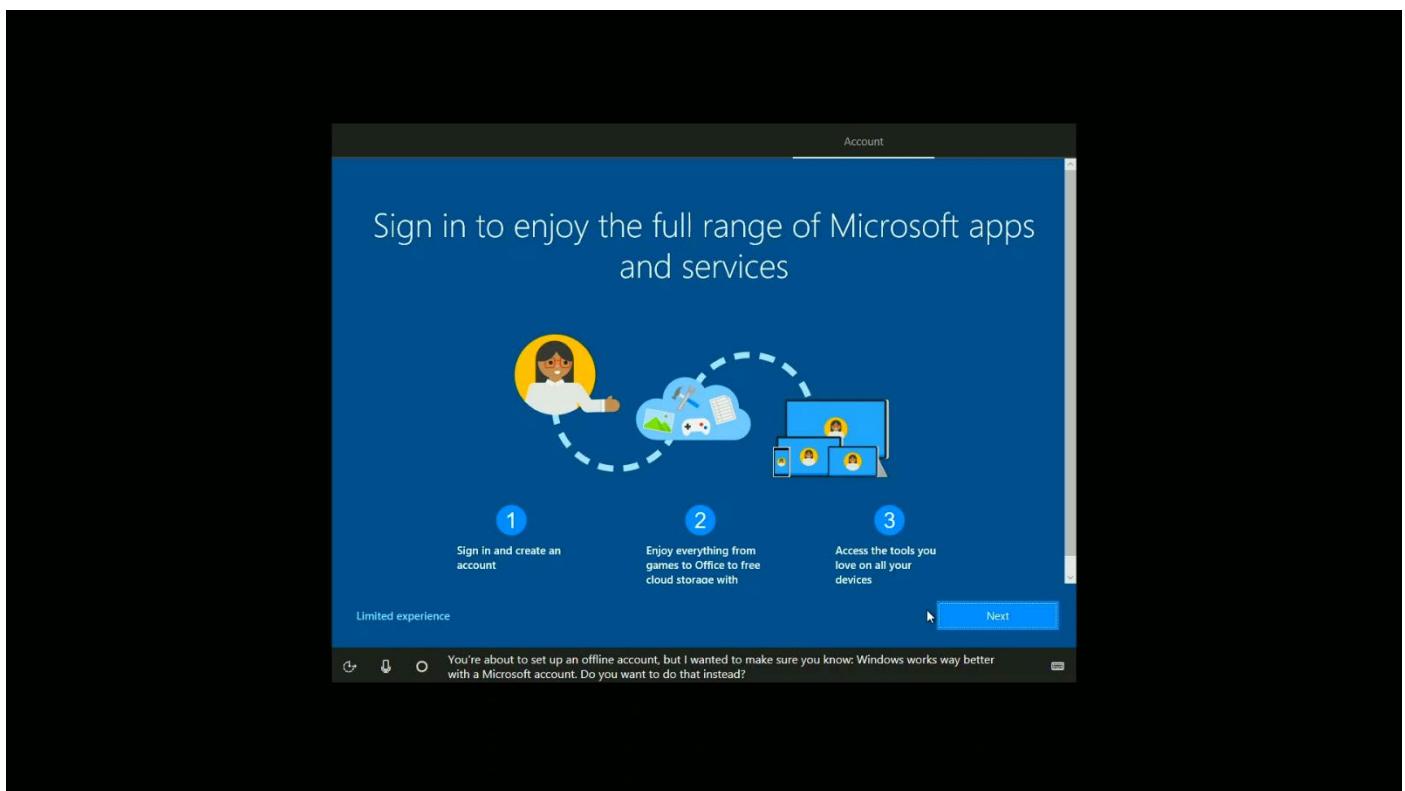
Step 13: Click “Yes”.



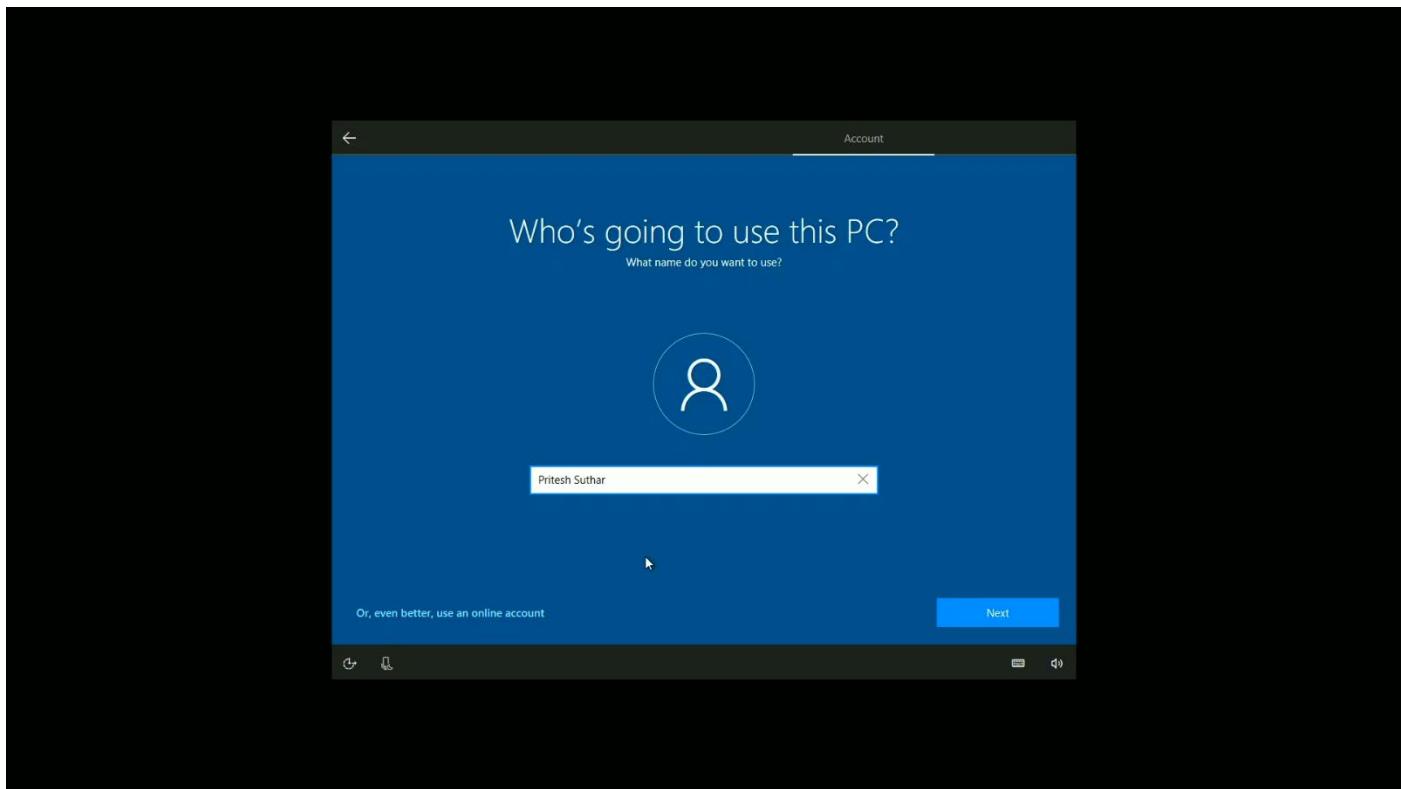
Step 14: Click “Offline Account”.



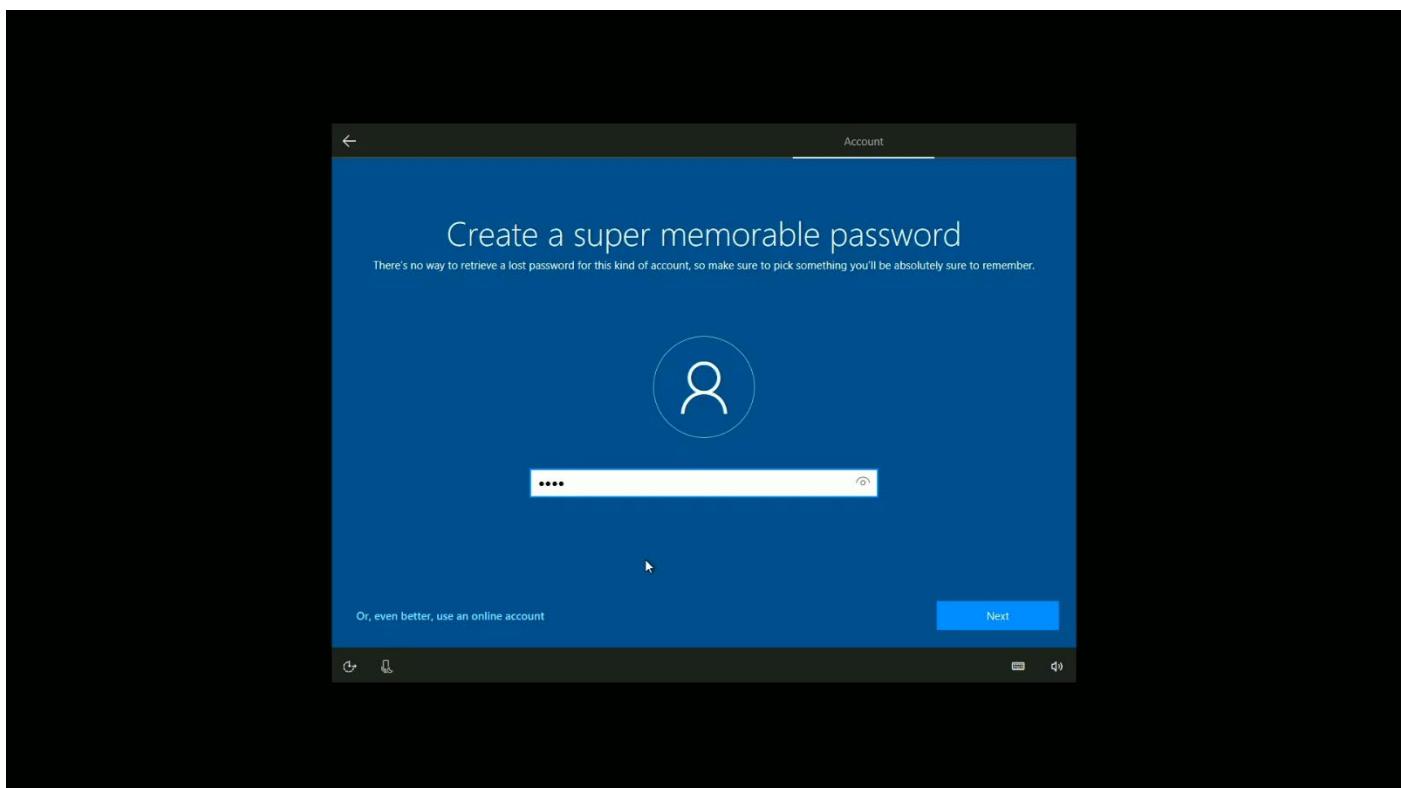
Step 15: Click “Limited Experience”.



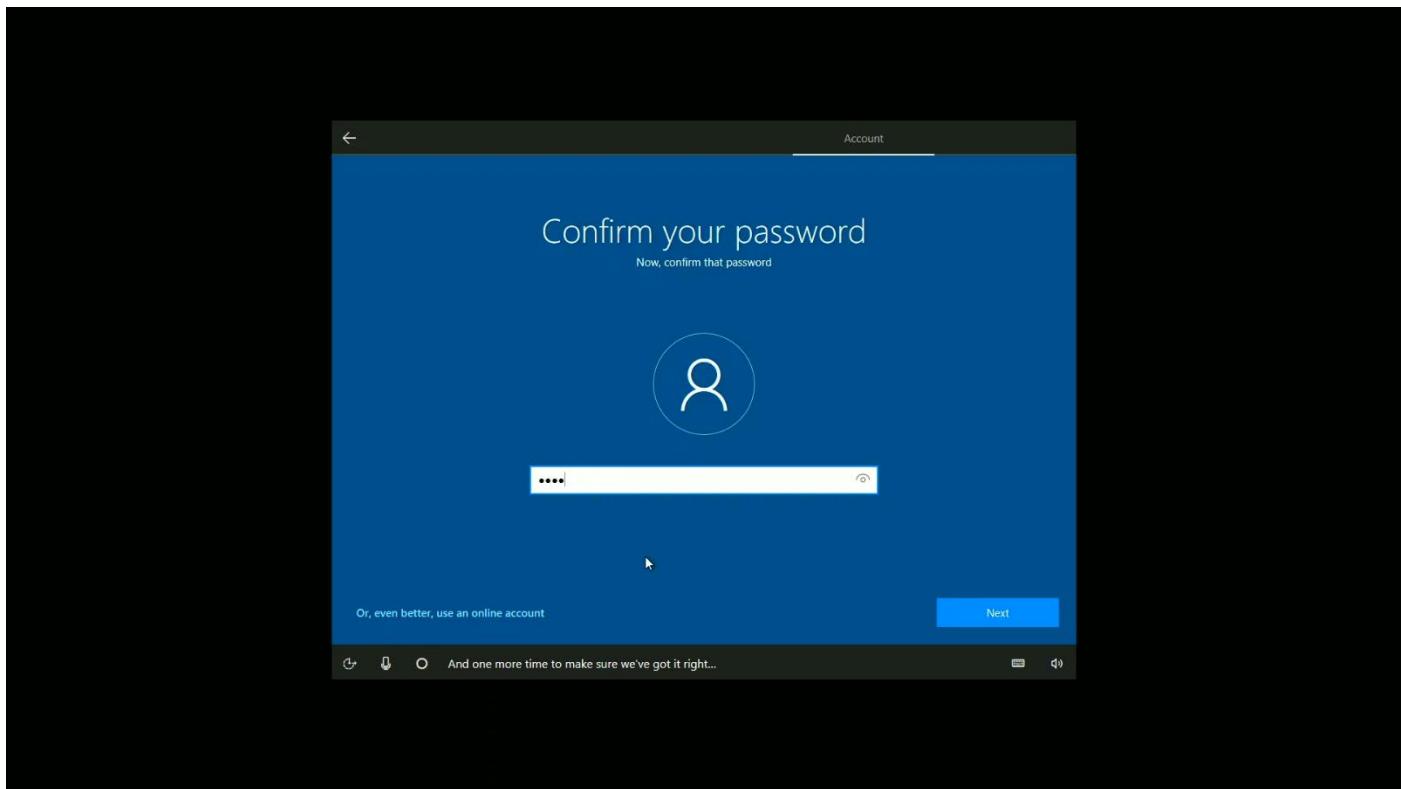
Step 16: Enter username and click “Next”.



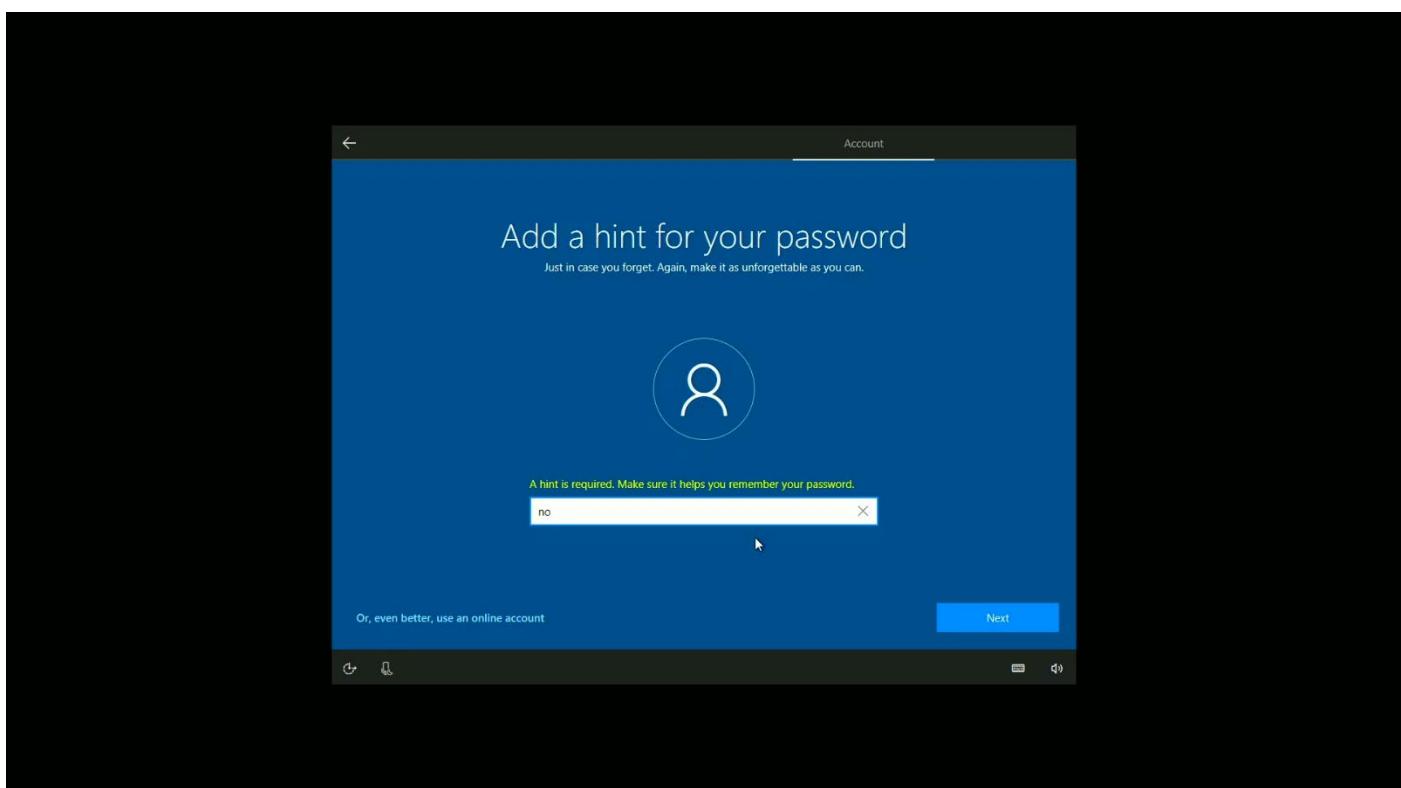
Step 17: Enter the password and click “Next”.



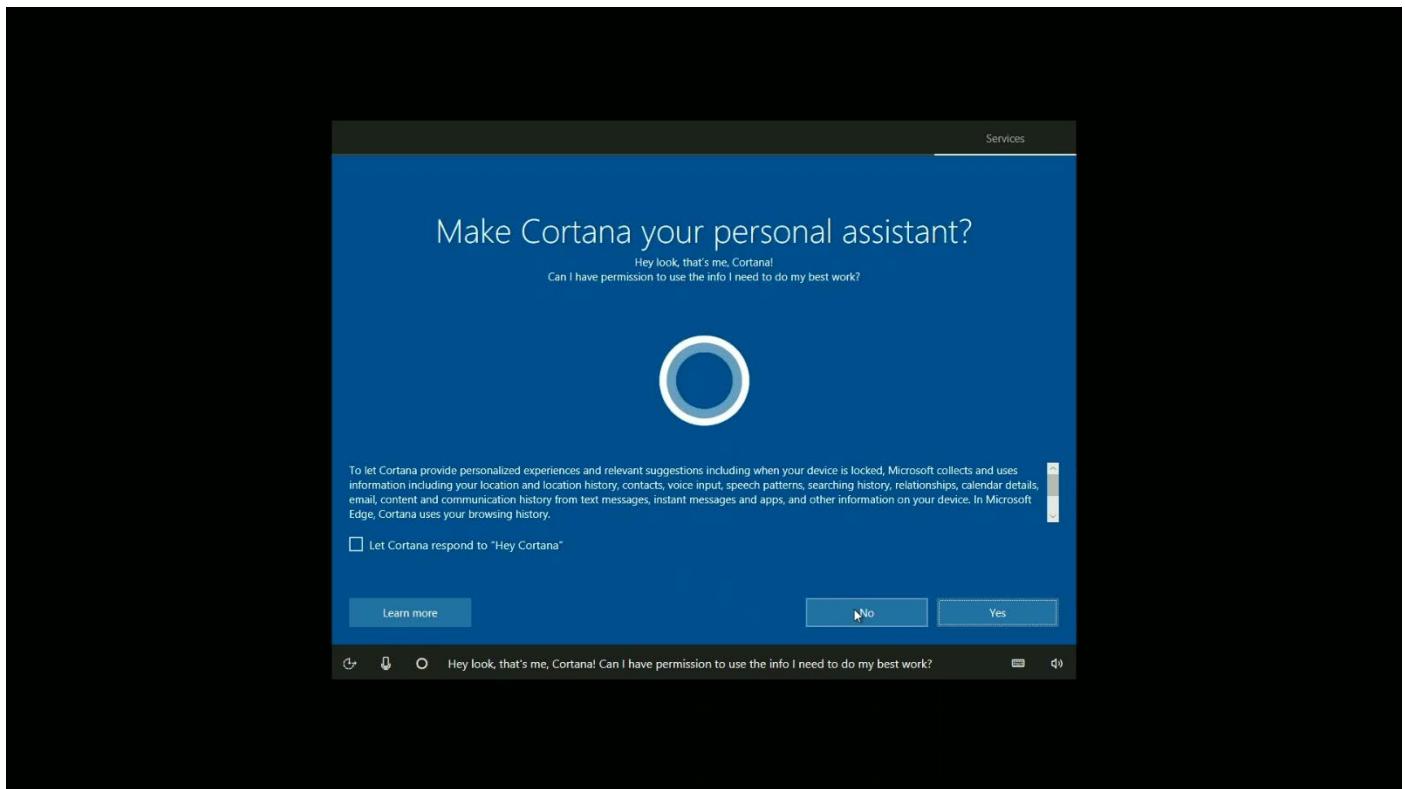
Step 18: Re-enter the password and click “Next”.



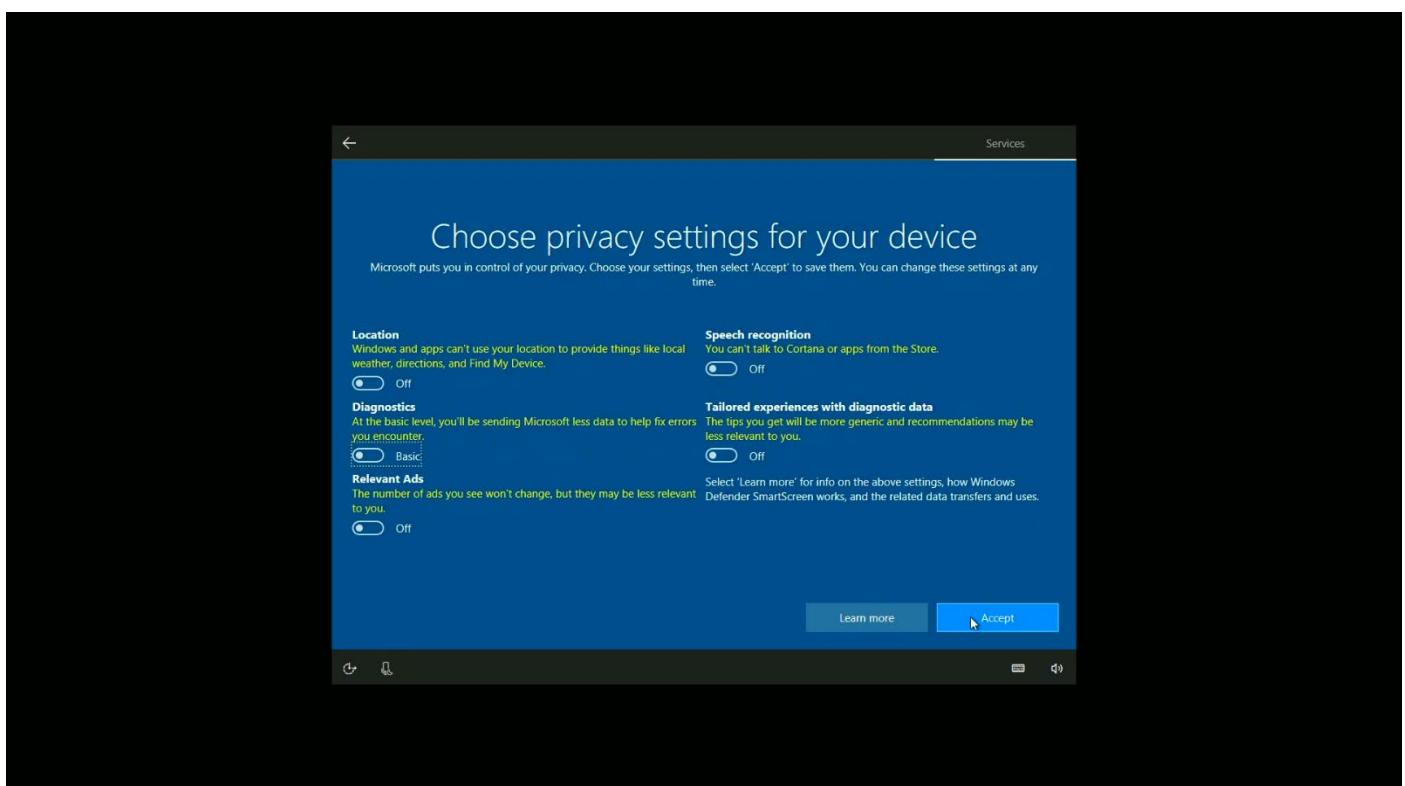
Step 19: Enter a hint for your password click “Next”.



Step 20: Click “No”.



Step 21: Uncheck all the checkboxes and click “Next”.





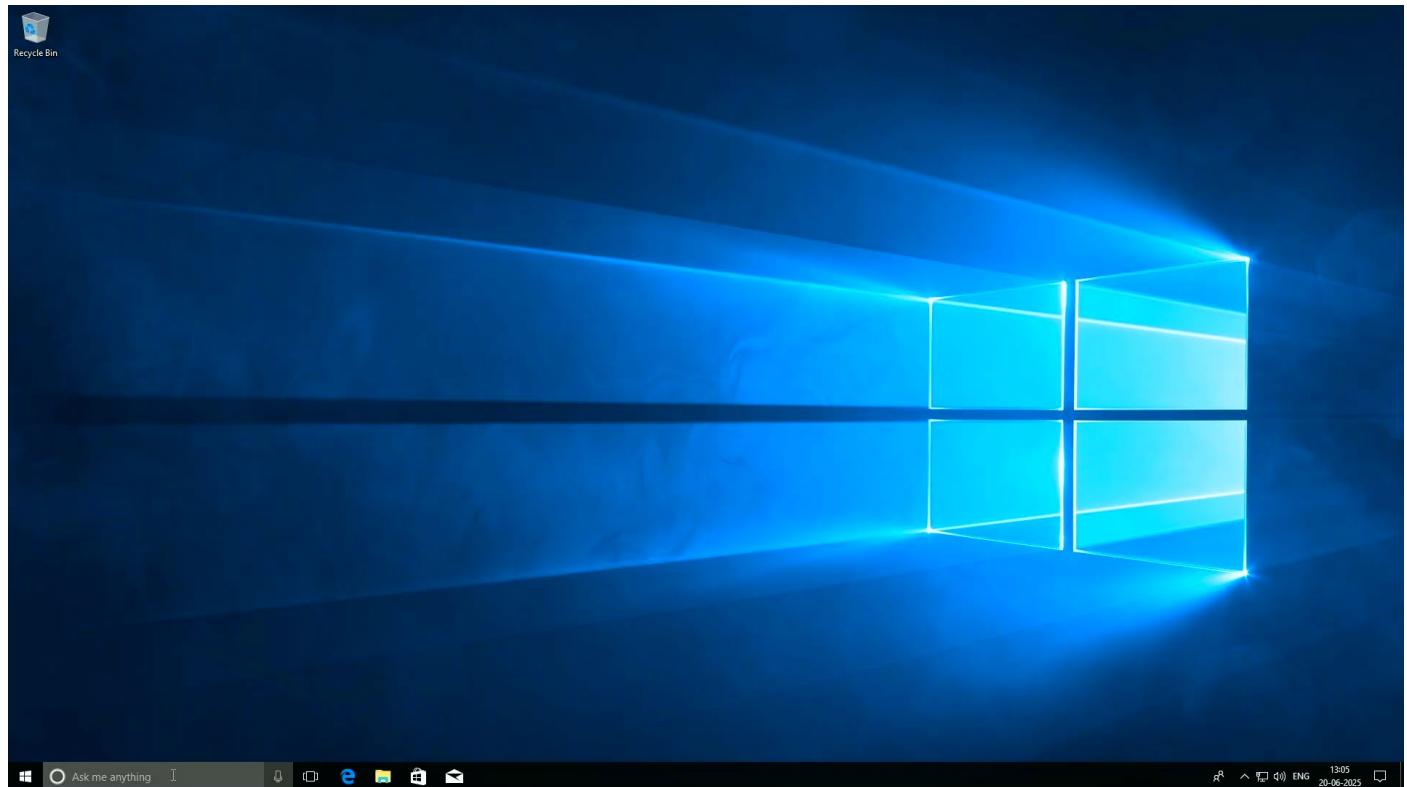
**DRS. KIRAN & PALLAVI PATEL GLOBAL UNIVERSITY**

Established Under Gujarat Private Universities (Amendment) Act, 2021 (Gujarat Act No. 15 of 2021)

KRISHNA SCHOOL OF EMERGING TECHNOLOGY & APPLIED RESEARCH (KSET)

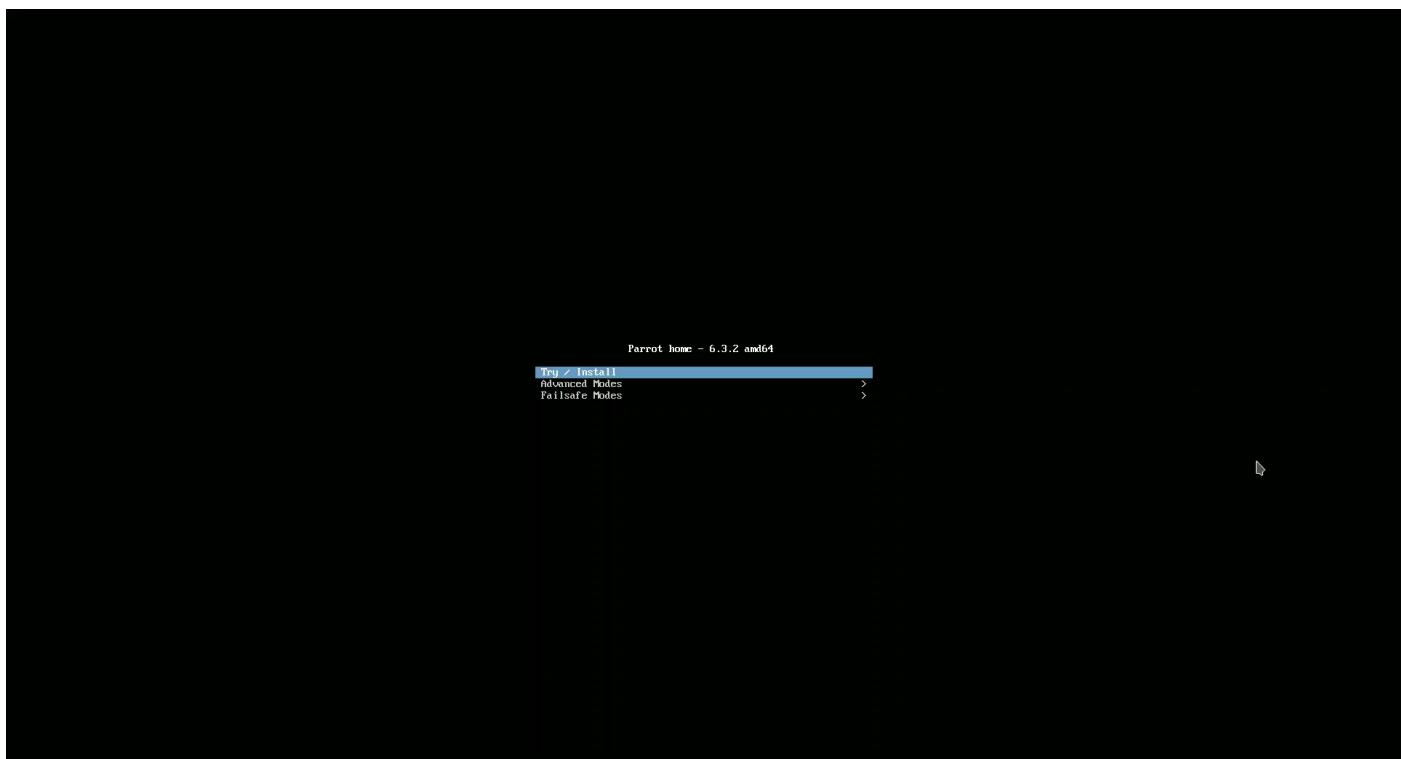
**KPGU**  
Vadodara

Step 22: Windows is successfully installed.

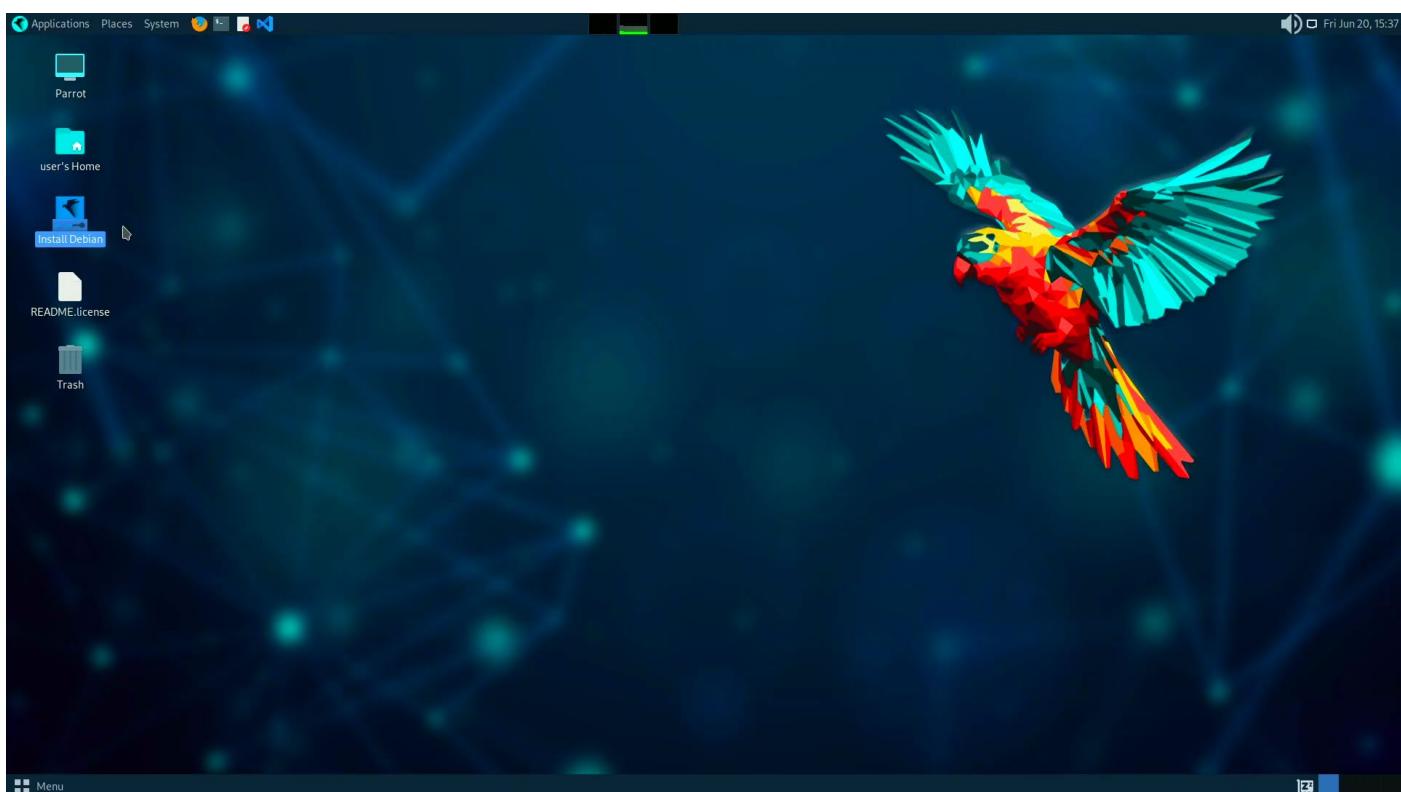


Parrot OS:

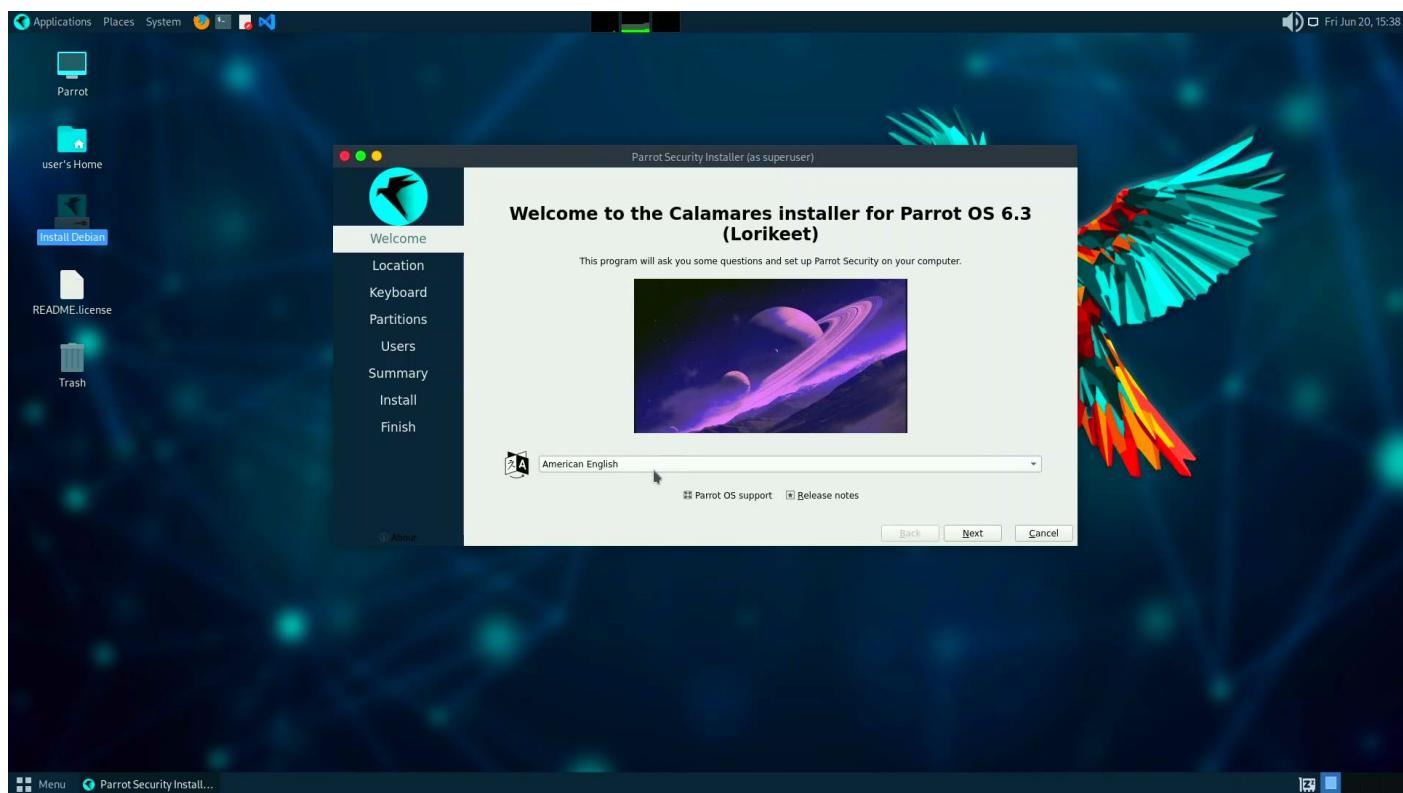
Step 1: Select “Try/Install” and press Enter.



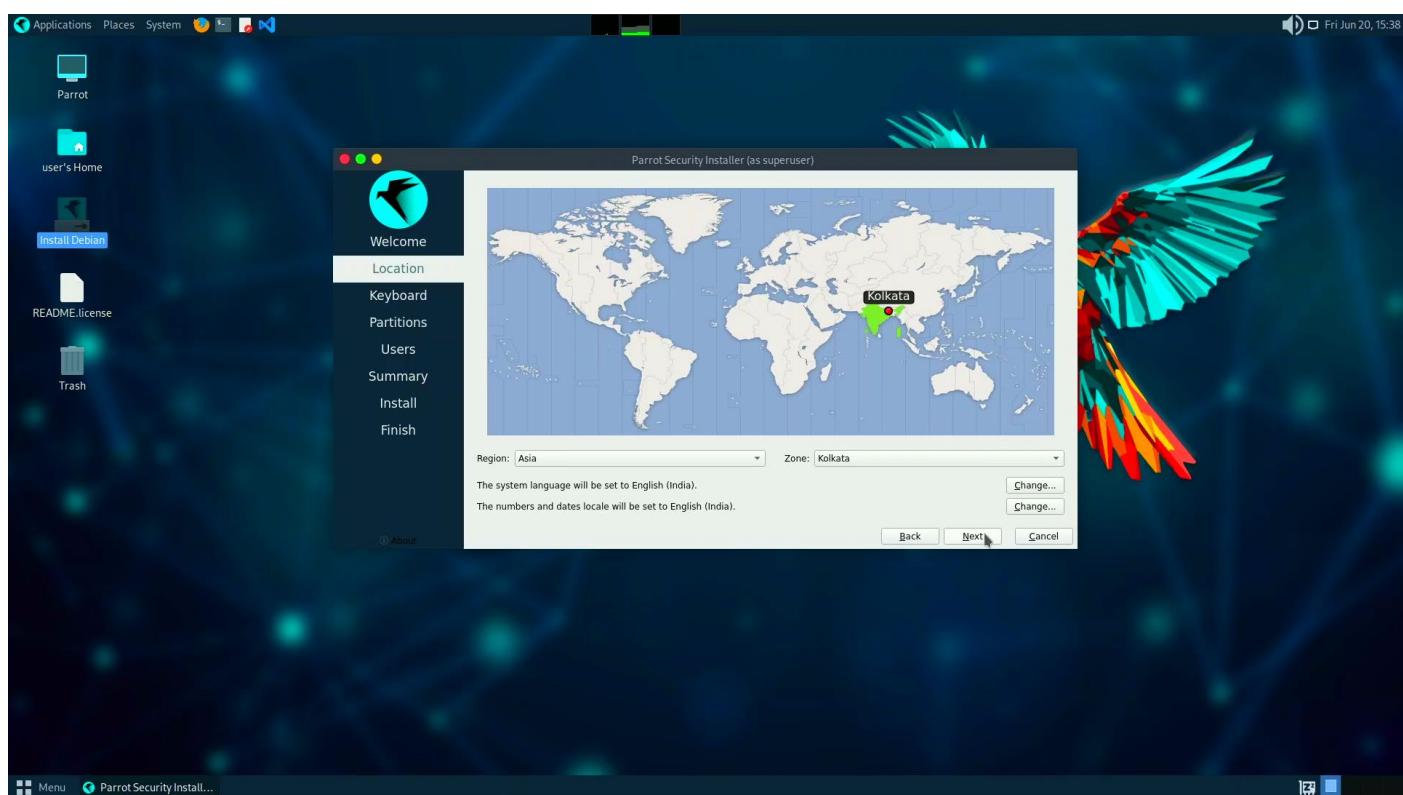
Step 2: Open “Install Debian”.



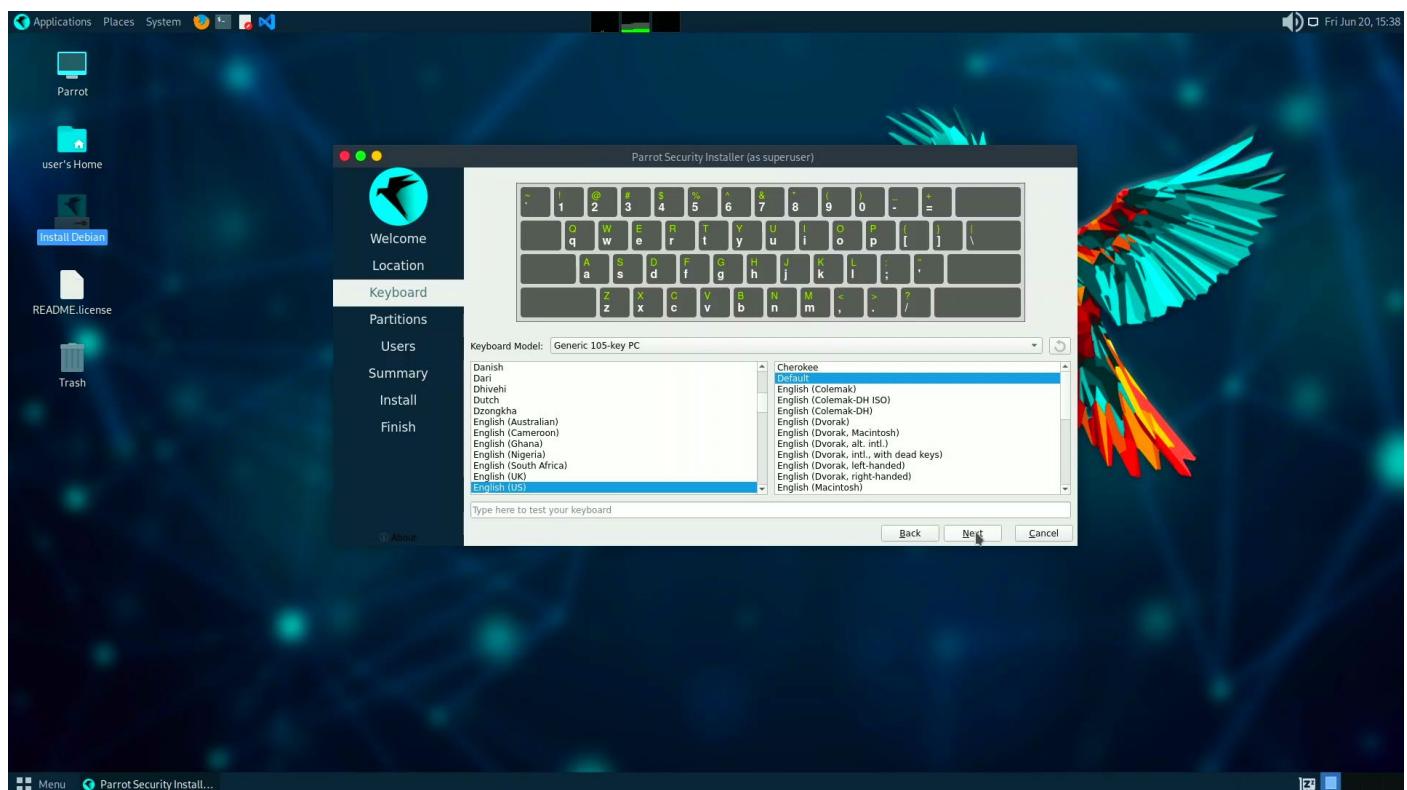
Step 3: Select system language and click “Next”.



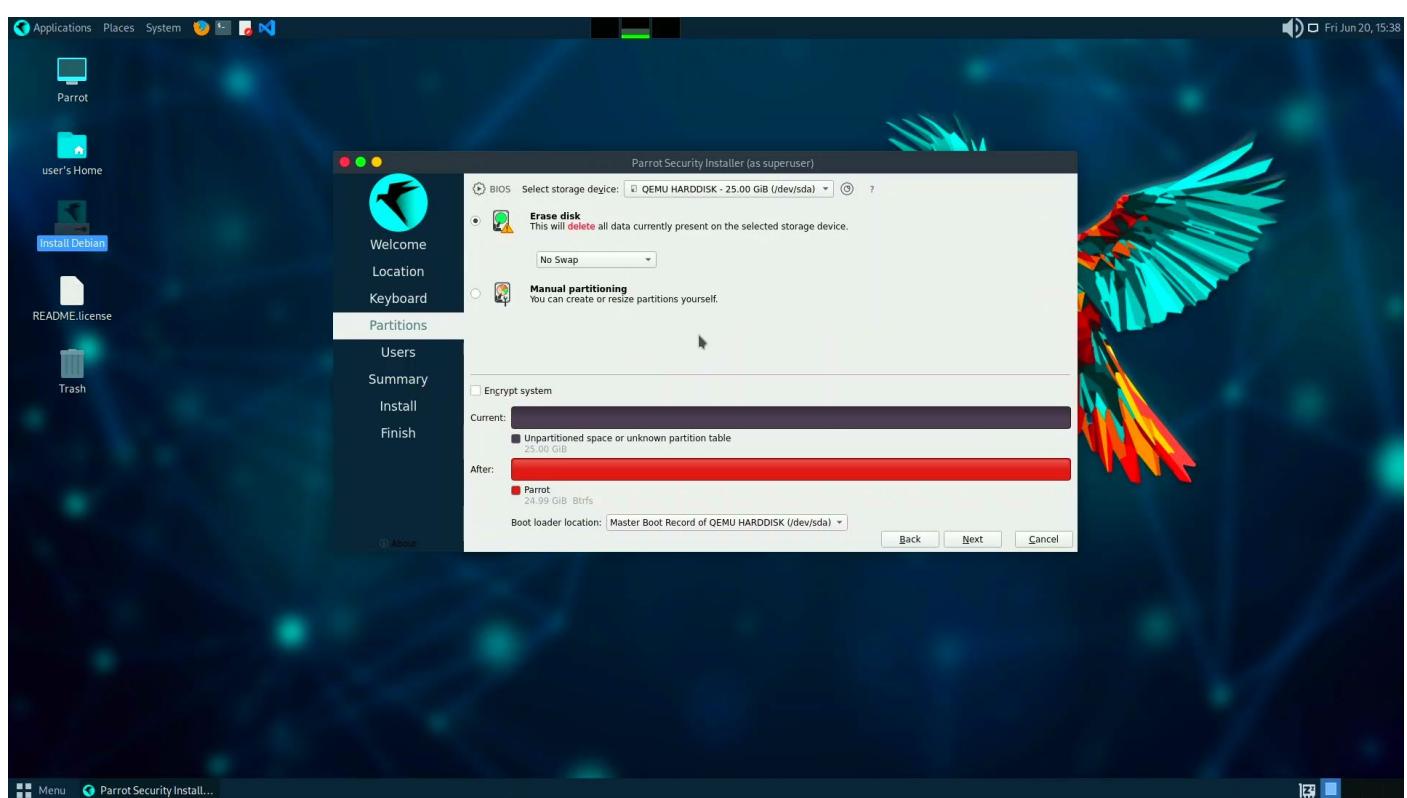
Step 4: Select region, time zone and click “Next”.



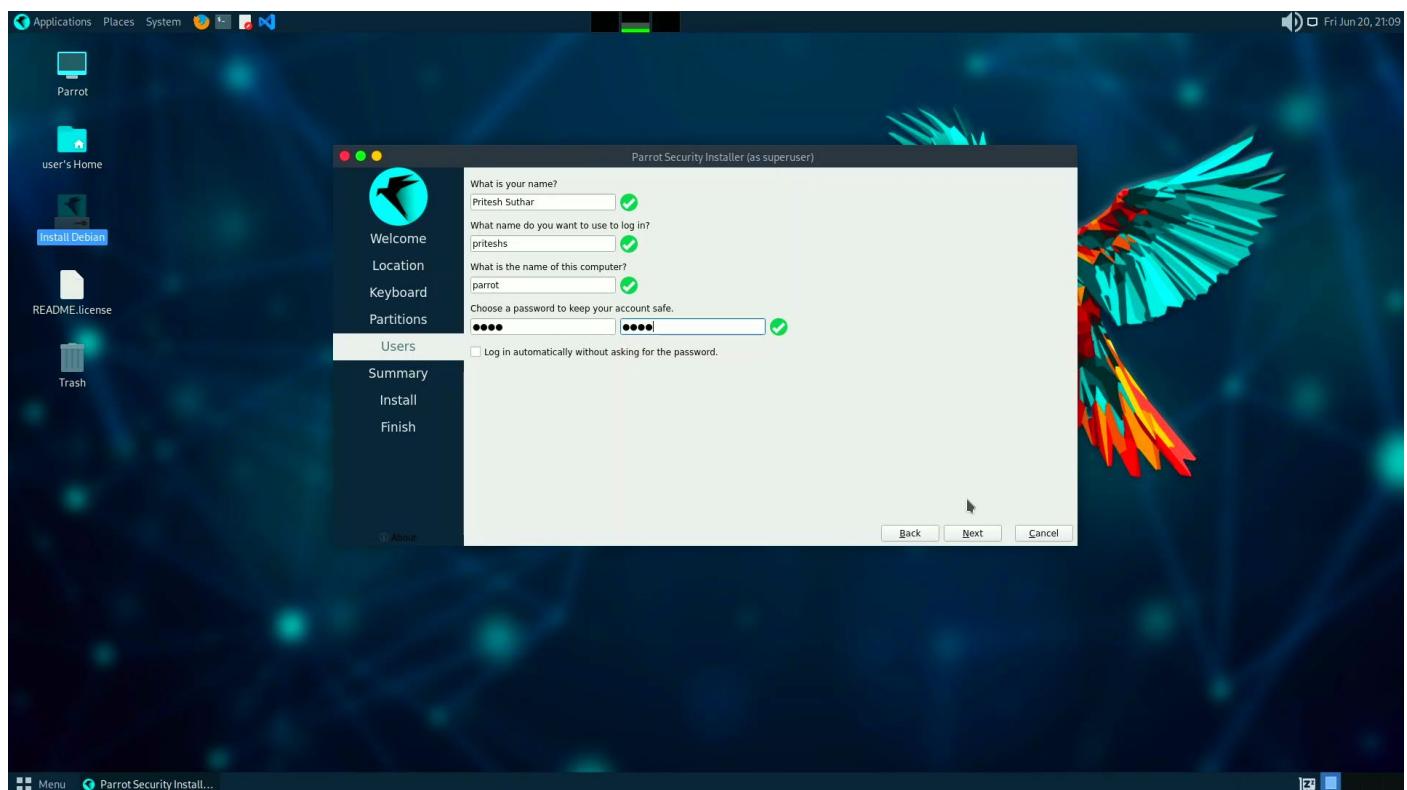
Step 5: Select keyboard language, layout and click “Next”.



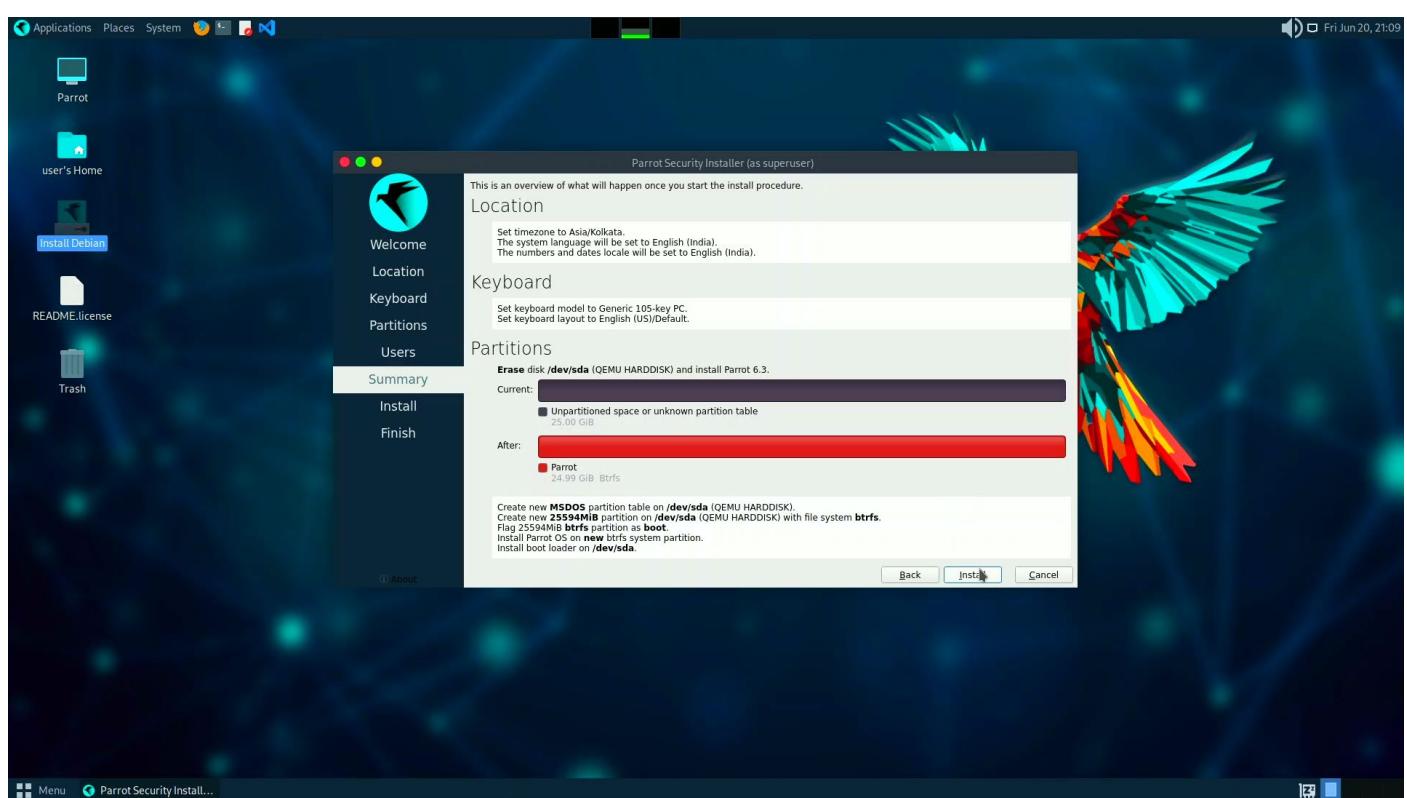
Step 6: Select “Erase Disk” click “Next”.



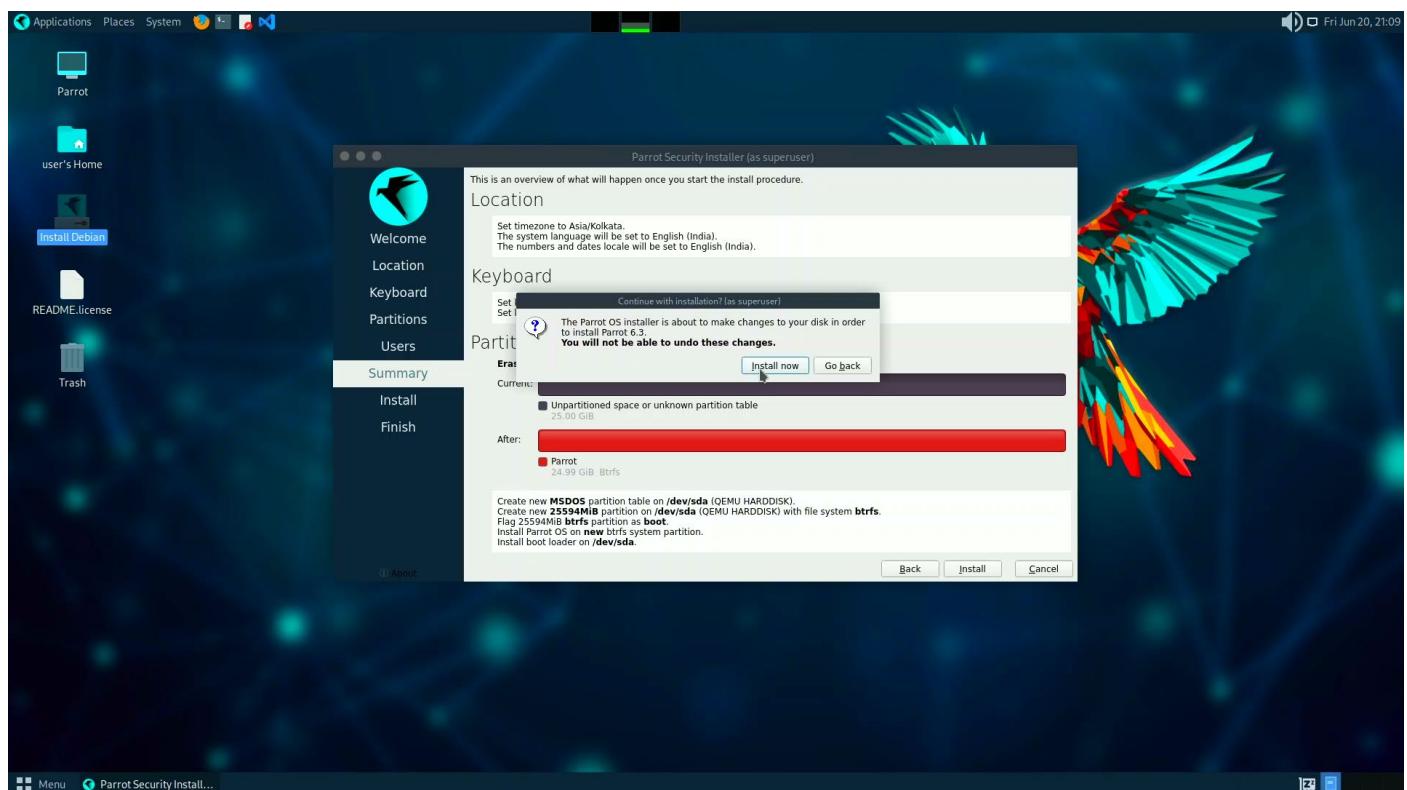
Step 7: Enter your name, username, device name, password click “Next”.



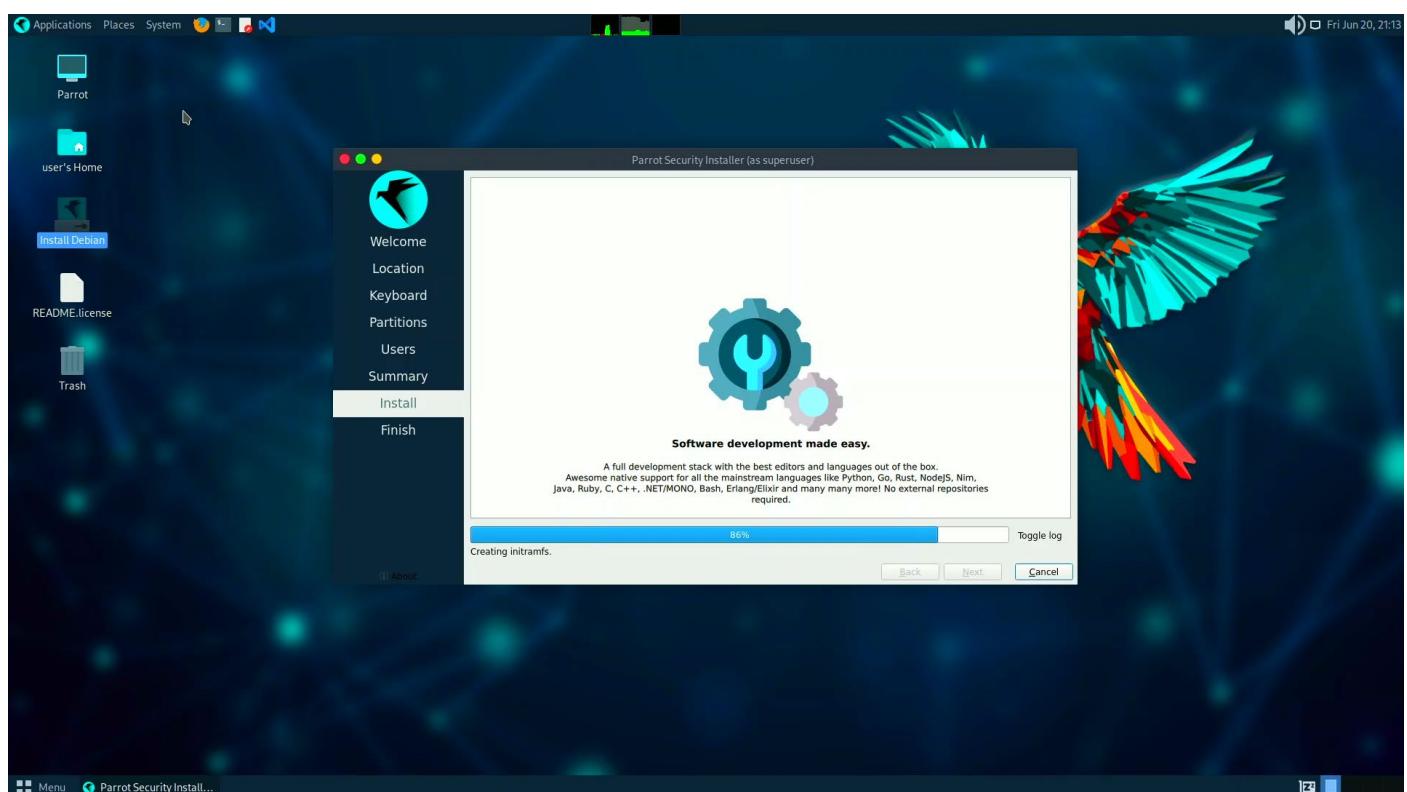
Step 8: Verify the information click “Next”.



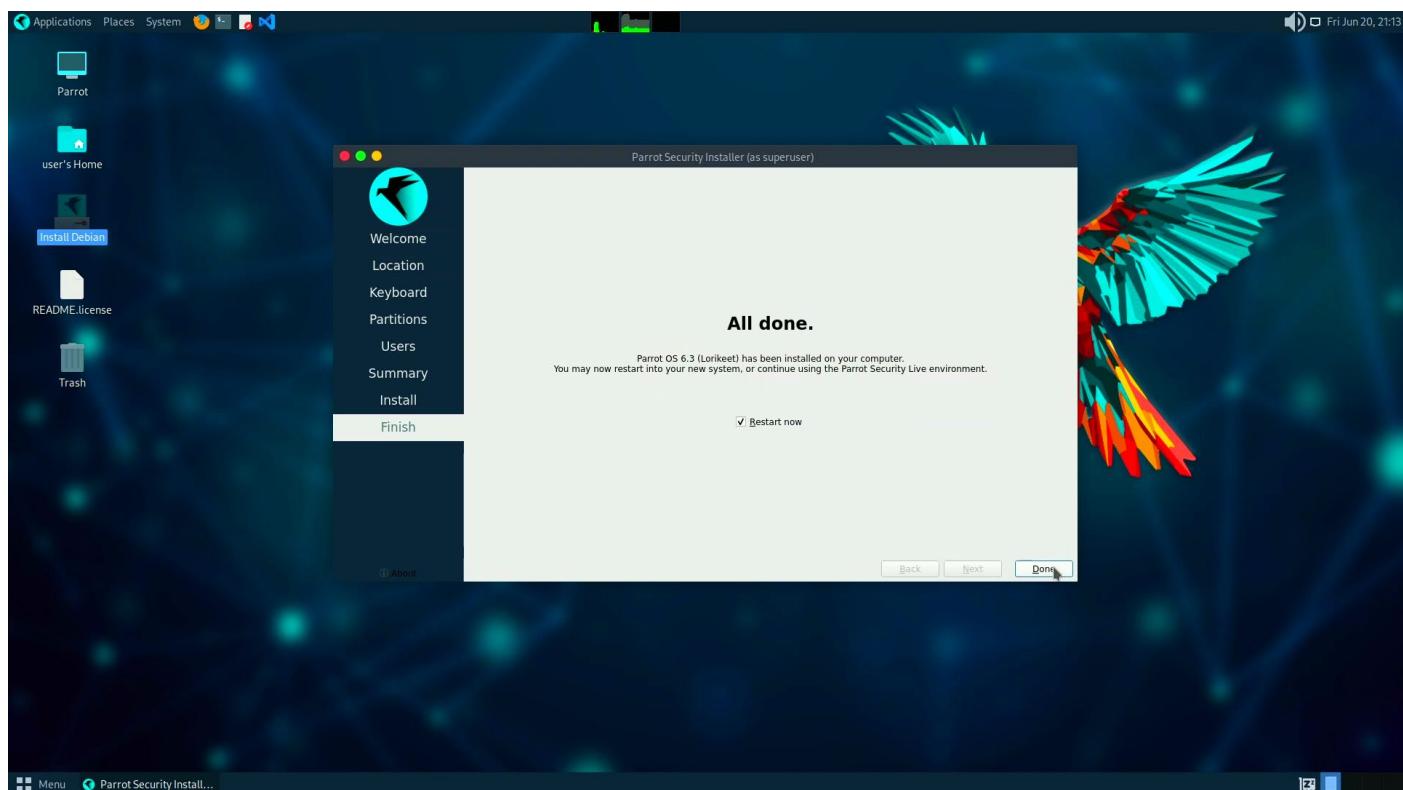
Step 9: Click “Install Now”.



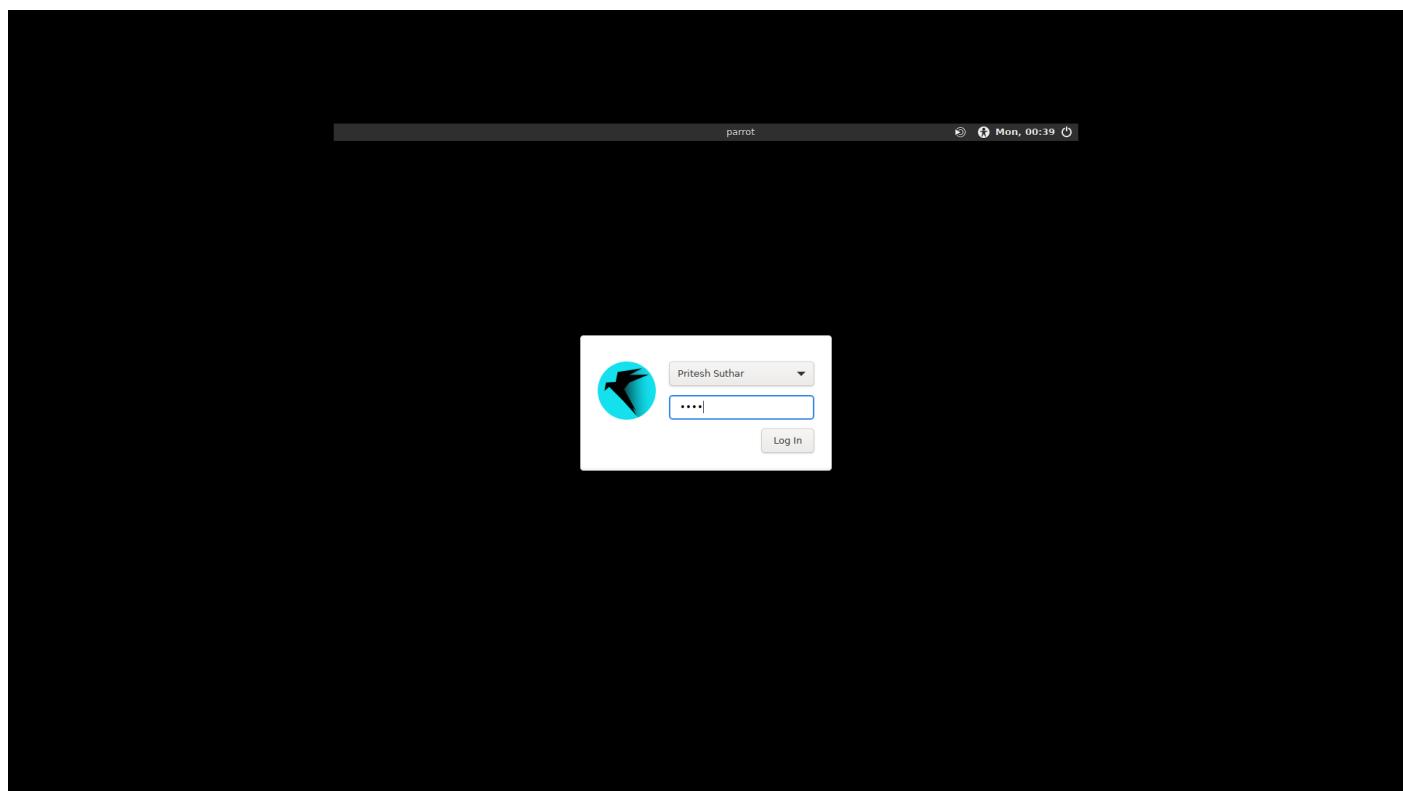
Step 10: Wait for the installation to get completed.



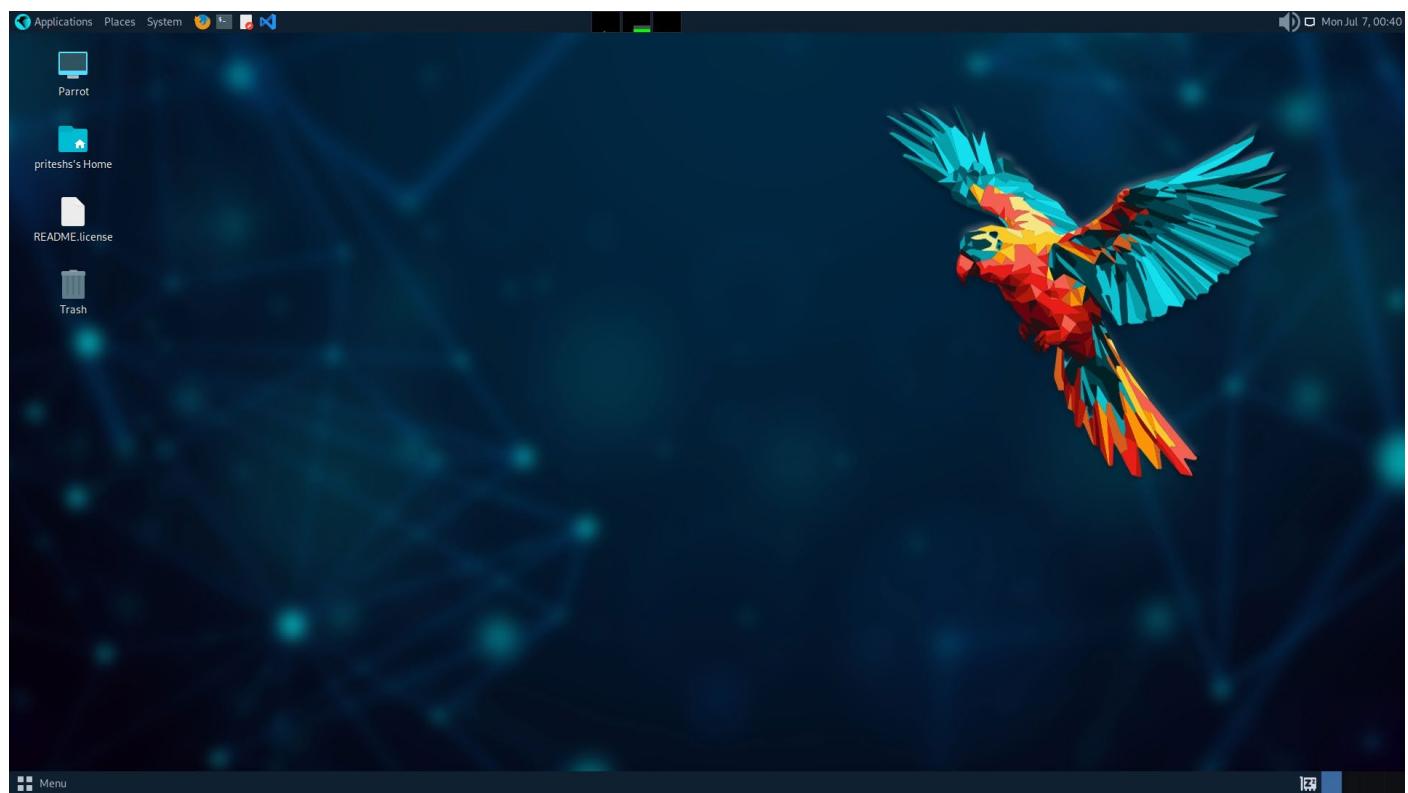
Step 11: Click “Done”



Step 12: Enter your password.

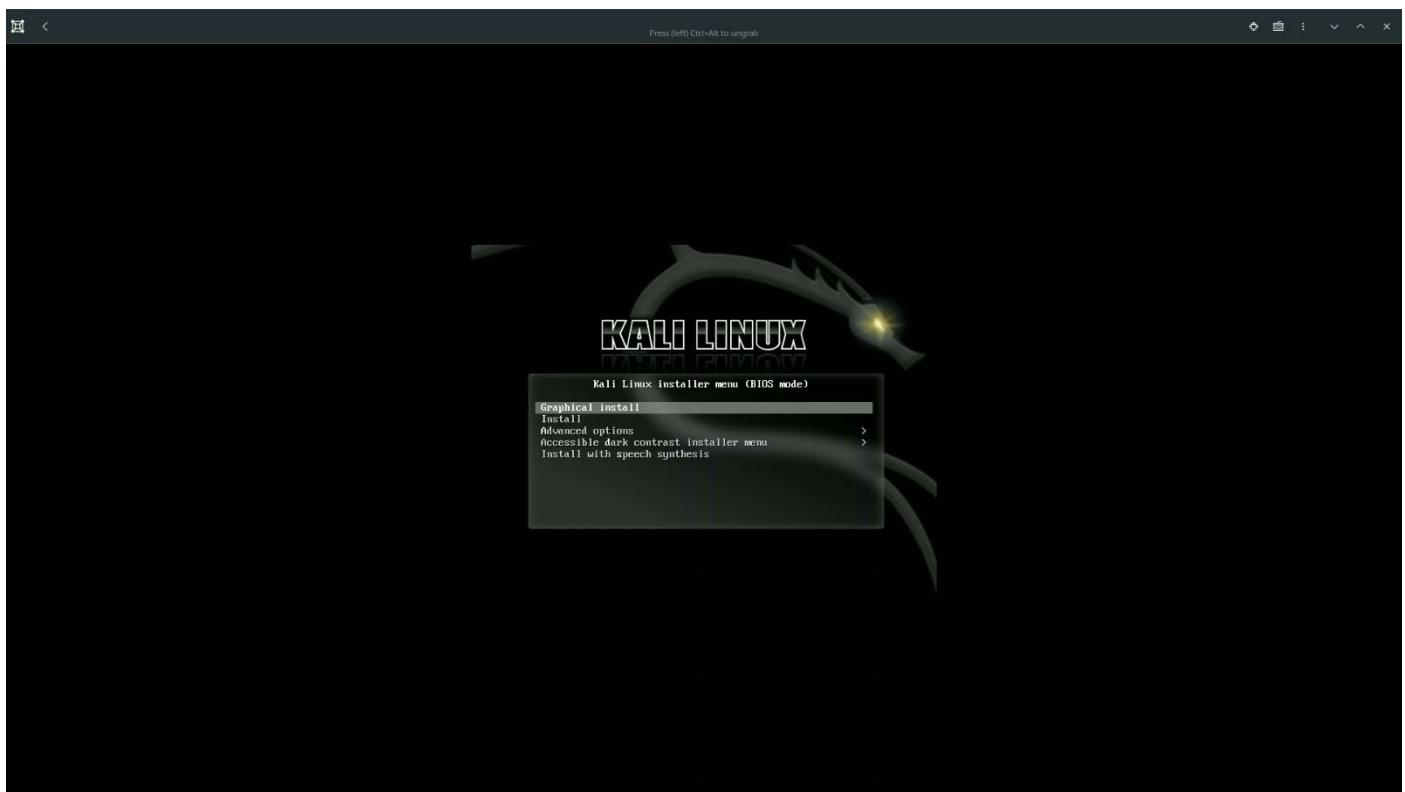


Step 13: Parrot OS is successfully installed.

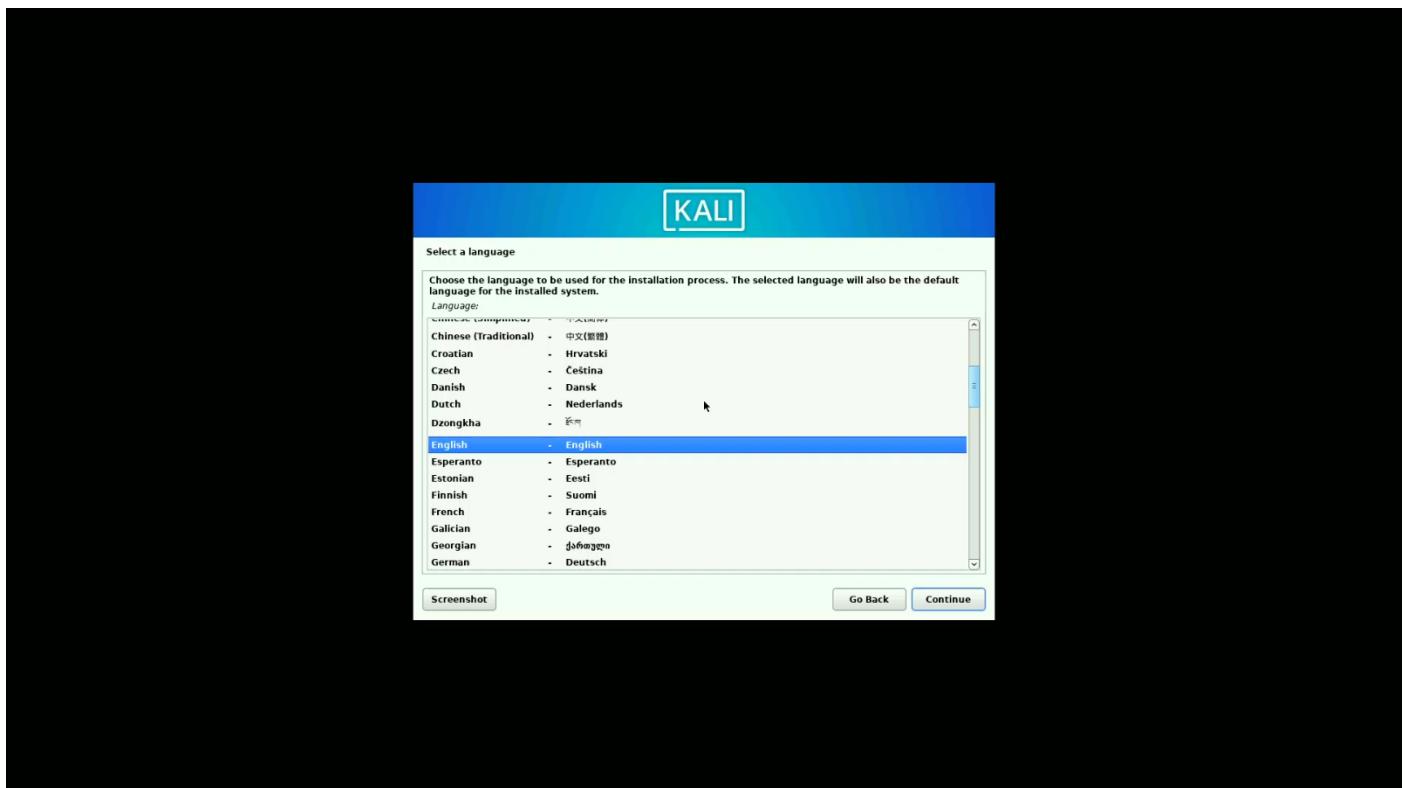


Kali Linux:

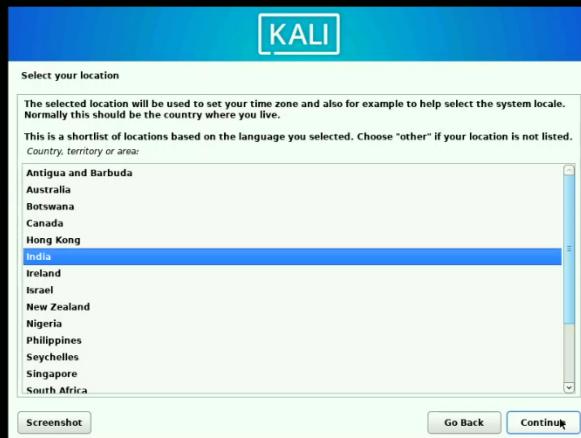
Step 1: Select “Graphical Install” and press Enter.



Step 2: Select language and click “Continue”.



Step 3: Select region and click “Continue”.



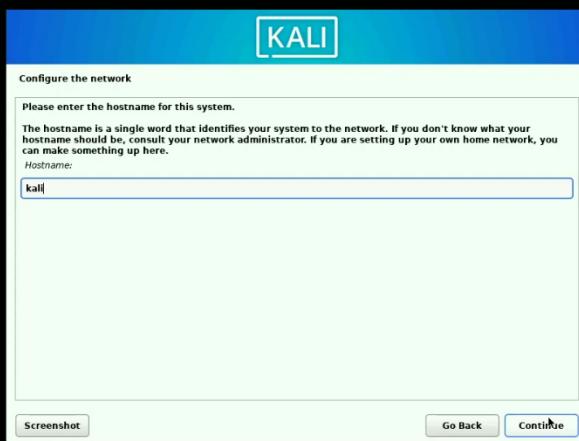
Step 4: Select keyboard layout click “Continue”.



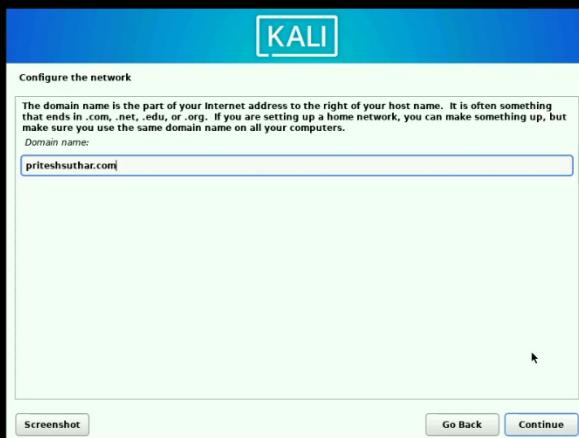
Step 5: Wait for the components to get loaded.



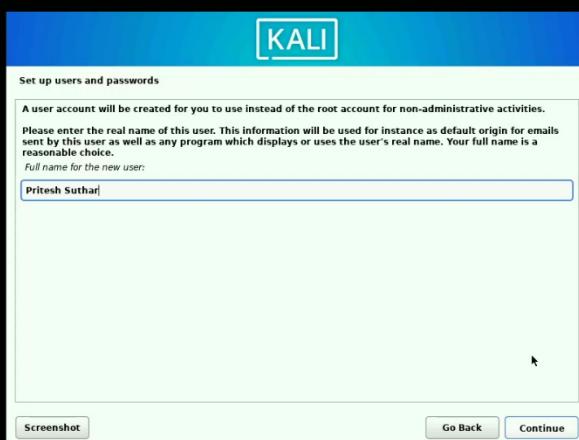
Step 6: Enter device name click “Continue”.



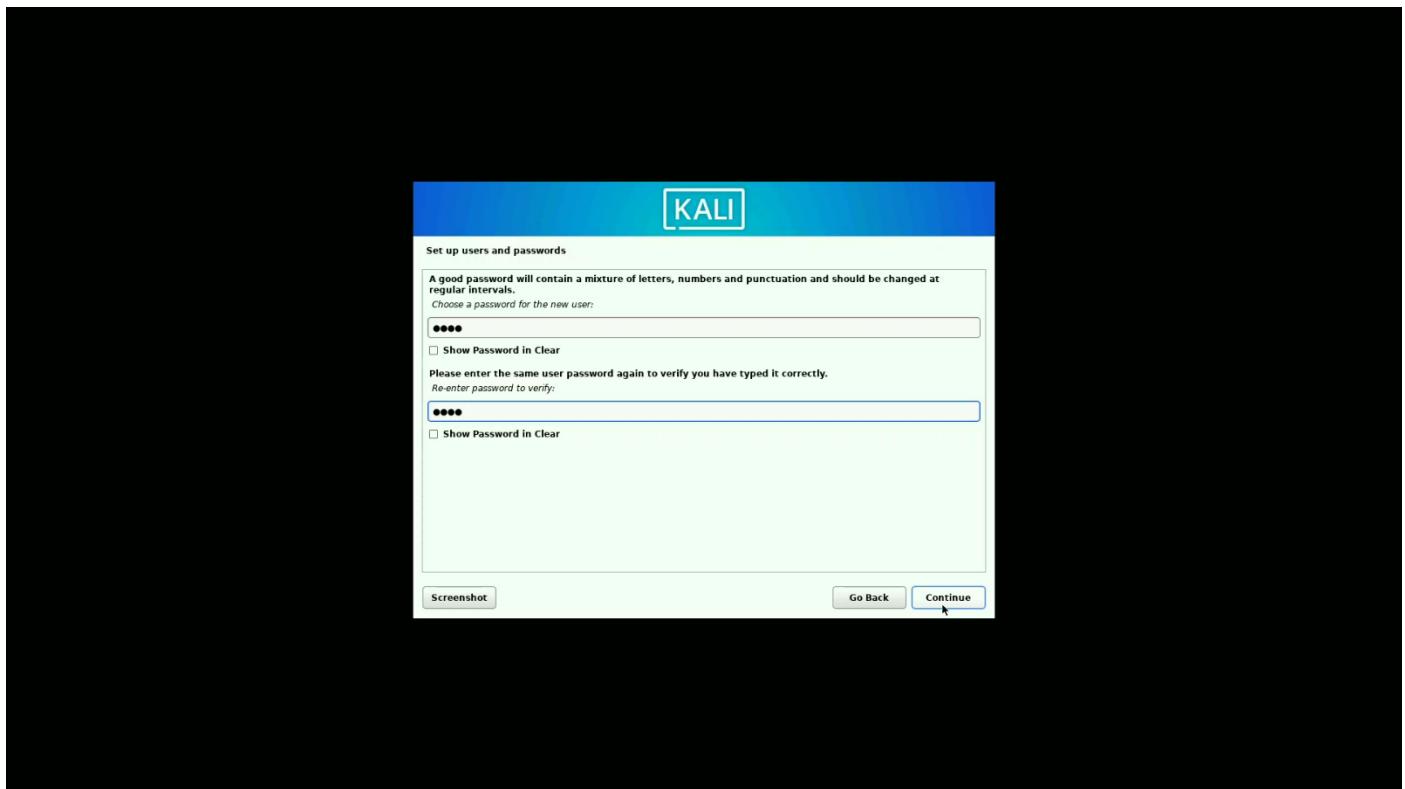
Step 7: Enter domain name click “Continue”.



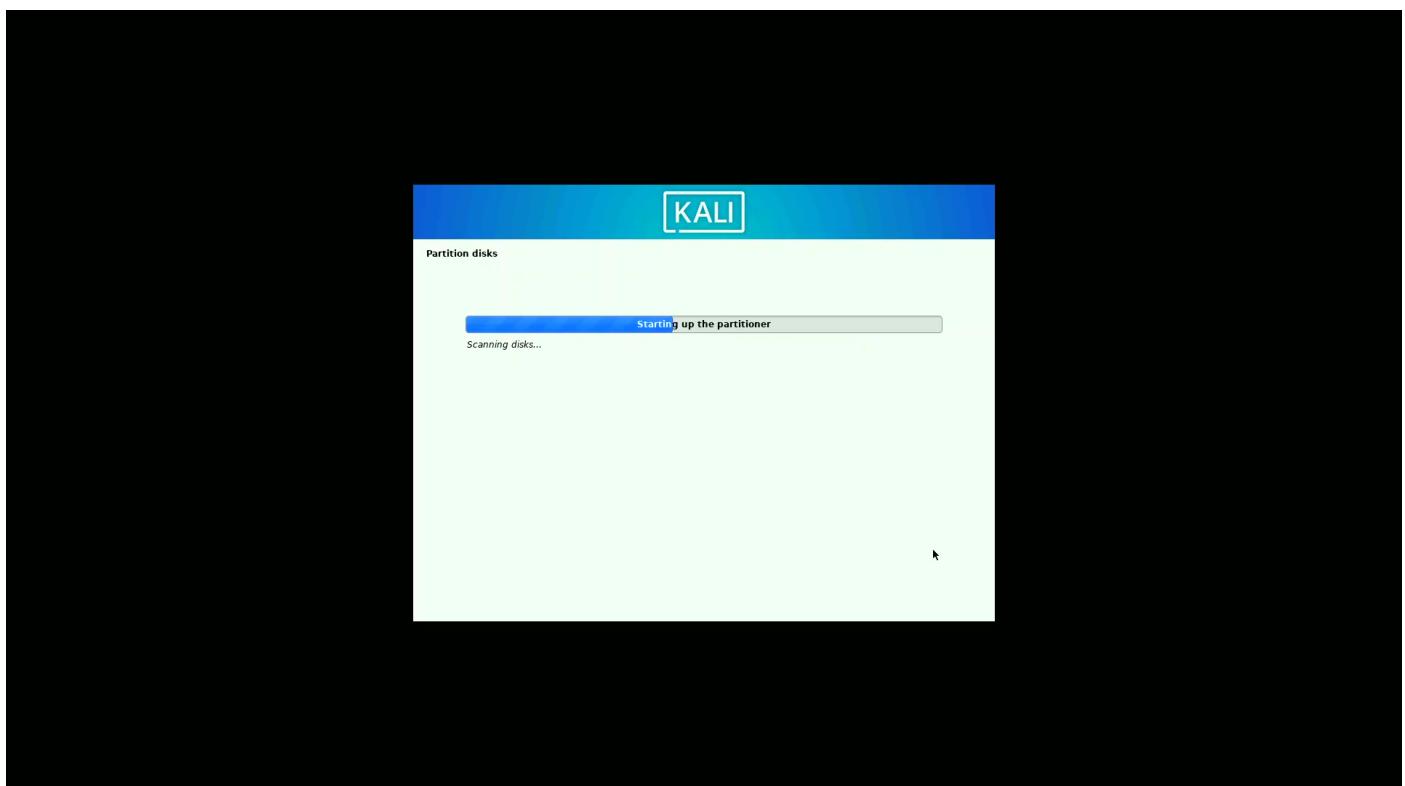
Step 8: Enter your full name click “Continue”.



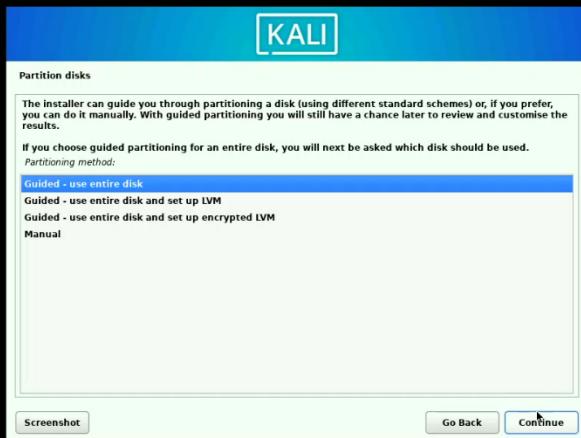
Step 9: Enter password click “Continue”.



Step 10: Wait for the disk manager to get loaded.



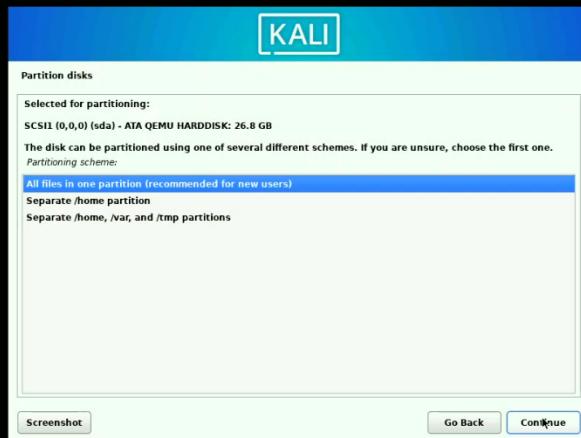
Step 11: Select “Guided – use entire disk” and click “Continue”.



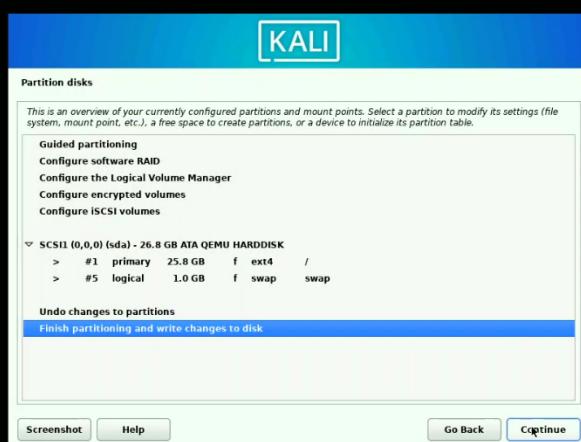
Step 12: Select Disk click “Continue”.



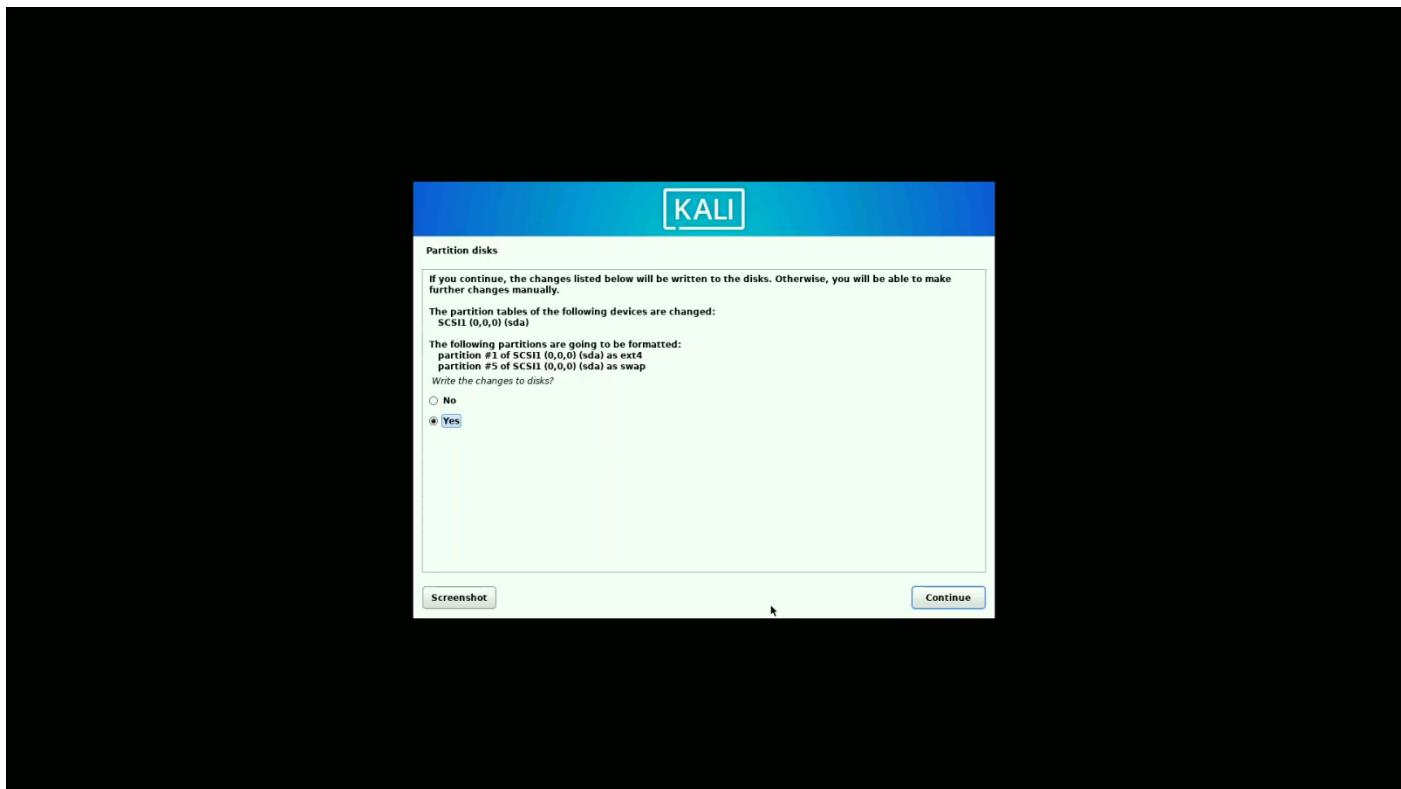
Step 13: Select “All files in one partition” and click “Continue”.



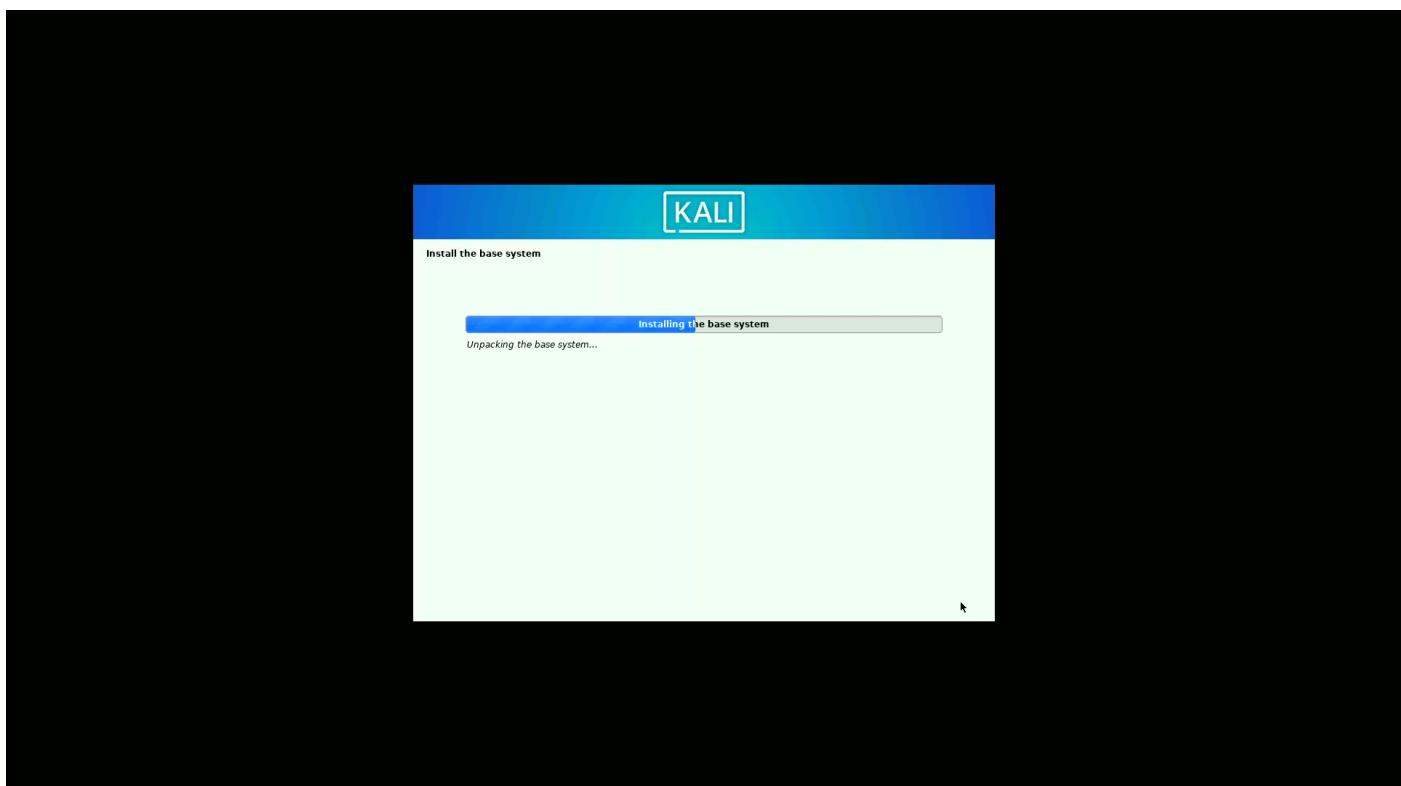
Step 14: Click “Continue”.



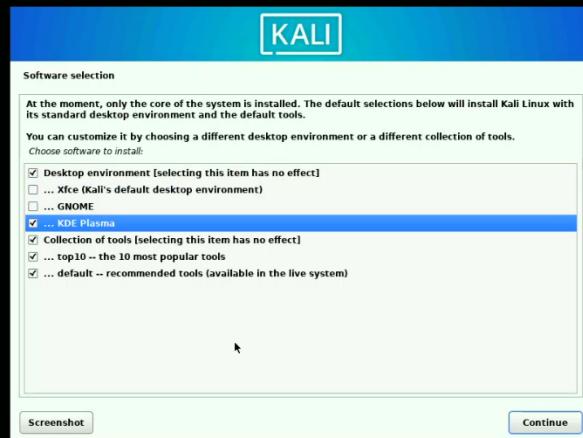
Step 15: Select “Yes” and click “Continue”.



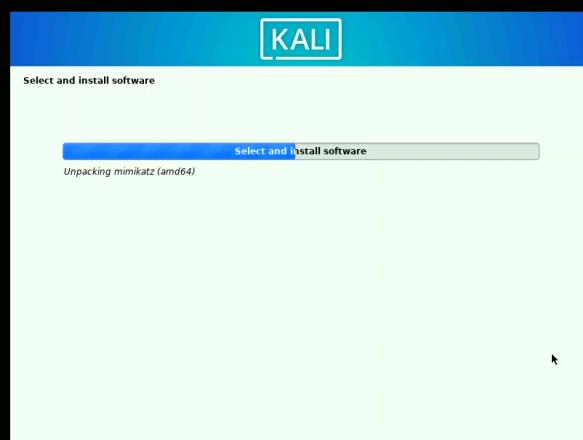
Step 16: Wait for the Base OS to get installed.



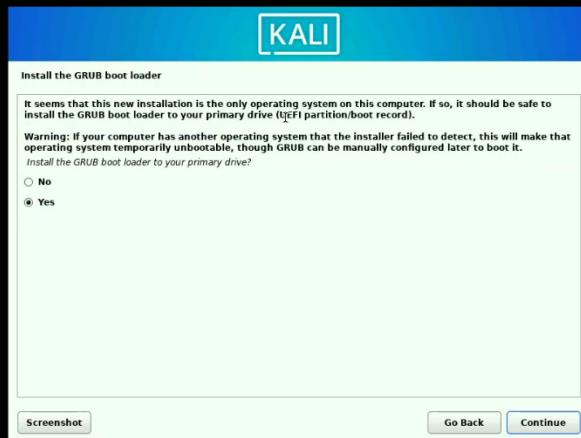
Step 17: Select the software's, Desktop Environment and click "Continue".



Step 18: Wait for the software's and DE to get installed.



Step 19: Select “Yes” and click “Continue”.



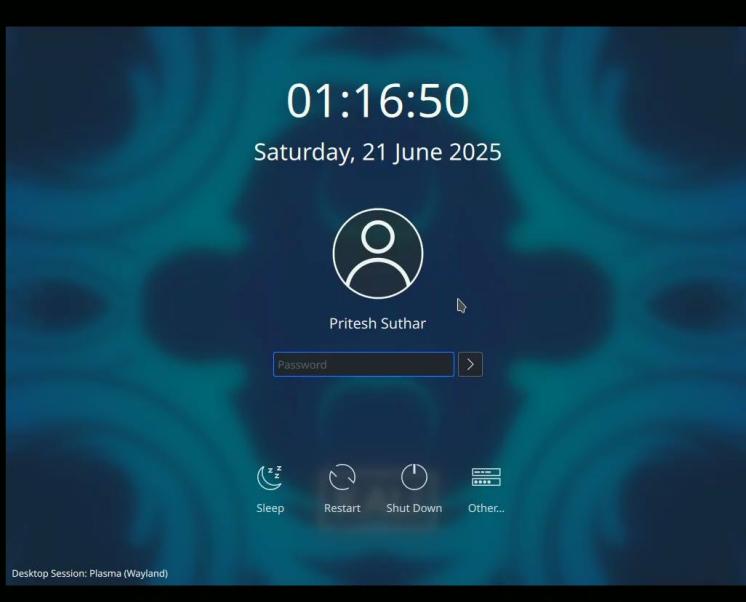
Step 20: Select disk and click “Continue”.



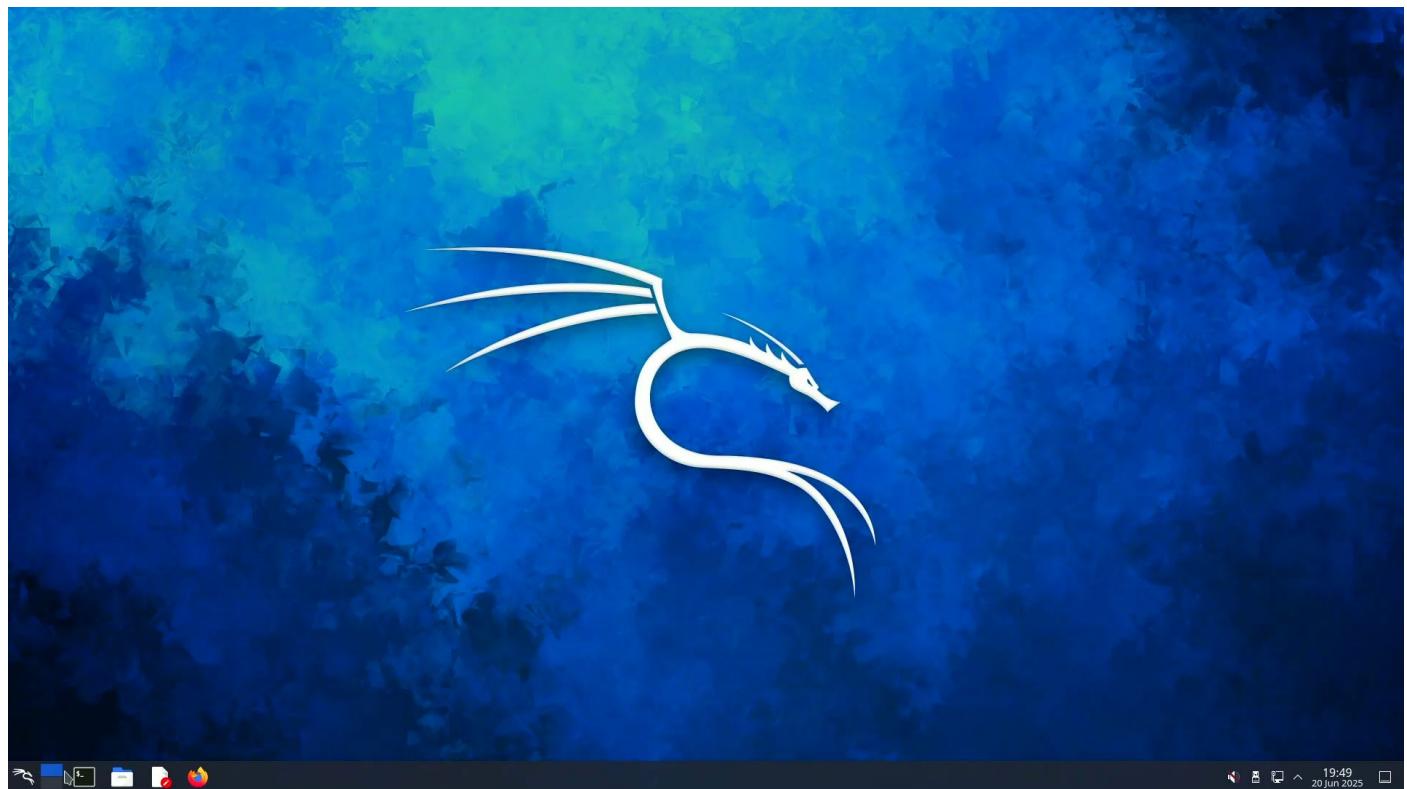
Step 21: Click “Continue”.



Step 22: Enter your password.



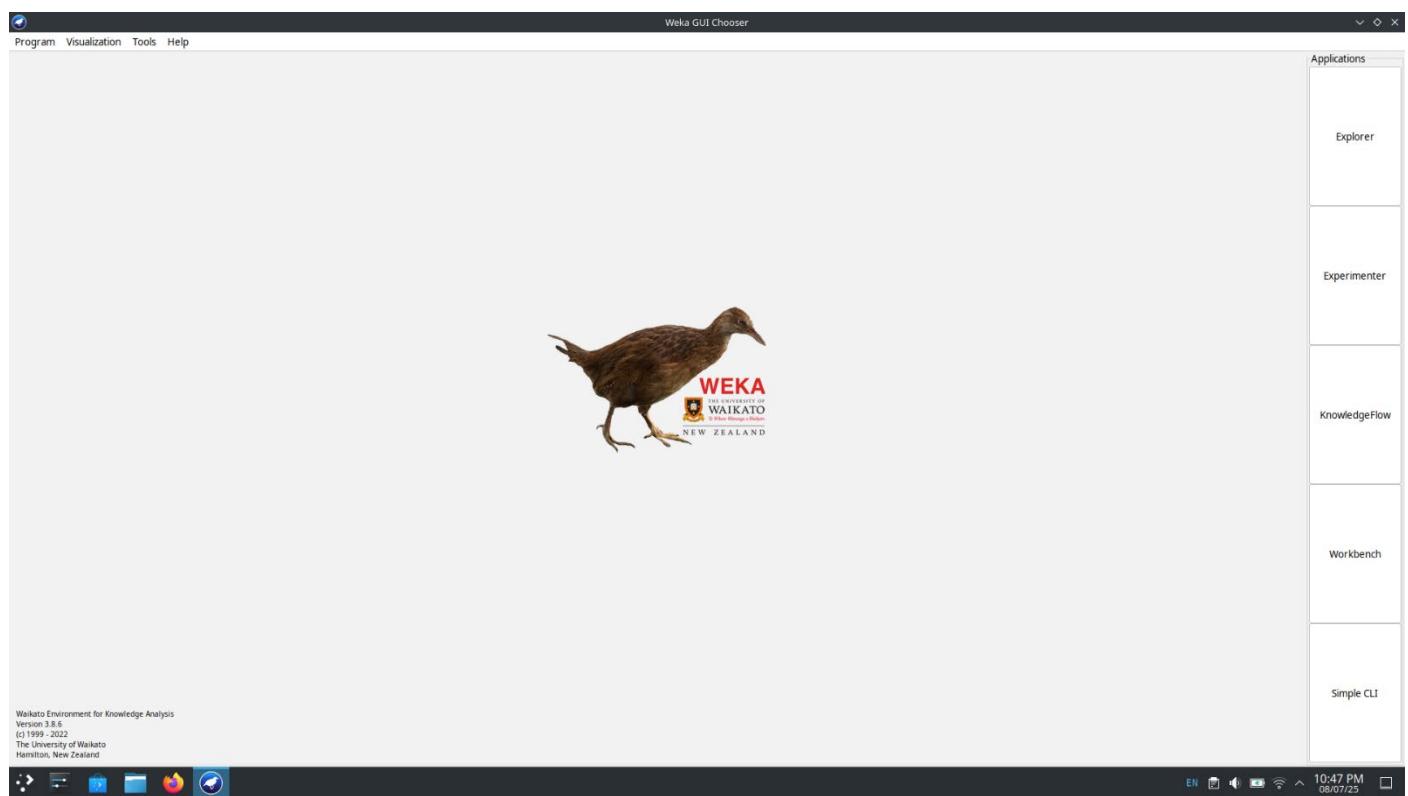
Step 23: Kali Linux is successfully installed.



## Practical – 2

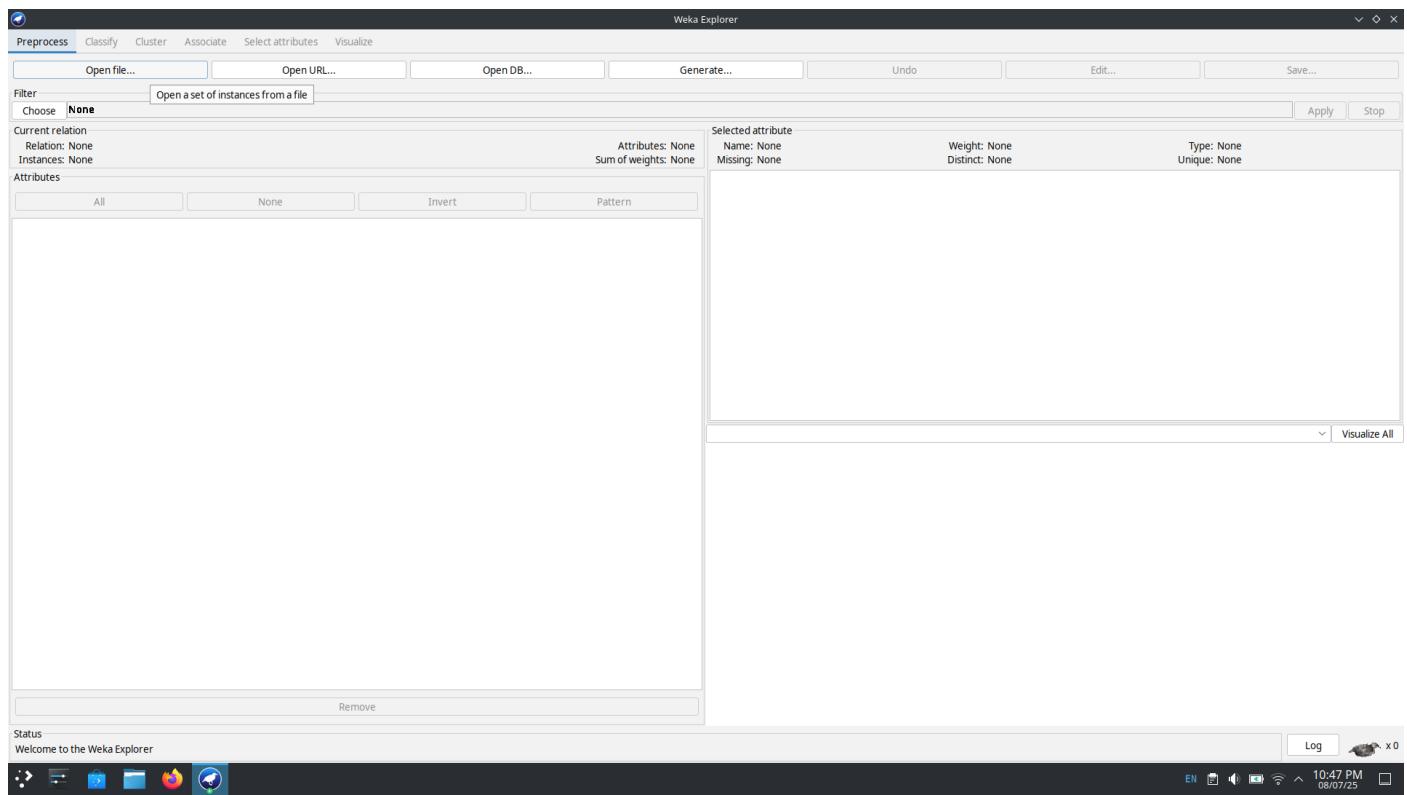
Aim: Explore WEKA Tool.

Step 1: Click on “Explorer”.

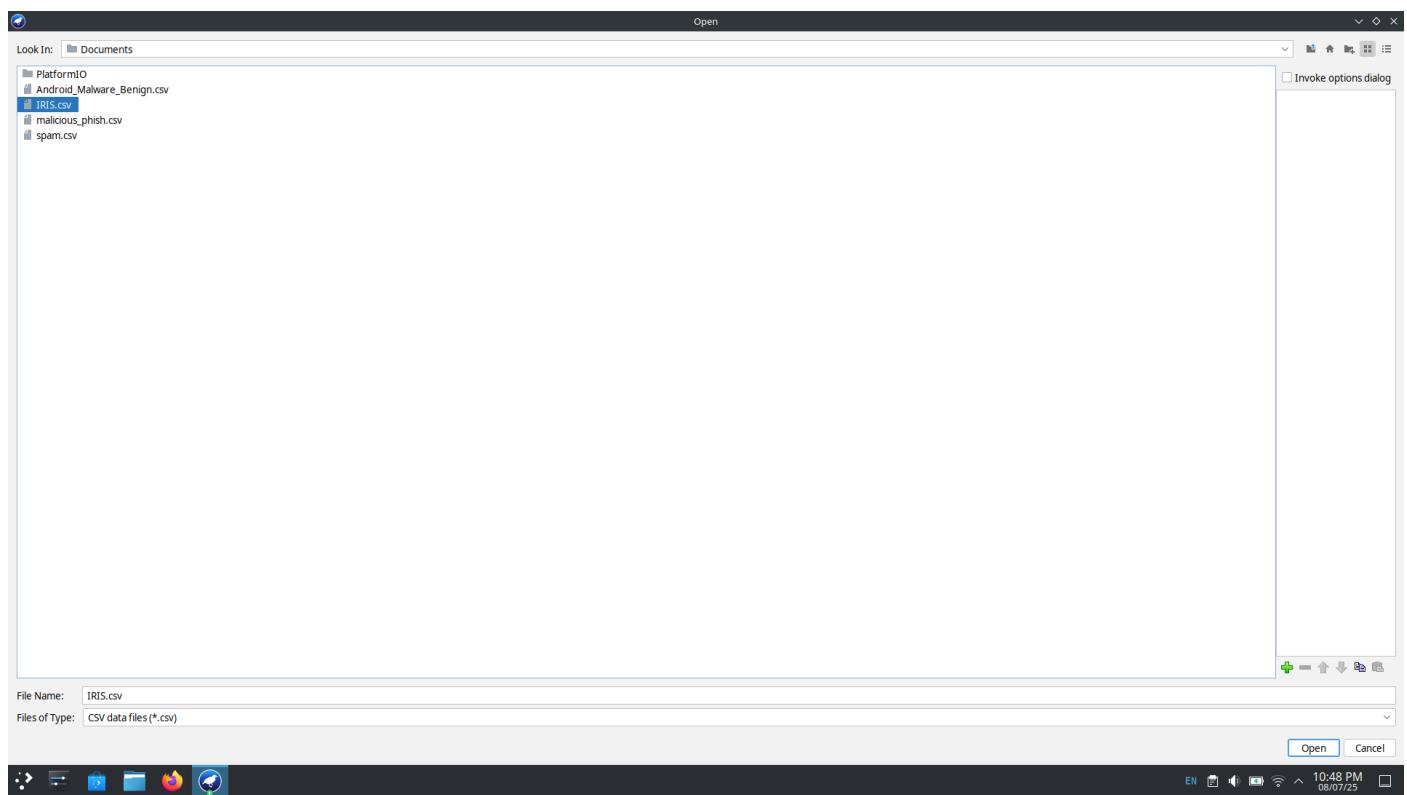




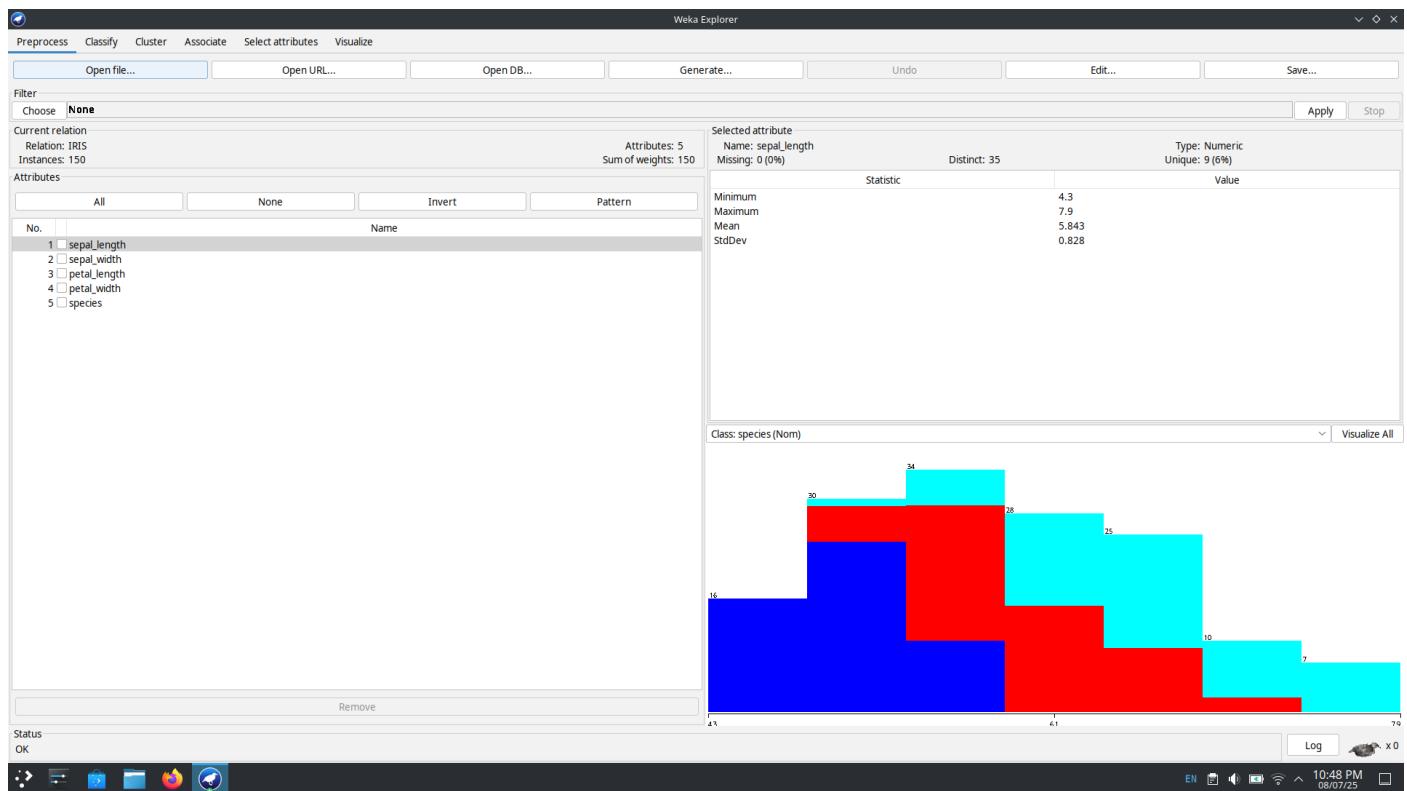
Step 2: Click on “Open File”.



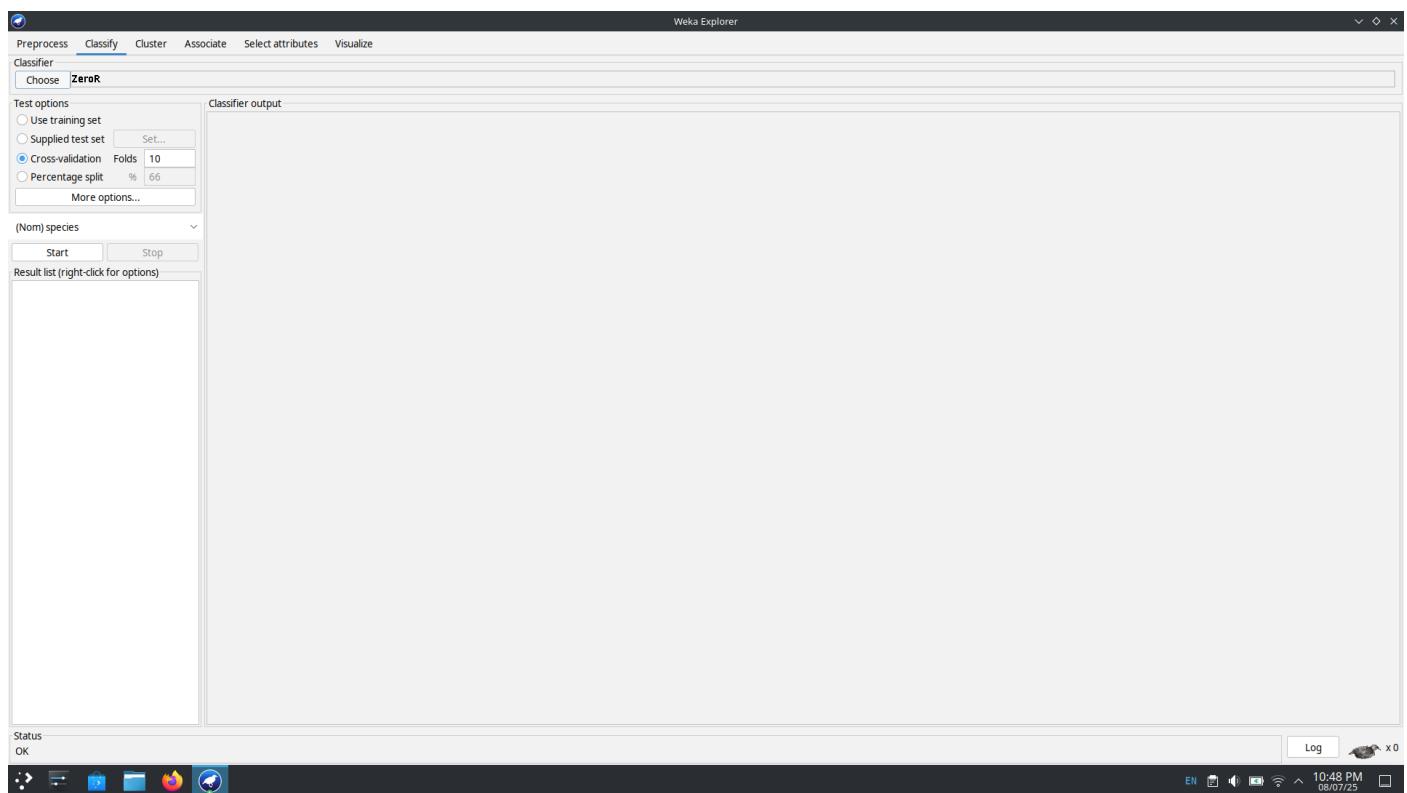
Step 3: Select and open “IRIS.csv”.



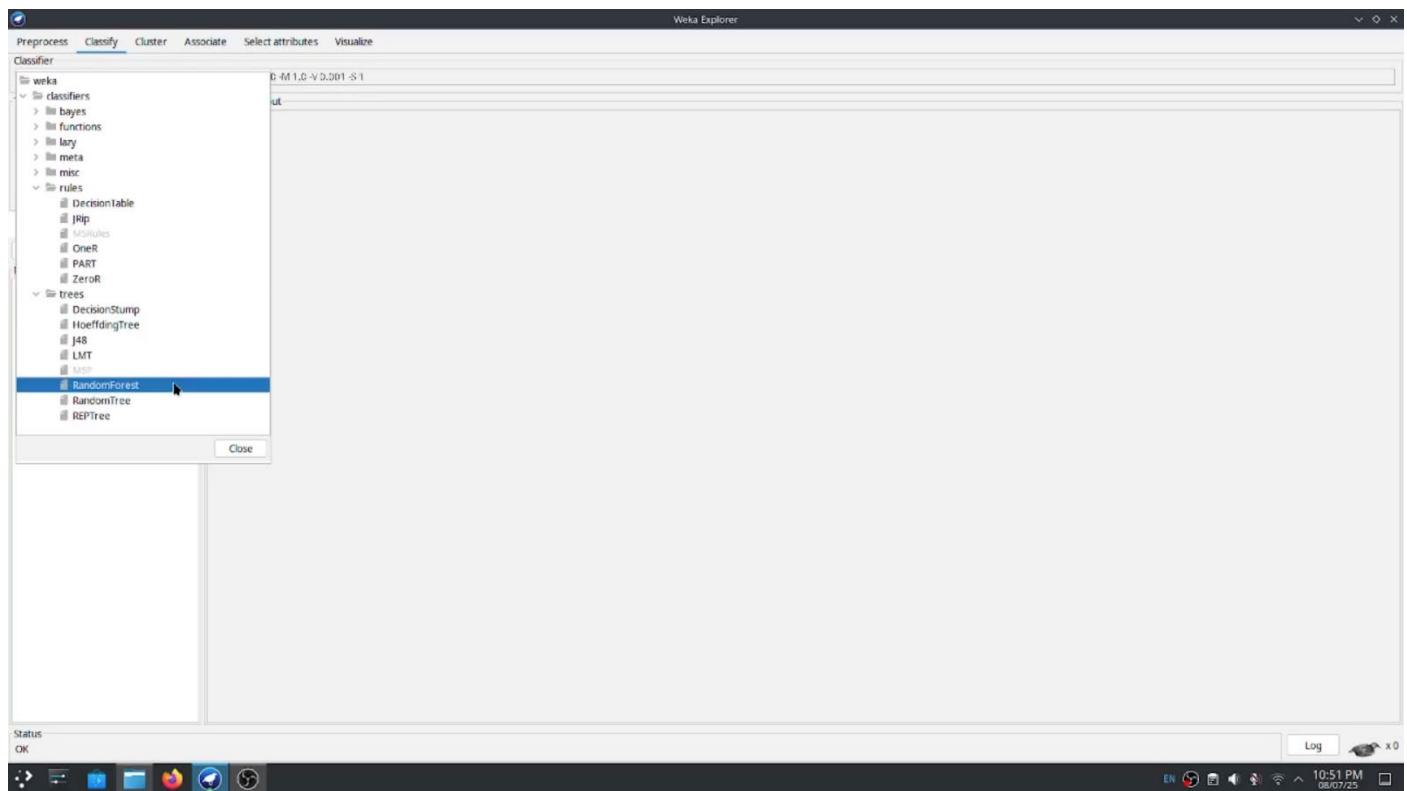
Step 4: Click on “Classify”.



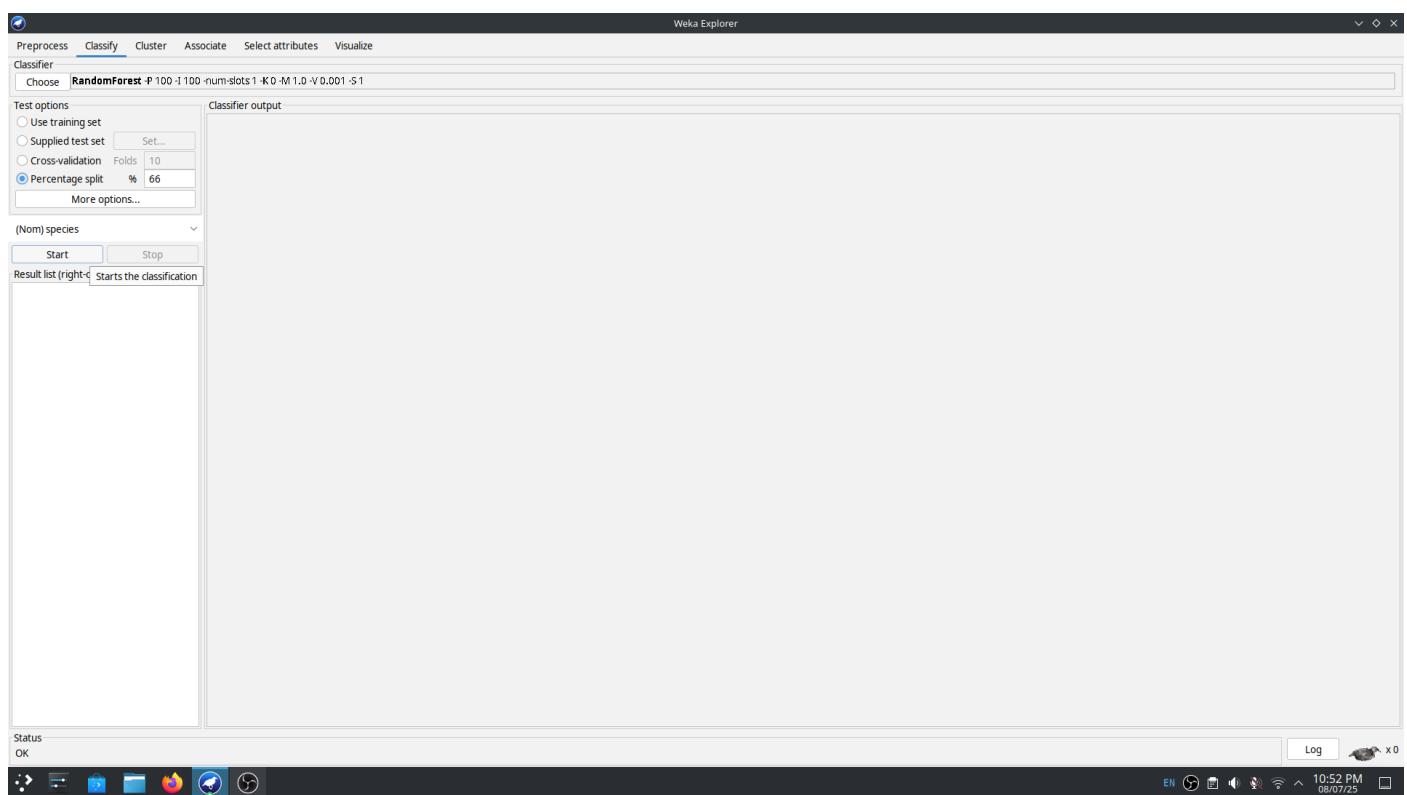
Step 5: Click on “Choose”.



Step 6: Select “RandomForest” available under “trees”.



Step 7: Select “Percentage Split”.



Step 8: Press on “Start” to train the model and review the output.

The screenshot shows the Weka Explorer interface with the following details:

- Classifier:** Choose **RandomForest** with parameters **-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1**.
- Test options:** Use training set.
- Classifier output:**
  - Run information:** Scheme: weka.classifiers.trees.RandomForest - P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1, Relation: IRIS, Instances: 150, Attributes: 5, Test mode: split 66.0% train, remainder test.
  - Classifier model (full training set):** RandomForest
  - Evaluation on test split:** Time taken to build model: 0.06 seconds, Time taken to test model on test split: 0.01 seconds.
  - Summary:**

	Correctly Classified Instances	96.0784 %
Incorrectly Classified Instances	2	3.9216 %
Kappa statistic	0.9408	
Mean absolute error	0.0349	
Root mean squared error	0.1432	
Relative absolute error	7.8349 %	
Root relative squared error	30.2995 %	
Total Number of Instances	51	
  - Detailed Accuracy By Class:**

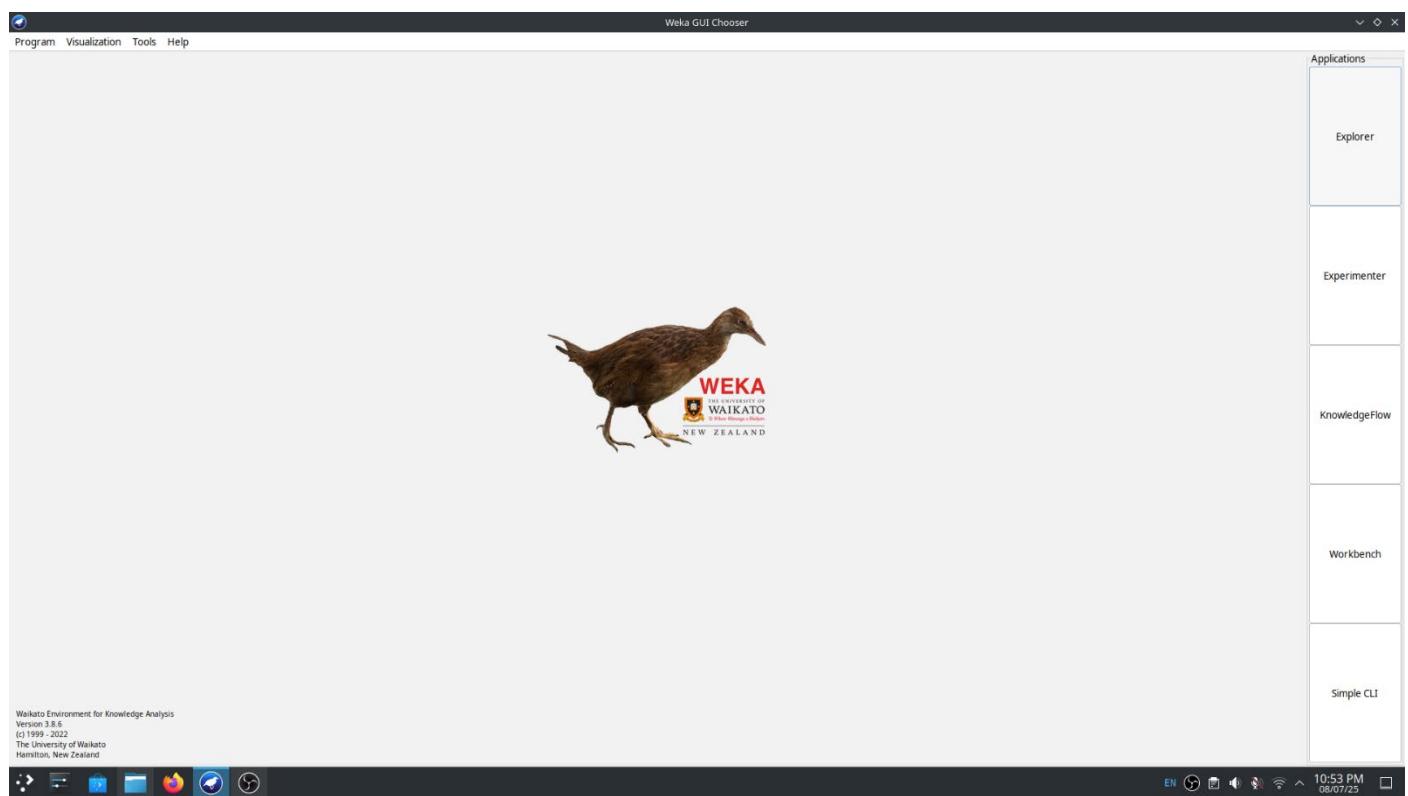
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-setosa
1.000	0.063	0.905	1.000	0.950	0.921	0.994	0.990	0.990	Iris-versicolor
0.882	0.000	1.000	0.882	0.938	0.913	0.994	0.987	0.987	Iris-virginica
Weighted Avg.	0.961	0.023	0.995	0.961	0.942	0.996	0.992	0.992	
  - Confusion Matrix:**

a b c	..- classified as
15 0 0	a = Iris-setosa
0 19 0	b = Iris-versicolor
0 2 15	c = Iris-virginica

## Practical – 3

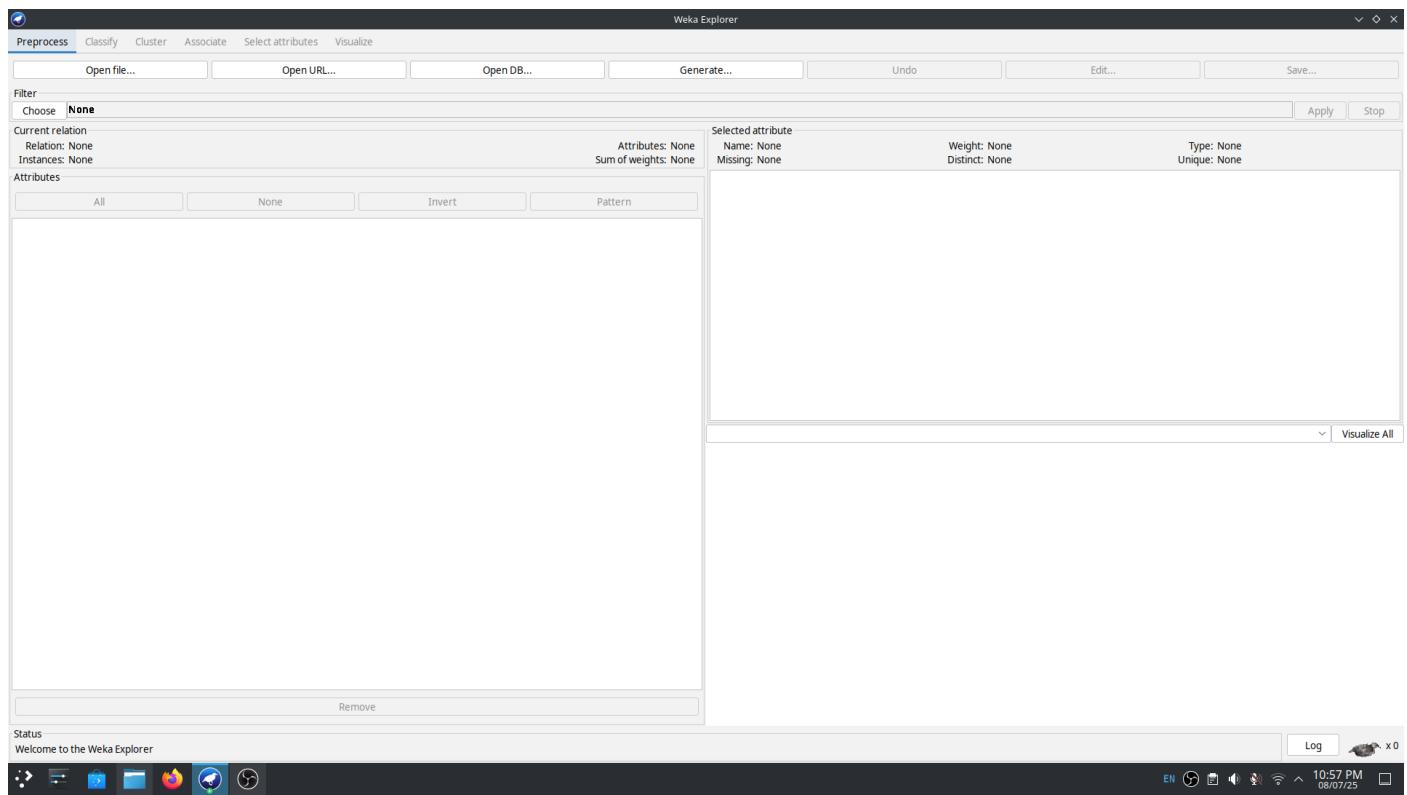
Aim: Android malware detection using WEKA tool.

Step 1: Click on “Explorer”.

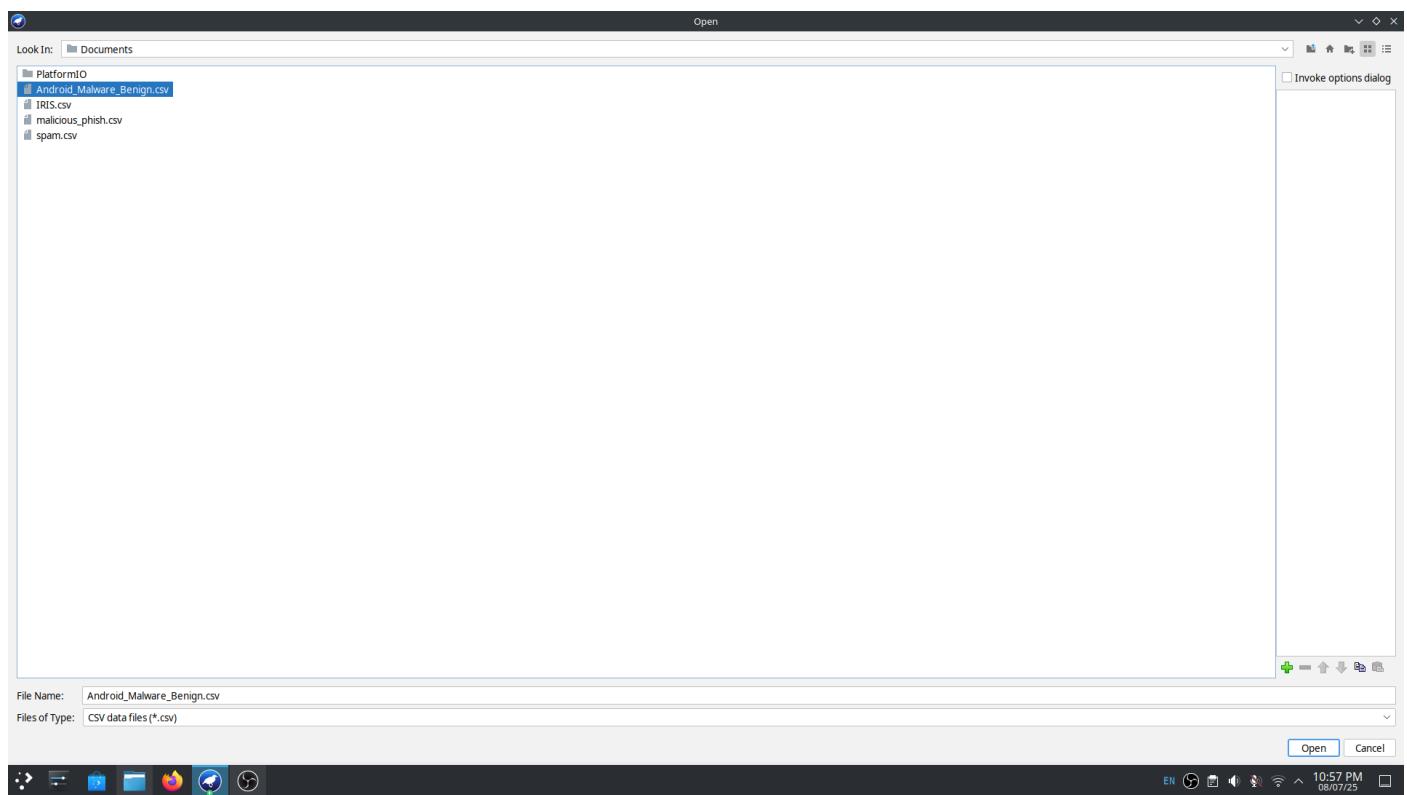




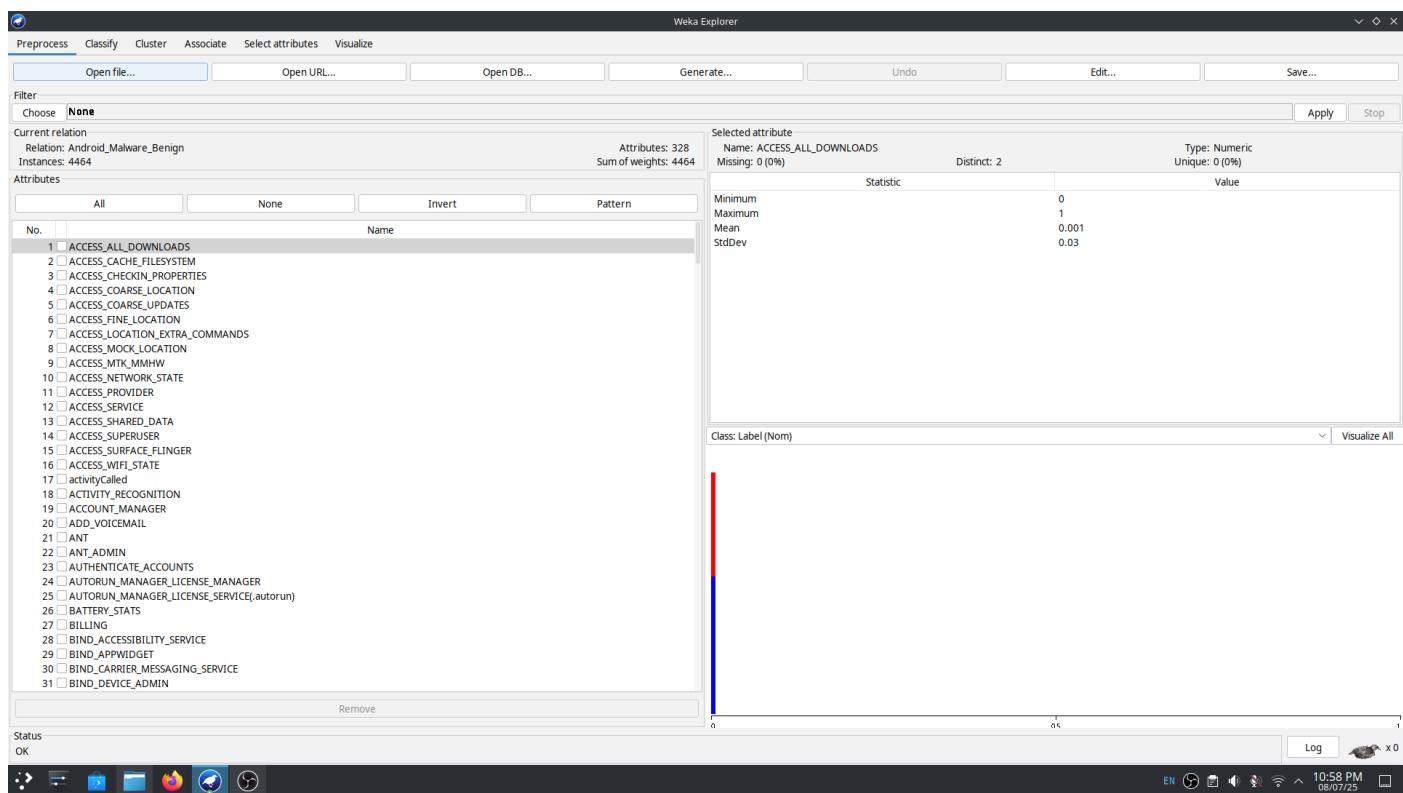
Step 2: Click on “Open File”.



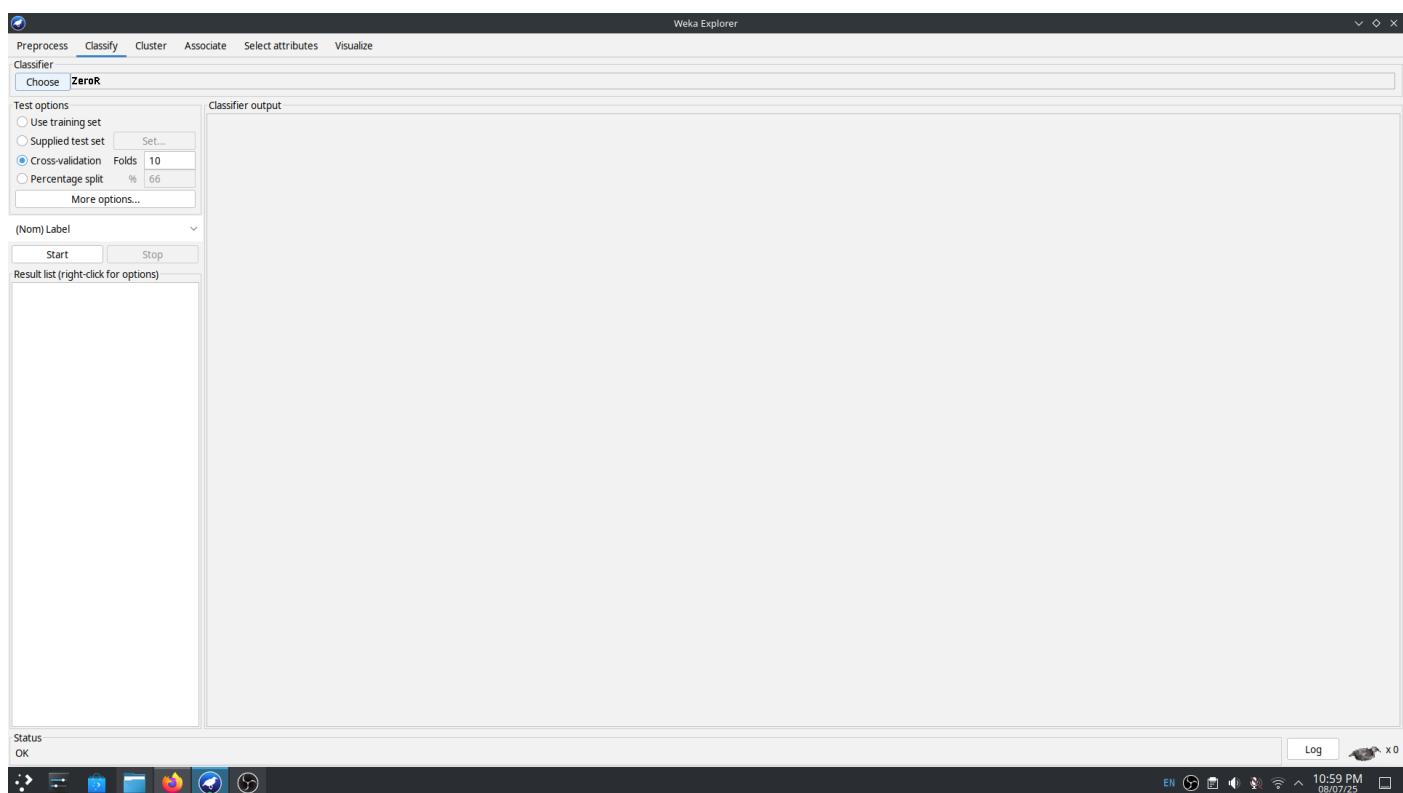
Step 3: Select and open “Android\_Malware\_Benign.csv”.



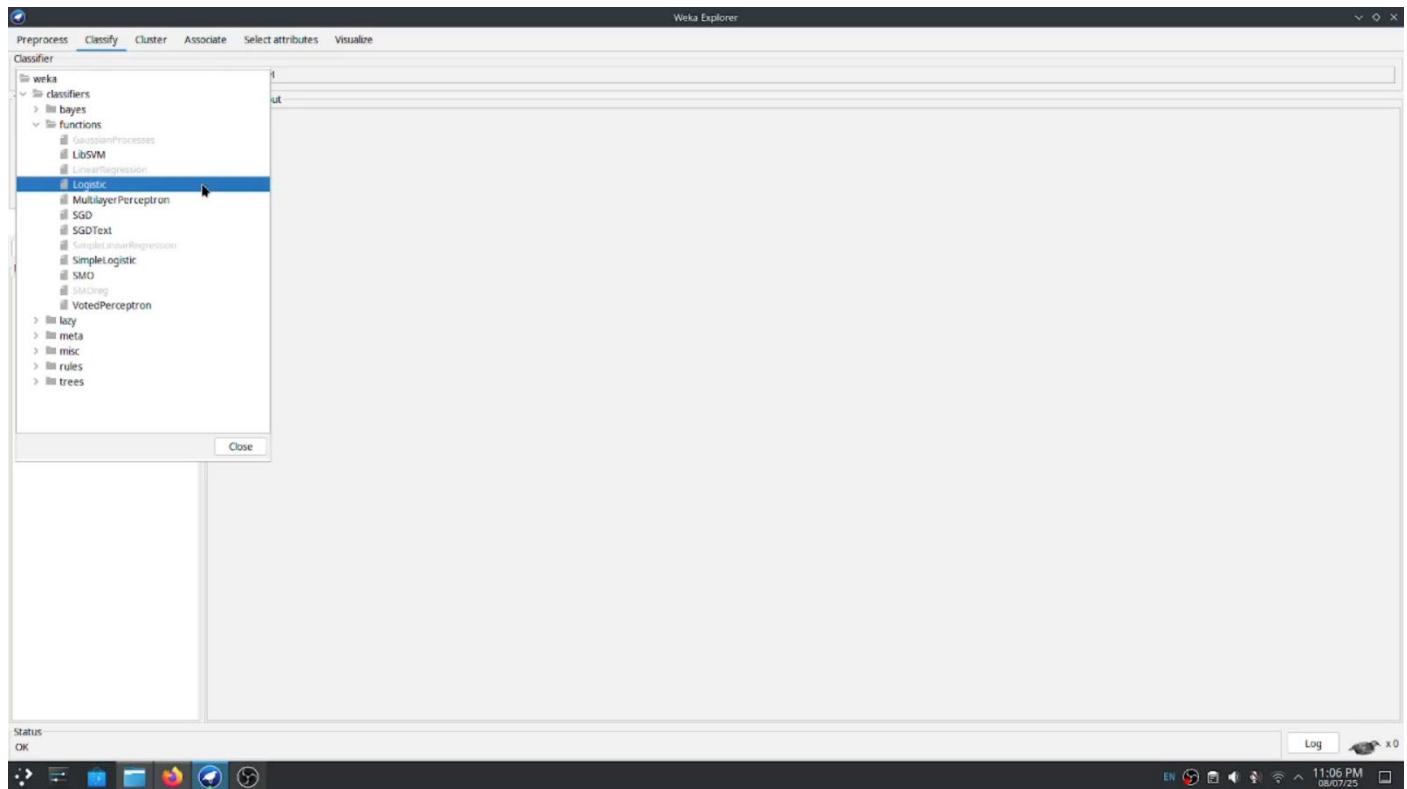
Step 4: Click on “Classify”.



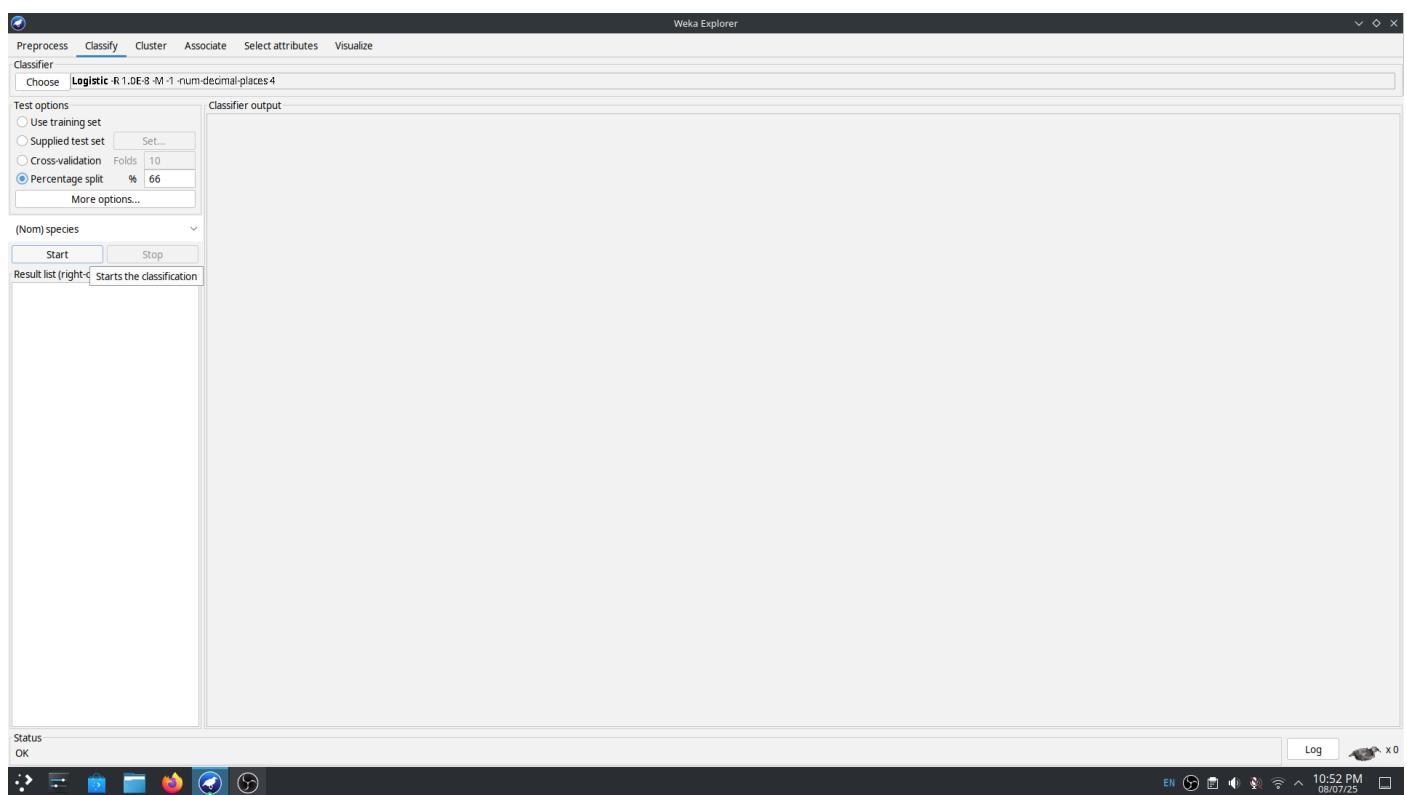
Step 5: Click on “Choose”.



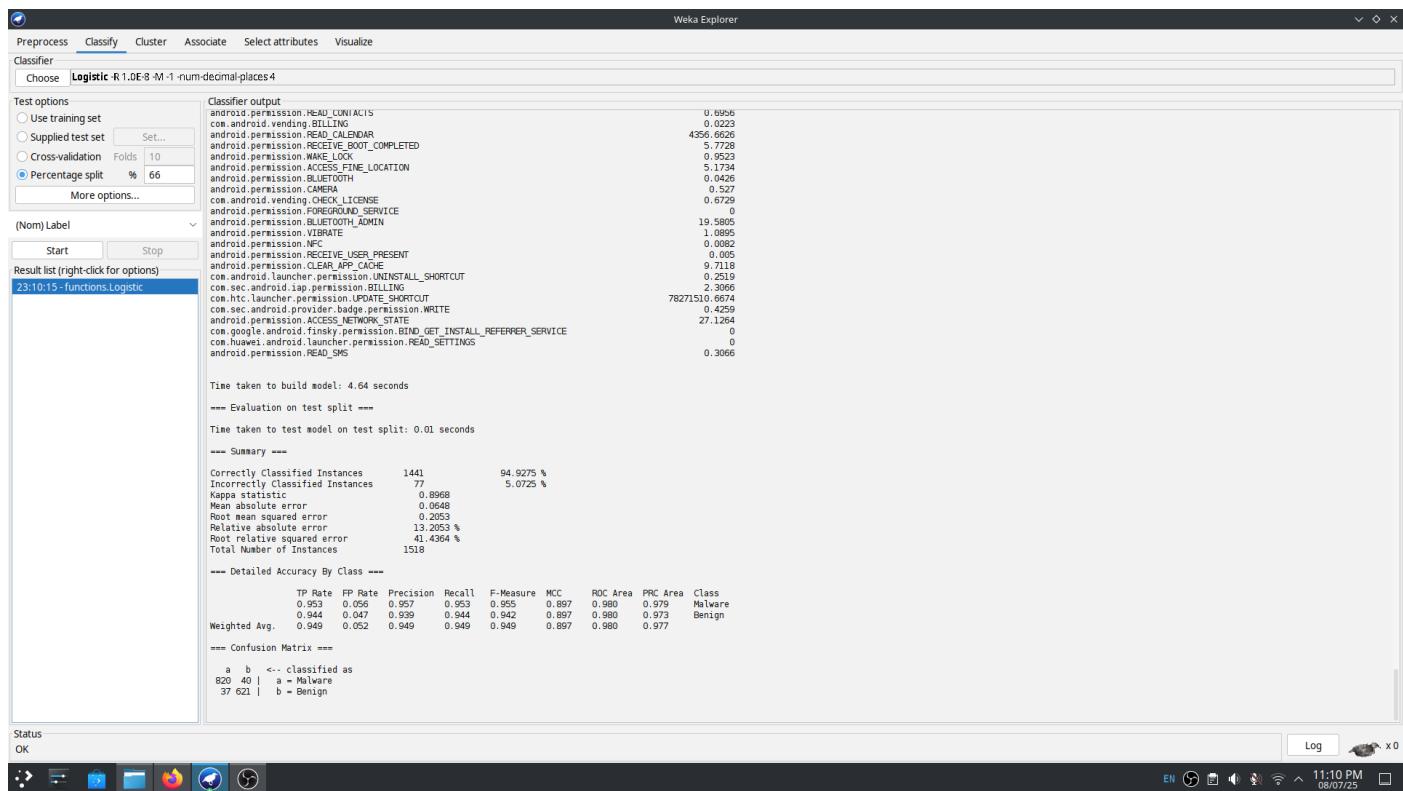
Step 6: Choose “Logistic” available under “functions”.



Step 7: Select “Percentage Split”.



Step 8: Click “Start” to train the model and review the output.



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **Logistic**: R:1.0E-8 M:-1 num-decimal-places 4

Test options:

- Use training set
- Supplied test set Set...
- Cross-validation Folds 10
- Percentage split % 66

(Nom) Label Start Stop

Result list (right-click for options) 23:10:15-functions-Logistic

Permission	Support
android.permission.HOME_ALIASES	0.0045
com.android.alarm.permission.BILLING	0.0223
android.permission.READ_CALENDAR	4256.6636
android.permission.RECEIVE_BOOT_COMPLETED	5.7728
android.permission.WAKE_LOCK	0.9523
android.permission.ACCESS_FINE_LOCATION	5.1734
android.permission.ACCESS_COARSE_LOCATION	0.0405
android.permission.CAMERA	0.527
com.android.vending.CHECK_LICENSE	0.6729
android.permission.FOREGROUND_SERVICE	0
android.permission.BLUETOOTH_ADMIN	19.5986
android.permission.VIBRATE	1.0895
android.permission.NFC	0.0082
android.permission.RECEIVE_USER_PRESENT	0.005
android.permission.CLEAR_APP_CACHE	9.7118
com.sec.android.provider.permission.UNINSTALL_SHORTCUT	0.2119
com.sec.android.permission.UPDATE_SHORTCUT	2.3066
com.sec.android.provider.badge.permission.WRITE	78271510.6674
android.permission.ACCESS_NETWORK_STATE	0.4259
com.android.permission.BIND_GET_INSTALL_REFERRER_SERVICE	27.1264
com.huawei.android.launcher.permission.READ_SETTINGS	0
android.permission.READ_SMS	0.3066

Time taken to build model: 4.64 seconds

\*\*\* Evaluation on test split \*\*\*

Time taken to test model on test split: 0.01 seconds

\*\*\* Summary \*\*\*

Category	Value	Description
Correctly Classified Instances	144	94.9275 %
Incorrectly Classified Instances	77	5.0725 %
Kappa statistic	0.8968	
Mean absolute error	0.0648	
Root mean square error	0.2053	
Relative absolute error	13.2053 %	
Root relative squared error	41.4264 %	
Total Number of Instances	1518	

\*\*\* Detailed Accuracy By Class \*\*\*

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Malware	0.953	0.056	0.957	0.953	0.955	0.897	0.980	0.979
Benign	0.944	0.047	0.959	0.944	0.942	0.897	0.980	0.973
Weighted Avg.	0.949	0.052	0.949	0.949	0.949	0.897	0.980	0.977

\*\*\* Confusion Matrix \*\*\*

	a	b	<- classified as
a	820	40	a = Malware
b	37	621	b = Benign

## Practical – 4

Aim: Perform malicious URL detection using SVM algorithm.

Cell:

```
import pandas as pd
import numpy as np
from urllib.parse import urlparse
import re
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler
from tld import get_tld, exceptions
import matplotlib.pyplot as plt
import seaborn as sns
```

Cell:

```
df = pd.read_csv('malicious_phish.csv')

sample_size = 50000
df = df.sample(n=sample_size, random_state=42).reset_index(drop=True)

expected_url_col = 'url'
expected_label_col = 'type'

df.dropna(subset=['url', 'type'], inplace=True)
df['Label'] = df['type'].apply(lambda x: 0 if x.lower() == 'benign' else 1)

print("\nBinary Class Distribution (0=Benign, 1=Malicious):")
print(df['Label'].value_counts())
```

Output:

Binary Class Distribution (0=Benign, 1=Malicious):

Label

0 32914

1 17086

Name: count, dtype: int64

Cell:

```
def get_num_directories(url):
    try:
        path = urlparse(url).path
        return path.count('/')
    except:
        return 0

def is_ip_address_url(url):
    try:
        host = urlparse(url).netloc
        if re.match(r"^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$", host):
            return 1
        return 0
    except:
        return 0

def get_tld_length(url):
    try:
        tld_info = get_tld(url, as_object=True, fail_silently=True)
        if tld_info:
            return len(tld_info.tld)
        return 0
    except exceptions.TldDomainNotFound:
        return 0
    except:
        return 0

sensitive_words = ['login', 'bank', 'account', 'update', 'verify', 'paypal',
'ebay', 'amazon', 'security', 'webscr', 'confirm']

def count_sensitive_words(url):
    count = 0
    url_lower = url.lower()
    for word in sensitive_words:
        if word in url_lower:
            count += 1
    return count
```

Cell:

```
df['url_length'] = df[expected_url_col].apply(len)

df['num_dots'] = df[expected_url_col].apply(lambda x: x.count('.'))

df['has_at_symbol'] = df[expected_url_col].apply(lambda x: 1 if '@' in x else 0)

df['has_double_slash_in_path'] = df[expected_url_col].apply(lambda x: 1 if x.count('//') > 1 else 0)

df['num_hyphens'] = df[expected_url_col].apply(lambda x: x.count('-'))

df['num_question_marks'] = df[expected_url_col].apply(lambda x: x.count('?'))

df['num_equals'] = df[expected_url_col].apply(lambda x: x.count('='))

df['num_directories'] = df[expected_url_col].apply(get_num_directories)

df['is_ip_address'] = df[expected_url_col].apply(is_ip_address_url)

df['hostname_length'] = df[expected_url_col].apply(lambda x:
len(urlparse(x).netloc))

df['tld_length'] = df[expected_url_col].apply(get_tld_length)

df['has_https'] = df[expected_url_col].apply(lambda x: 1 if x.startswith('https')
else 0)

df['num_sensitive_words'] = df[expected_url_col].apply(count_sensitive_words)
```

Cell:

```
features = [
    'url_length', 'num_dots', 'has_at_symbol', 'has_double_slash_in_path',
    'num_hyphens', 'num_question_marks', 'num_equals', 'num_directories',
    'is_ip_address', 'hostname_length', 'tld_length', 'has_https',
    'num_sensitive_words'
]

X = df[features]
y = df['Label']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```



Cell:

```
x_train, x_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
                                                    random_state=42, stratify=y)

print(f"\nTraining set size: {x_train.shape[0]}")
print(f"Testing set size: {x_test.shape[0]}")
print(f"Training set Malicious ratio: {y_train.sum() / len(y_train):.2f}")
print(f"Testing set Malicious ratio: {y_test.sum() / len(y_test):.2f}")
```

Output:

```
Training set size: 40000
Testing set size: 10000
Training set Malicious ratio: 0.34
Testing set Malicious ratio: 0.34
```

Cell:

```
svm_model = SVC(kernel='rbf', C=1.0, random_state=42, verbose=True,
                  shrinking=False)

svm_model.fit(X_train, y_train)
```

Output:

```
[LibSVM].....*
optimization finished, #iter = 10916
obj = -7960.662193, rho = 0.242135
nSV = 8948, nBSV = 8515
Total nSV = 8948
```

Cell:

```
y_pred = svm_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"\nAccuracy: {accuracy:.4f}")

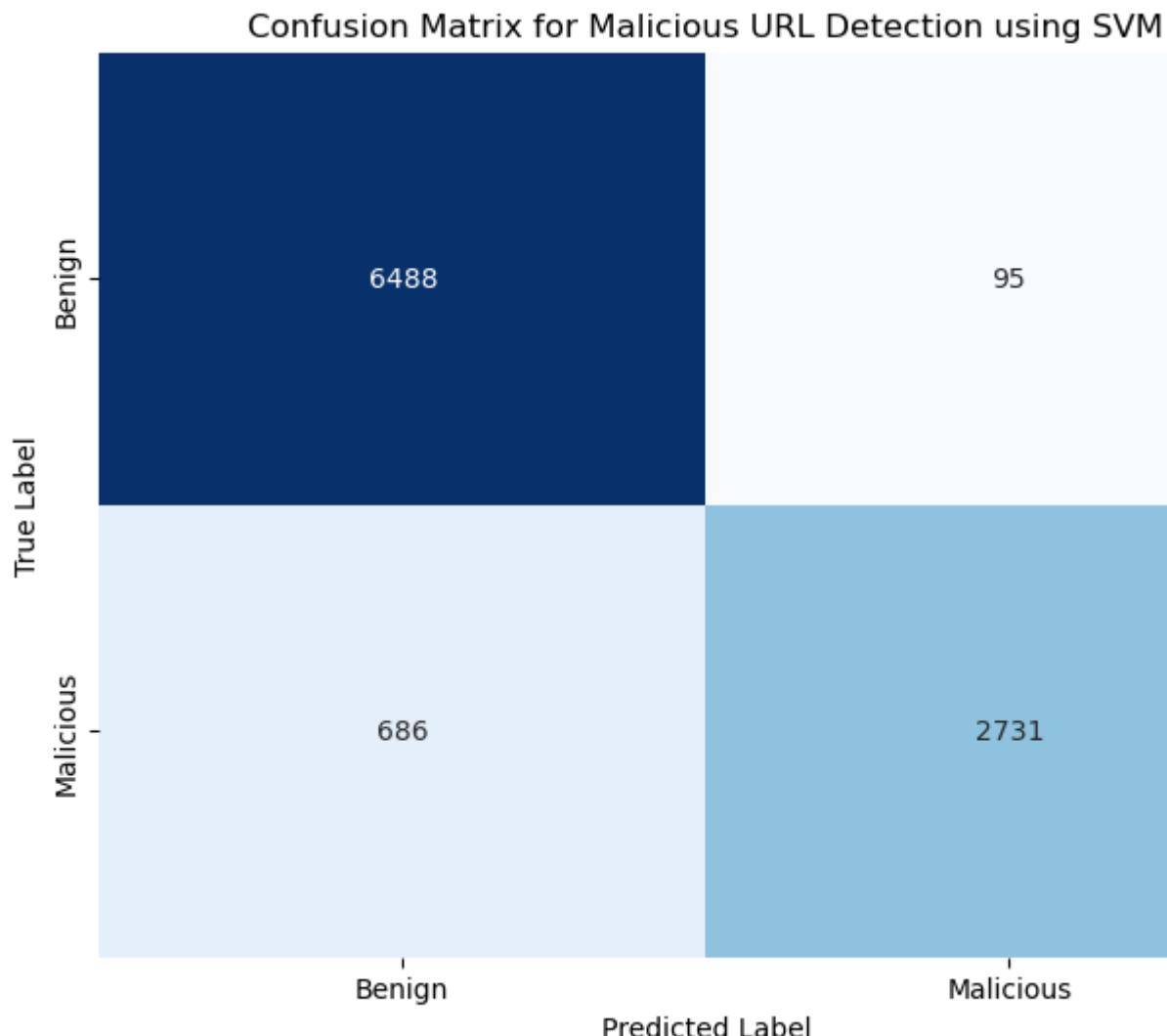
print("\nConfusion Matrix:")
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['Benign', 'Malicious'],
            yticklabels=['Benign', 'Malicious'])

plt.title('Confusion Matrix for Malicious URL Detection using SVM')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

Output:

**Accuracy: 0.9219**

Confusion Matrix:



## Practical – 5

Aim: Perform SMS spam detection using Logistic regression algorithm.

Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('spam.csv', encoding='latin-1')[['v1', 'v2']]
df.columns = ['label', 'text']

df['label_num'] = df['label'].map({'ham': 0, 'spam': 1})
X_train, X_test, y_train, y_test = train_test_split(
    df['text'], df['label_num'], test_size=0.2, random_state=99
)

vectorizer = CountVectorizer()

X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
clf_lr = LogisticRegression(solver='liblinear', random_state=42)
clf_lr.fit(X_train_vec, y_train)
y_pred_lr = clf_lr.predict(X_test_vec)

accuracy = accuracy_score(y_test, y_pred_lr)
cm = confusion_matrix(y_test, y_pred_lr)

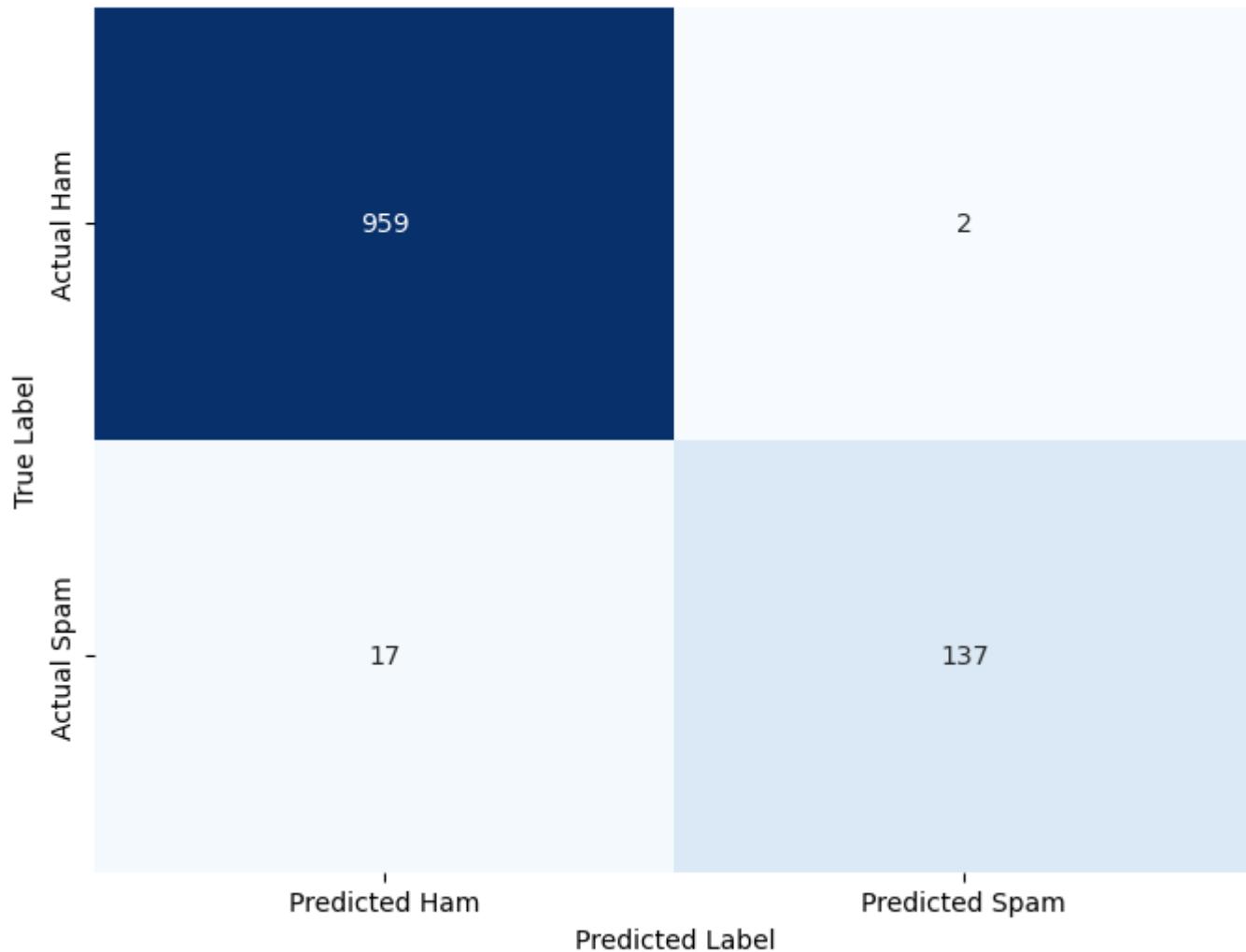
print(f'Accuracy: {accuracy:.4f}')

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['Predicted Ham', 'Predicted Spam'],
            yticklabels=['Actual Ham', 'Actual Spam'])
plt.title('Confusion Matrix for Multinomial Naive Bayes')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

Output:

Accuracy: 0.9830

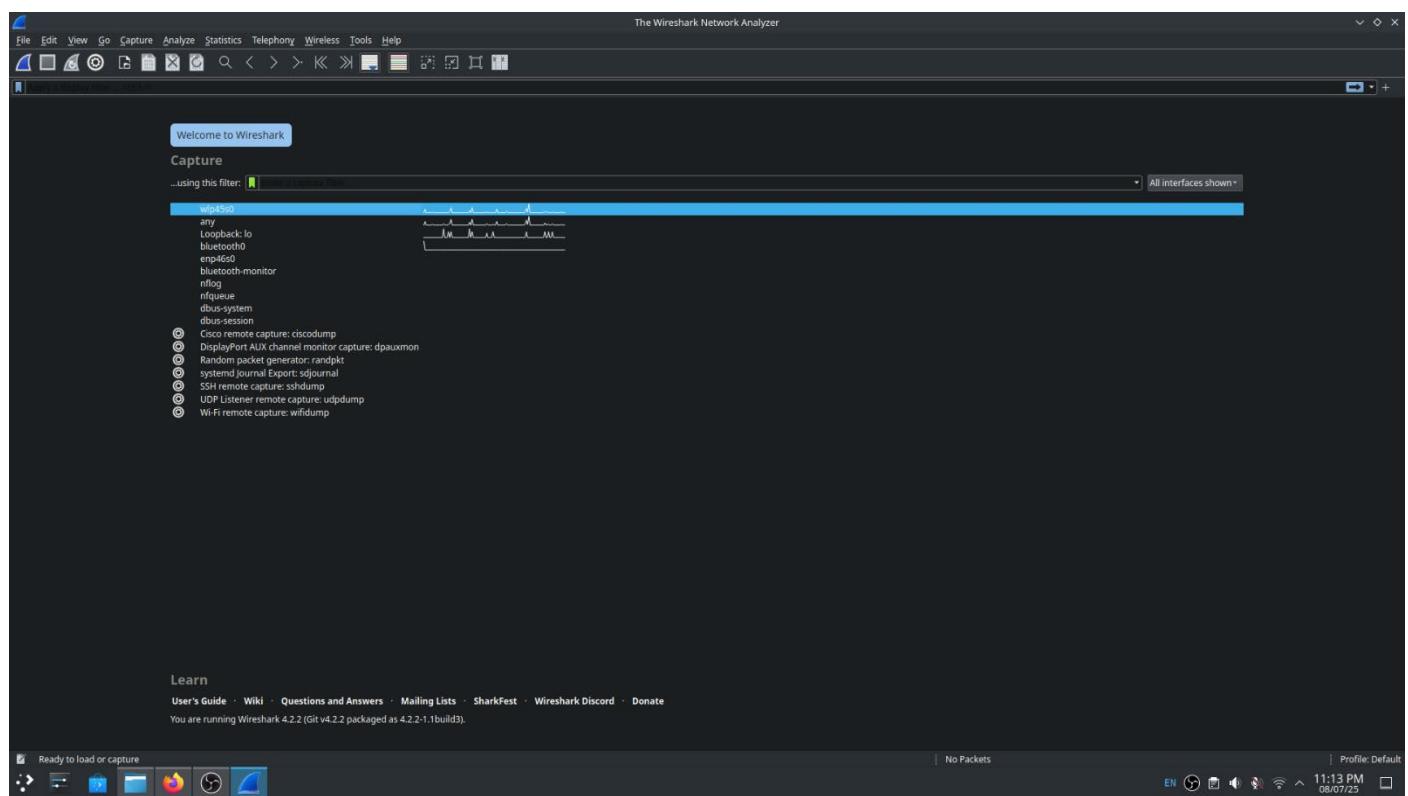
Confusion Matrix for Multinomial Naive Bayes



## Practical – 6

Aim: Capture and analyse network traffic using packet capture tools

Step 1: Select and open “wlp4s0”.





Step 2: Type “tcp” in filters section and hit Enter to view tcp packets.

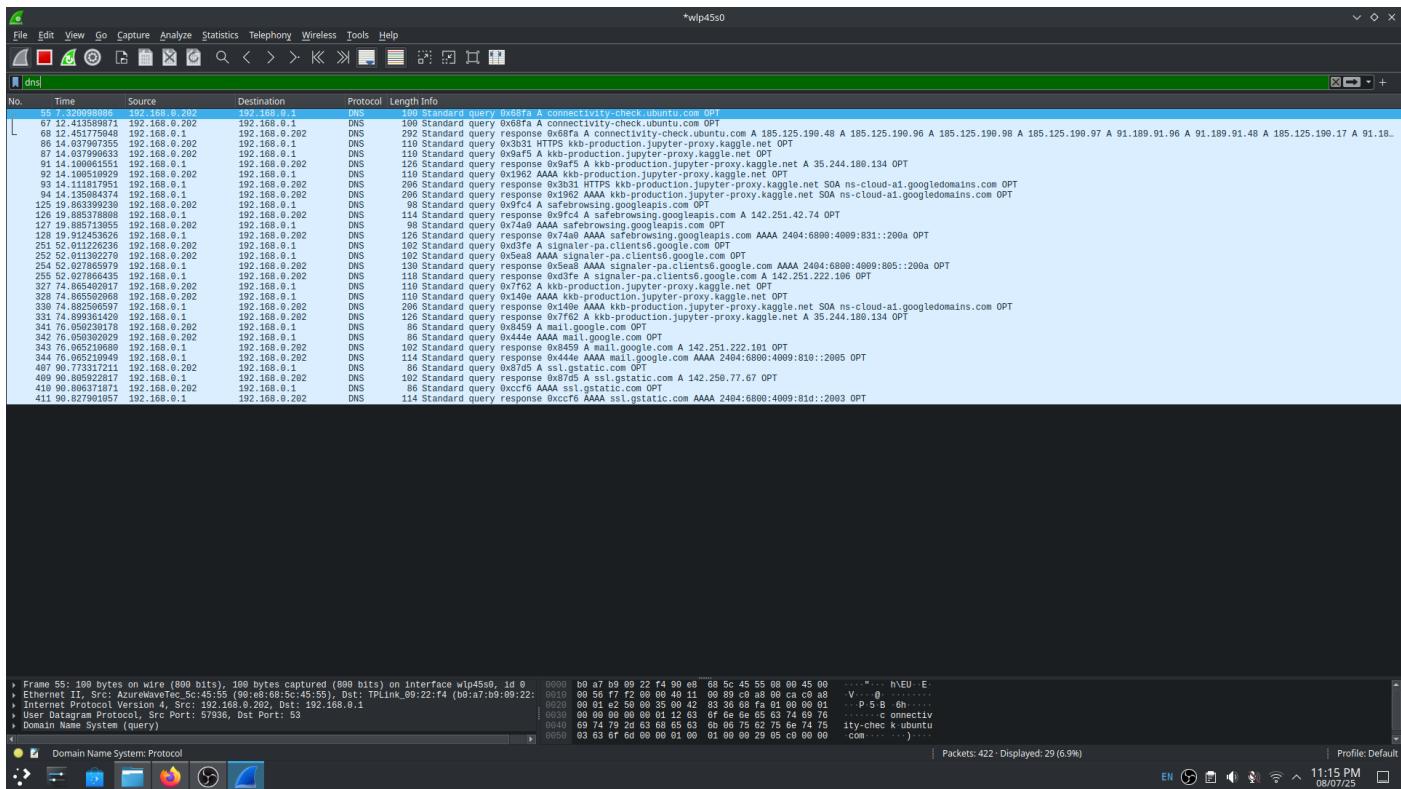
Wireshark Screenshot showing TCP traffic. The interface is wlp4s0. The traffic shows multiple ACK and FIN/ACK frames being exchanged between two hosts, likely during a slow start or recovery phase of a connection. The packet details and bytes panes show the structure of the TCP segments.

Step 3: Type “udp” in filters section and hit Enter to view udp packets.

Wireshark Screenshot showing UDP traffic. The interface is wlp4s0. The traffic shows multiple UDP datagrams, mostly from port 53 (DNS), indicating a DNS query and response exchange. The packet details and bytes panes show the structure of the UDP segments.



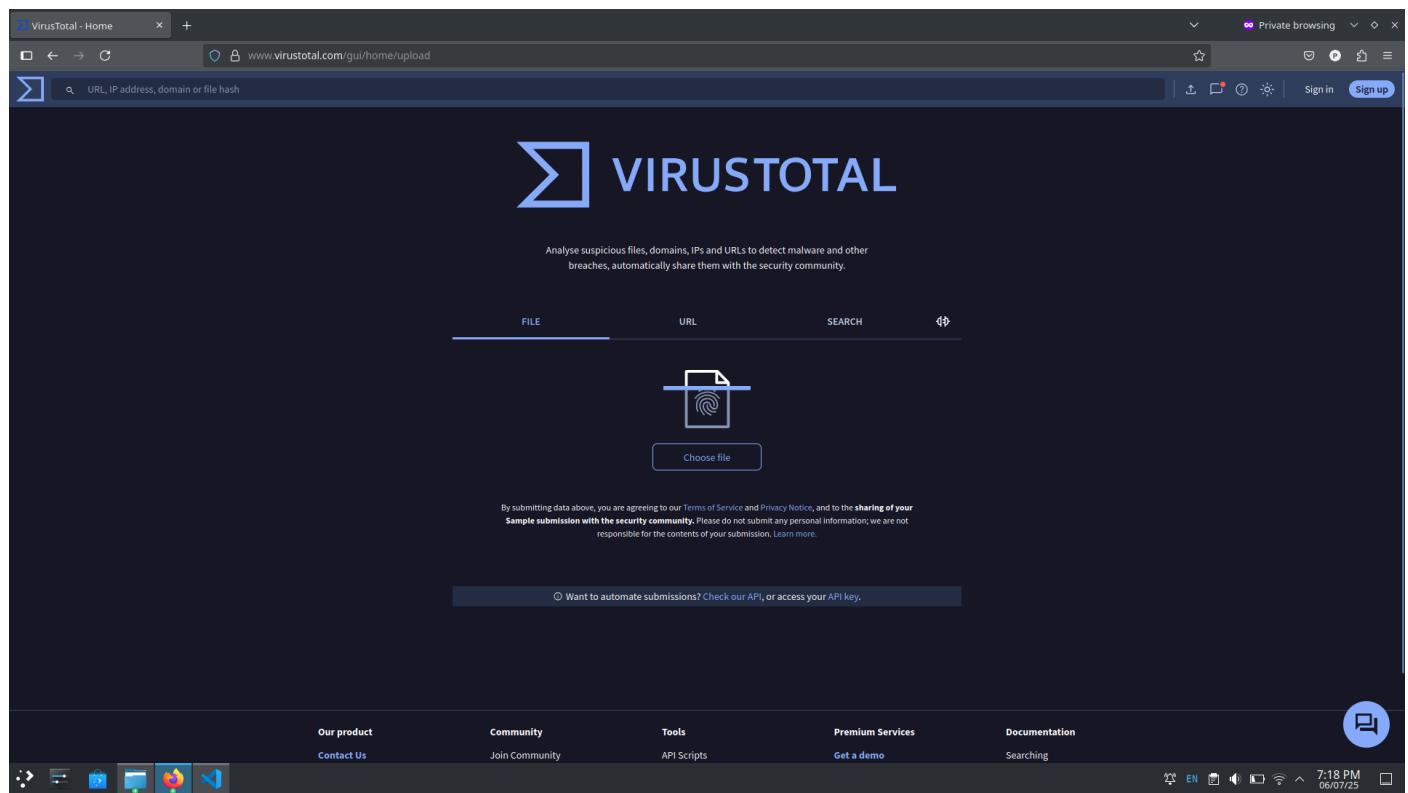
Step 4: Type “dns” in filters section and hit Enter to view dns packets.



## Practical – 7

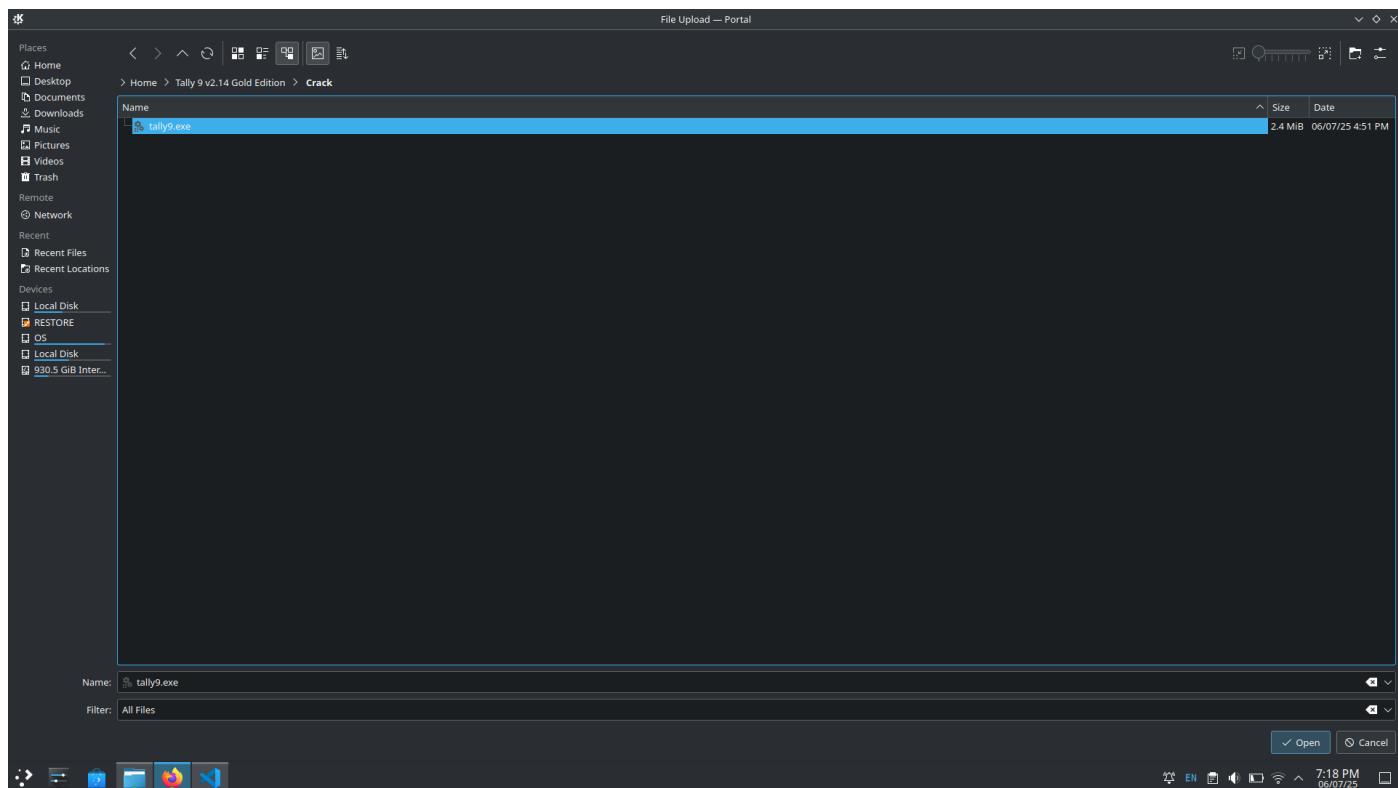
Aim: Identification of Virus infected file using virustotal.com

Step 1: Click “Choose file”.

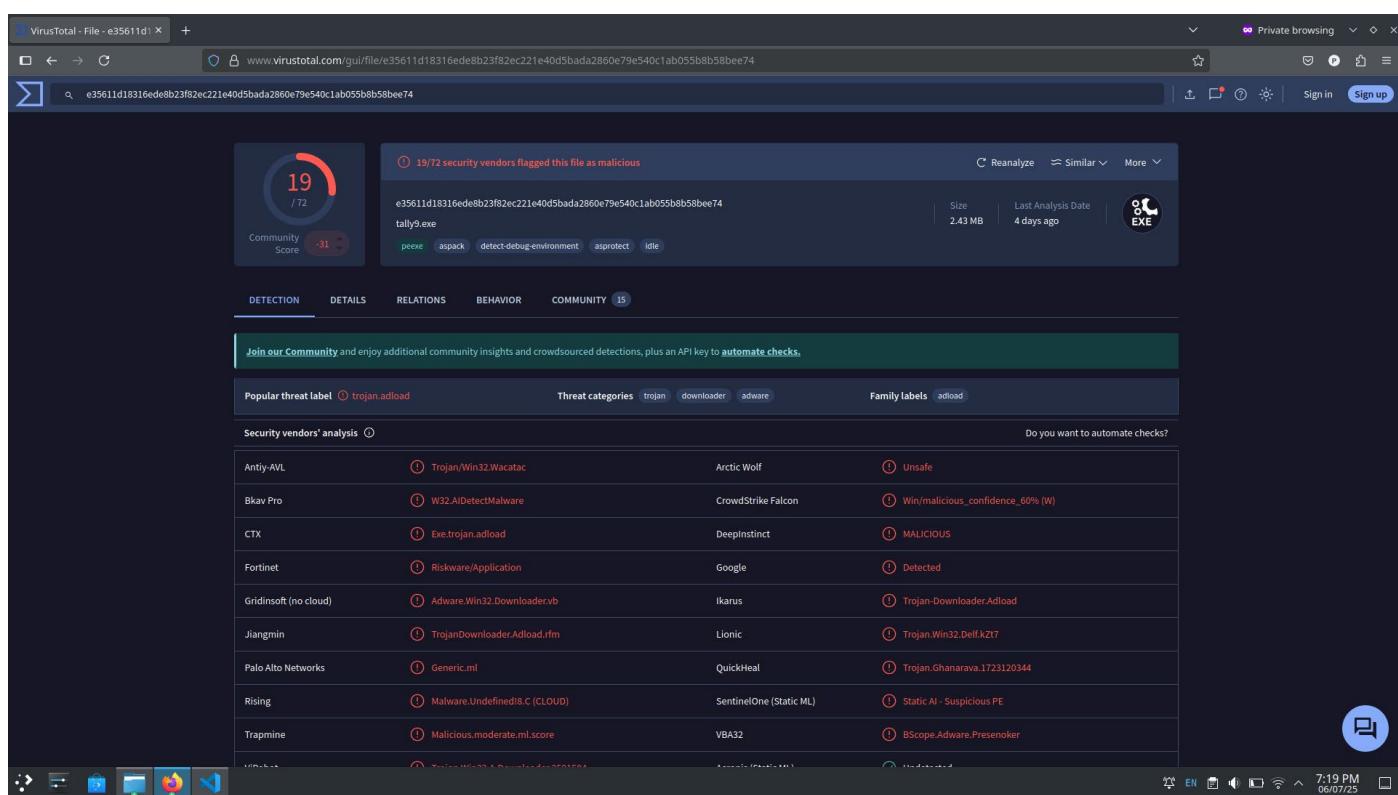


The screenshot shows the homepage of VirusTotal. At the top, there is a navigation bar with tabs for FILE, URL, and SEARCH. Below the tabs, there is a large input field with a paper icon and a "Choose file" button. A note below the input field states: "By submitting data above, you are agreeing to our [Terms of Service](#) and [Privacy Notice](#), and to the **sharing of your sample submission with the security community**. Please do not submit any personal information; we are not responsible for the contents of your submission. [Learn more](#)". At the bottom of the page, there is a footer with links for "Our product", "Community", "Tools", "Premium Services", and "Documentation". The footer also includes icons for social media and other services, as well as system status indicators like battery level and network connection.

Step 2: Select and open any file you want to scan.



Step 3: Review the detected threats.



A screenshot of a web browser displaying the VirusTotal analysis page for the file "tally9.exe". The URL is "www.virustotal.com/gui/file/e35611d18316ede8b23f82ec221e40d5bada2860e79e540c1ab055b8b58bee74". The main summary indicates that 19 out of 72 security vendors flagged the file as malicious. The file details show a community score of -31 and a size of 2.43 MB. The "DETECTION" tab is active, showing a table of vendor analysis results:

Vendor	Threat Category	Family Label
Anti-AVL	Trojan/Win32.Wacatac	Arctic Wolf
Bkav Pro	W32.AIDetectMalware	CrowdStrike Falcon
CTX	Exe.trojan.adload	DeepInstinct
Fortinet	Riskware/Application	Google
Gridinsoft (no cloud)	Adware.Win32Downloader.vb	Ikarus
Jiangmin	TrojanDownloader.Adload.rfm	Lionic
Palo Alto Networks	Generic.ml	QuickHeal
Rising	Malware.Undefined!B.C (CLOUD)	SentinelOne (Static ML)
Trapmine	Malicious.moderate.ml.score	VBA32



VirusTotal - File - e35611d18316ede8b23f82ec221e40d5bada2860e79e540c1ab055b8b58bee74

www.virustotal.com/gui/file/e35611d18316ede8b23f82ec221e40d5bada2860e79e540c1ab055b8b58bee74

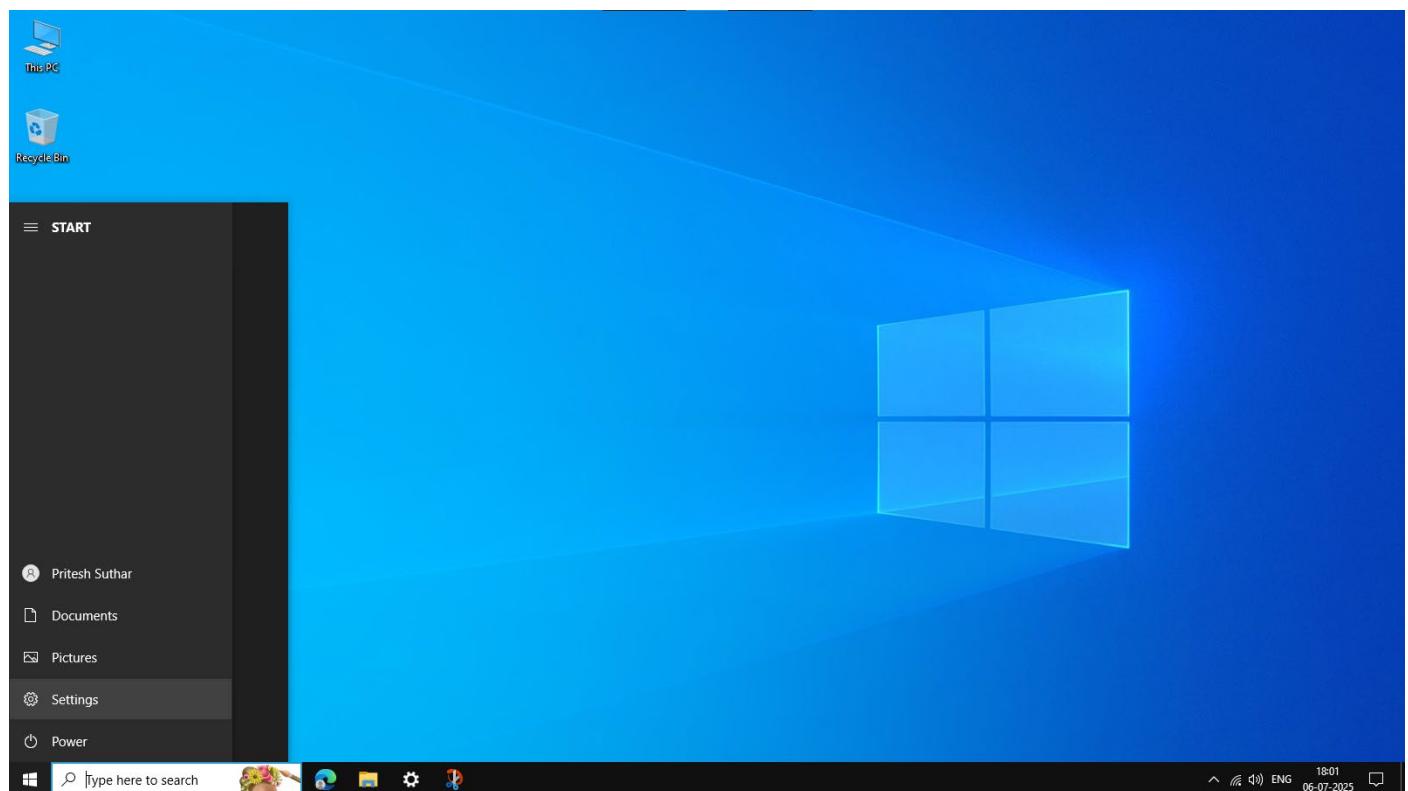
Scanner	Result	Definition
ViRobot	<span style="color: red;">!</span> Trojan.Win32.A.Downloader.2581504	Acronis (Static ML)
AhnLab-V3	<span style="color: green;">✓</span> Undetected	Alibaba
AliCloud	<span style="color: green;">✓</span> Undetected	ALYac
Arcabit	<span style="color: green;">✓</span> Undetected	Avast
AVG	<span style="color: green;">✓</span> Undetected	Avira (no cloud)
Baidu	<span style="color: green;">✓</span> Undetected	BitDefender
ClamAV	<span style="color: green;">✓</span> Undetected	CMC
Cynet	<span style="color: green;">✓</span> Undetected	DrWeb
Elastic	<span style="color: green;">✓</span> Undetected	Emsisoft
eScan	<span style="color: green;">✓</span> Undetected	ESET-NOD32
GData	<span style="color: green;">✓</span> Undetected	Huorong
K7AntiVirus	<span style="color: green;">✓</span> Undetected	K7GW
Kaspersky	<span style="color: green;">✓</span> Undetected	Kingssoft
Malwarebytes	<span style="color: green;">✓</span> Undetected	MaxSecure
McAfee Scanner	<span style="color: green;">✓</span> Undetected	Microsoft
NANO-Antivirus	<span style="color: green;">✓</span> Undetected	Panda
Sangfor Engine Zero	<span style="color: green;">✓</span> Undetected	SecureAge
Skyhigh (SWG)	<span style="color: green;">✓</span> Undetected	Sophos
SUPERAntiSpyware	<span style="color: green;">✓</span> Undetected	Symantec
TACHION	<span style="color: green;">✓</span> Undetected	TEHTRIS

## Practical – 8

Aim: Keep operating systems and software up to date by applying patches and updates.

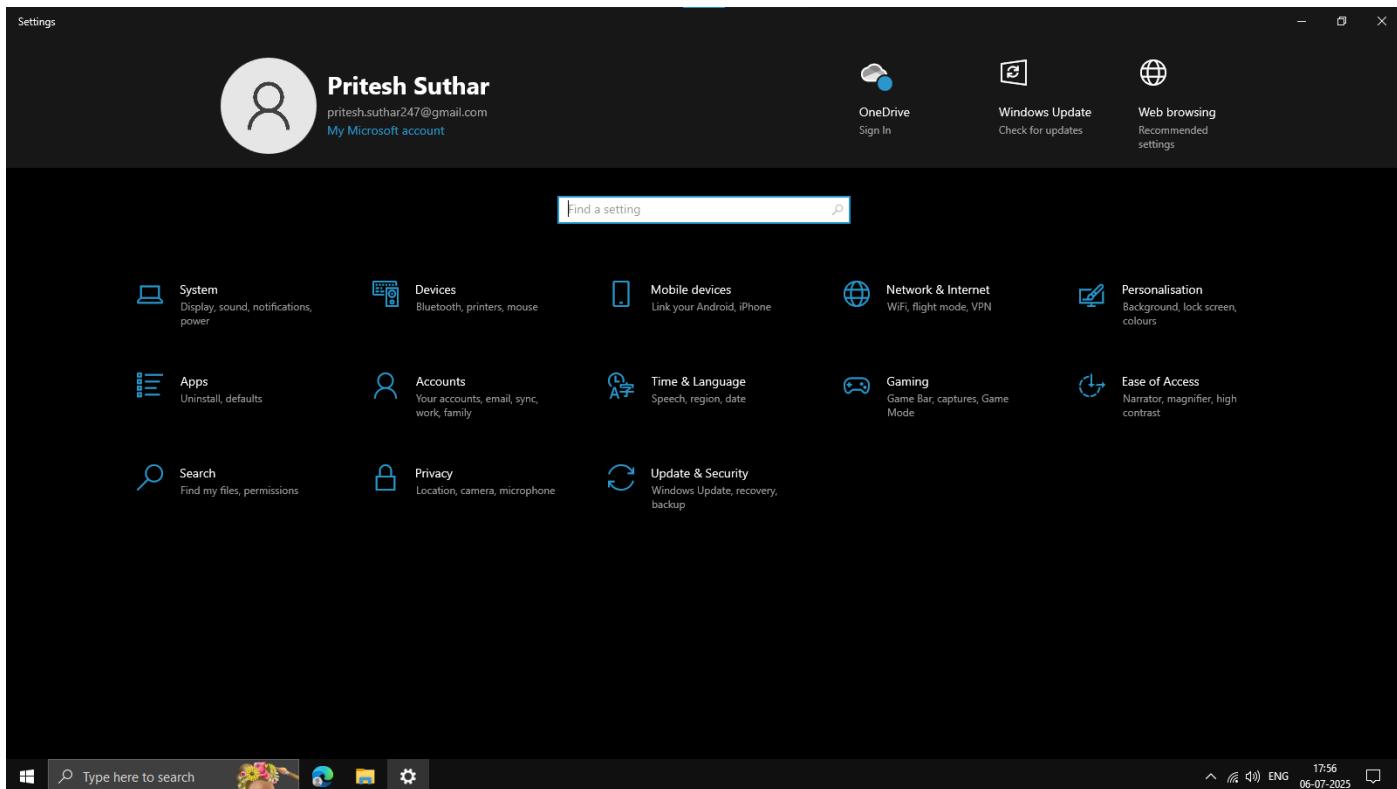
Windows 10:

Step 1: Press the windows icon and click on “Settings”.

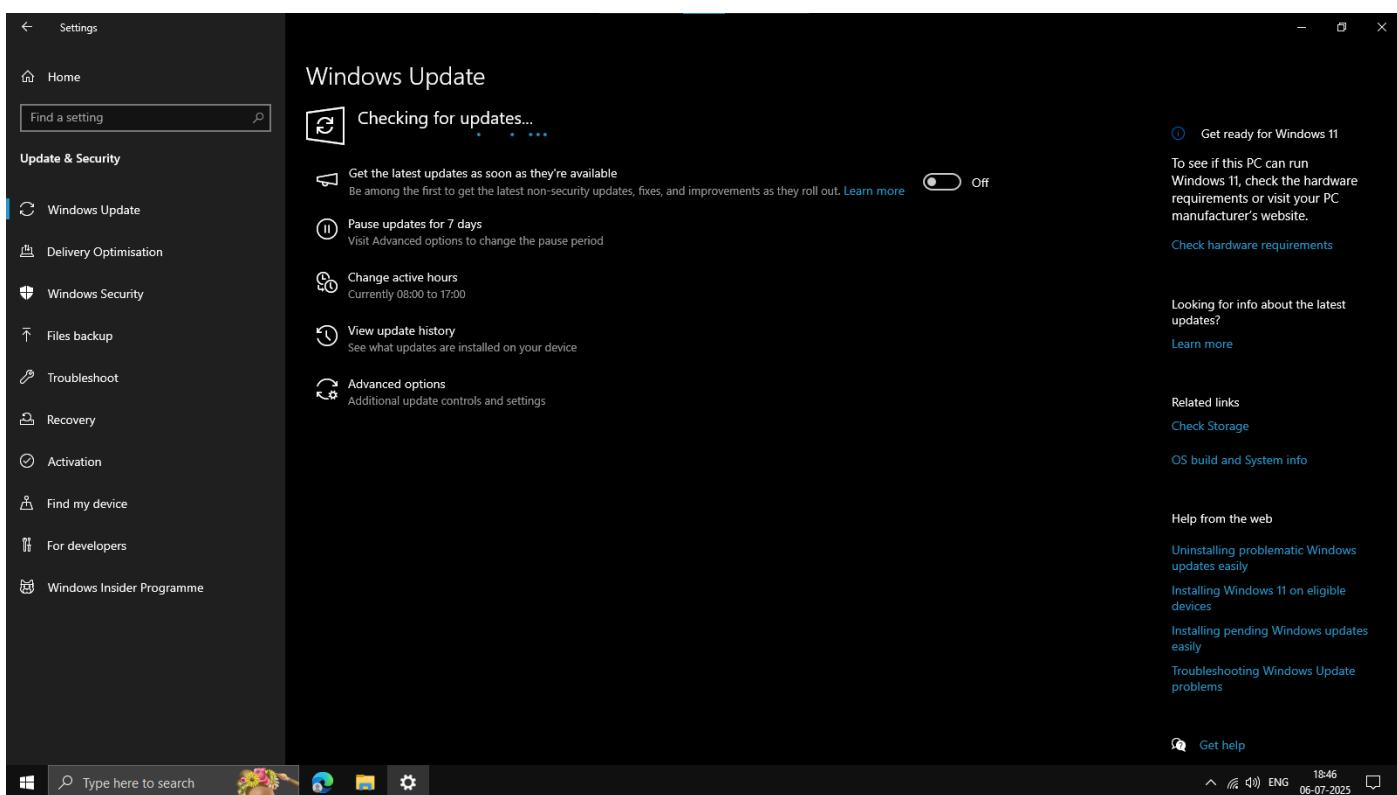




Step 2: Click on “Updates & Security”.

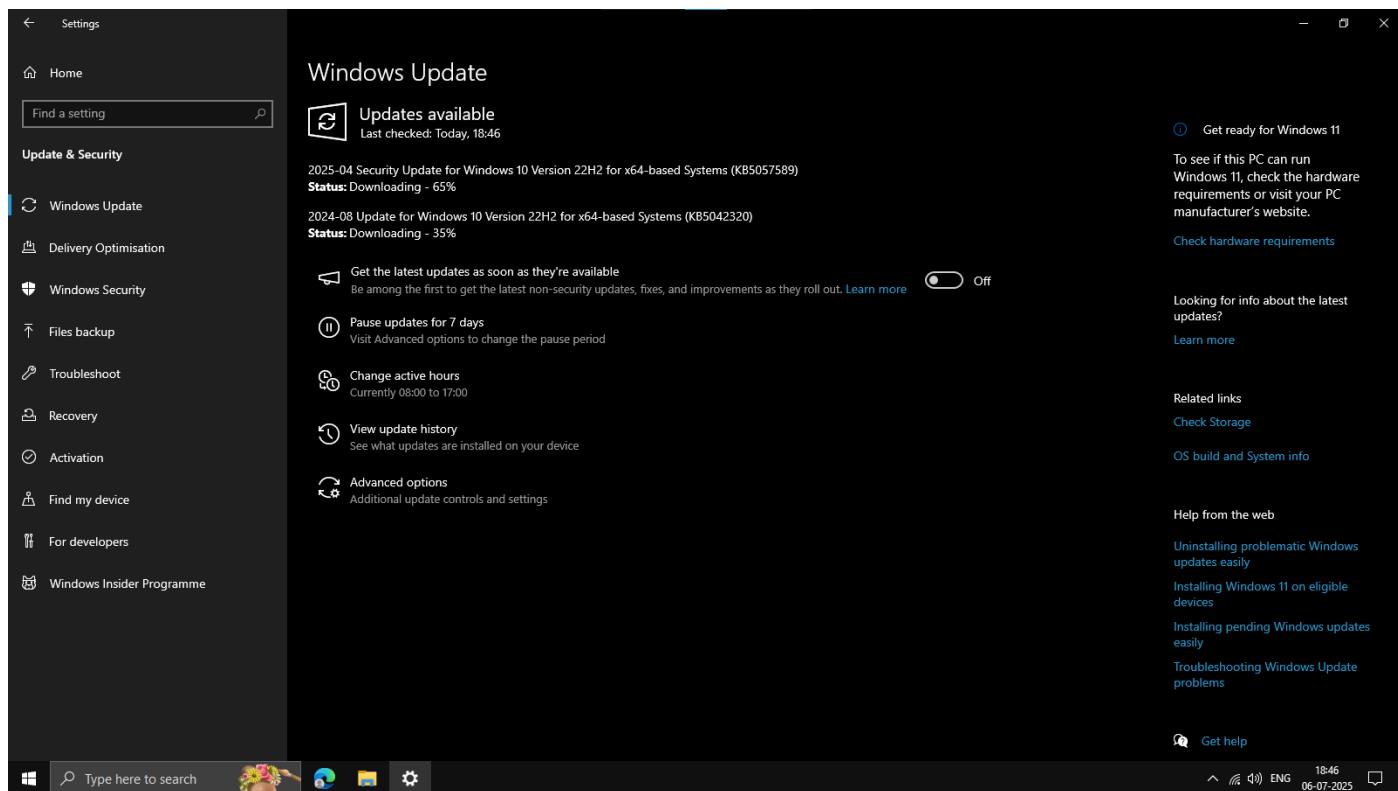


Step 3: Wait for the system to check for the updates.

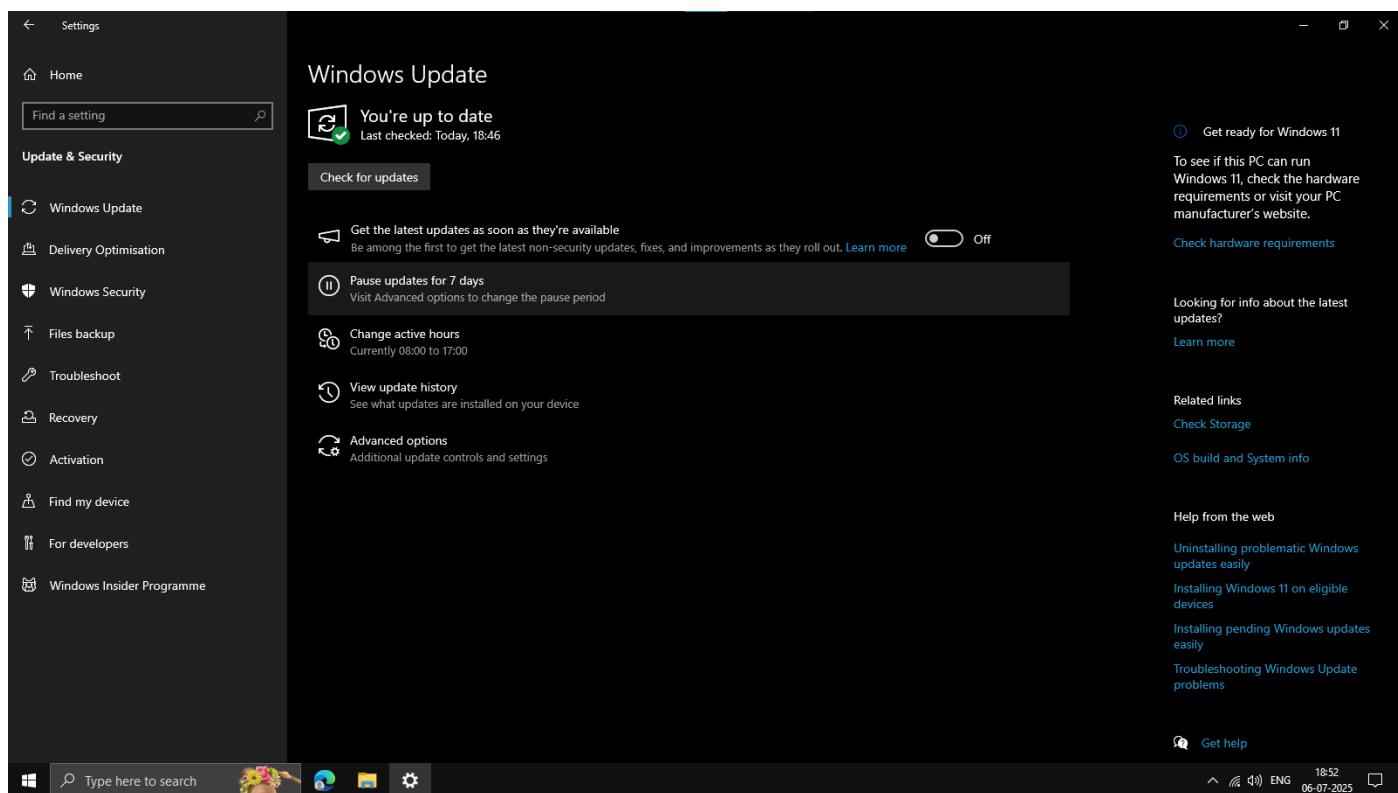




Step 4: Wait for the updates to get downloaded and install.

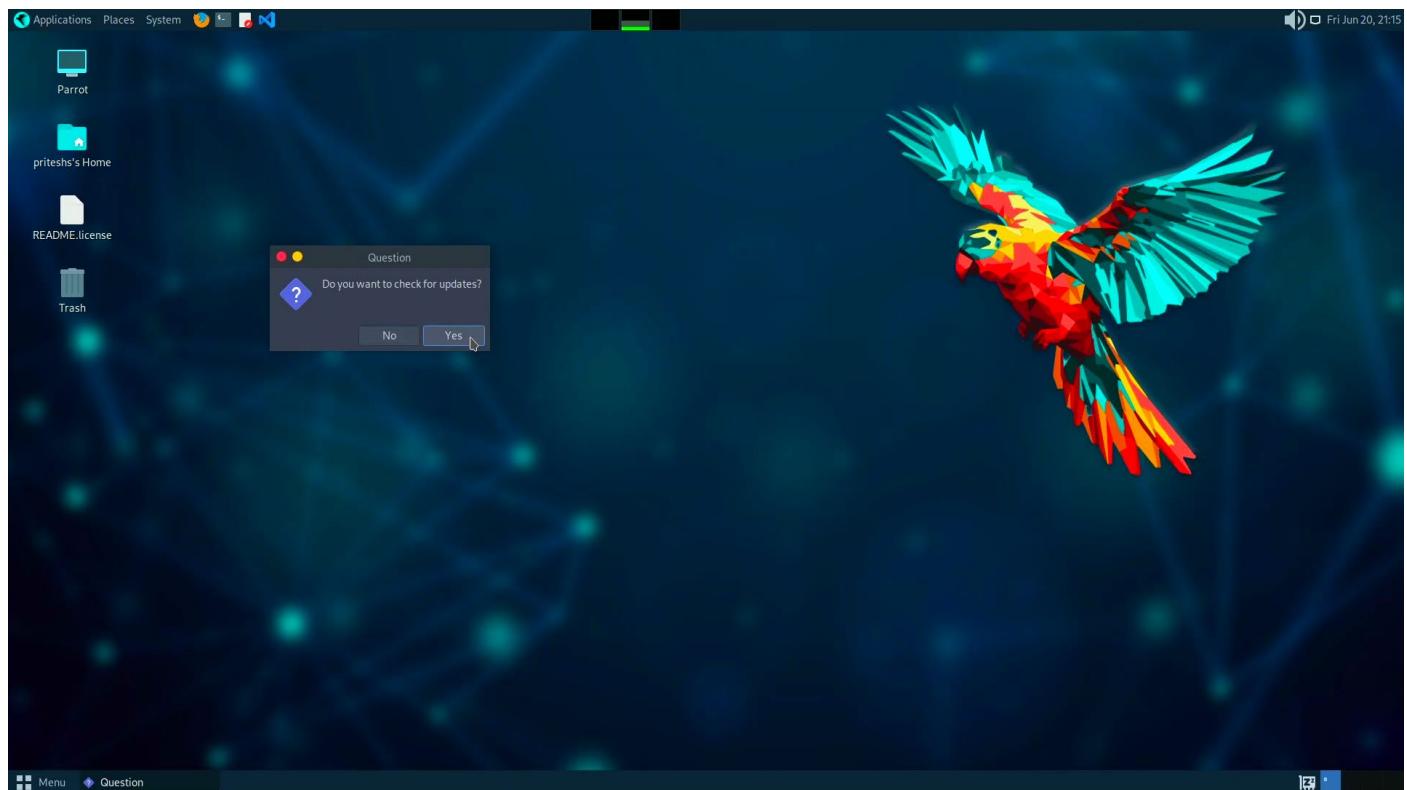


Step 5: Click on “Check for updates” to verify there are no more pending updates.

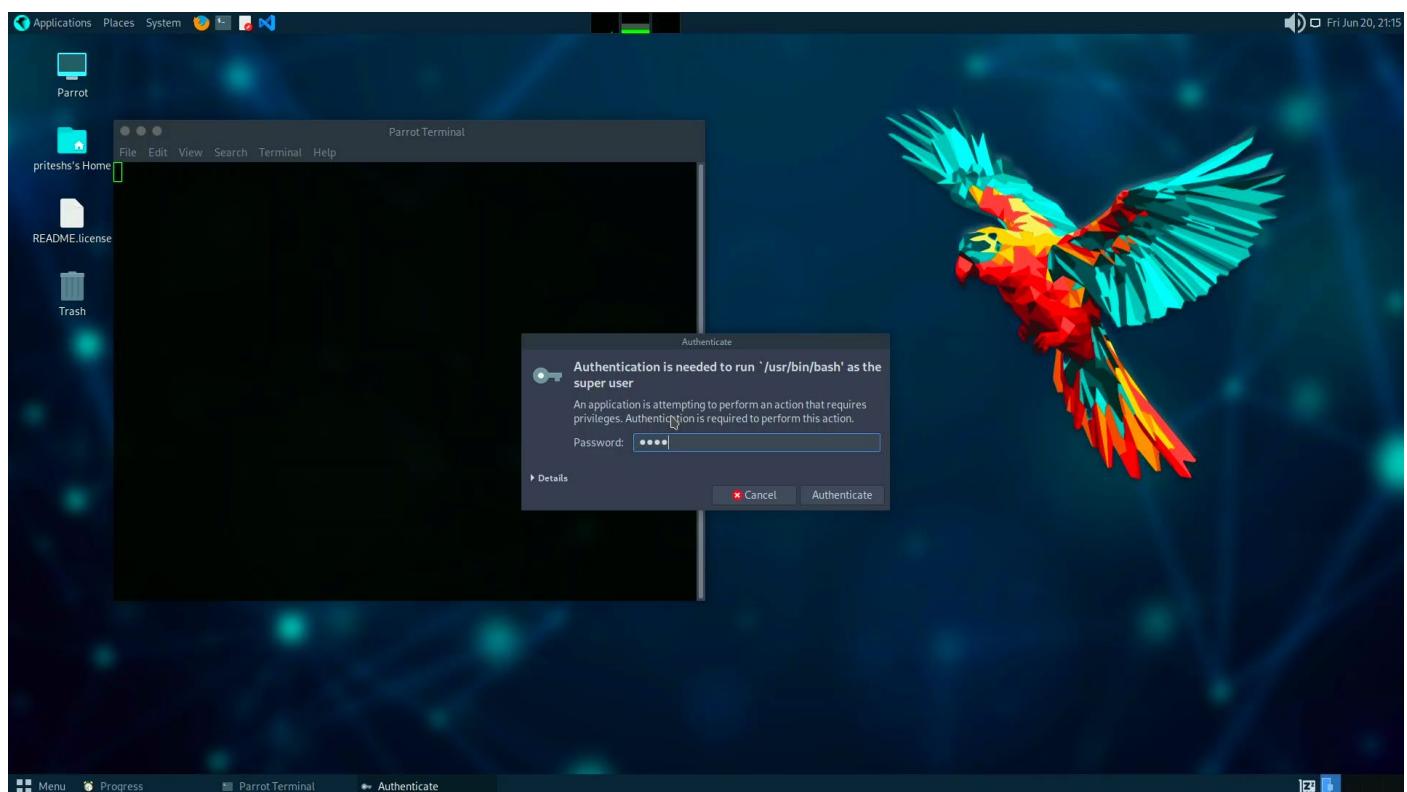


Parrot OS:

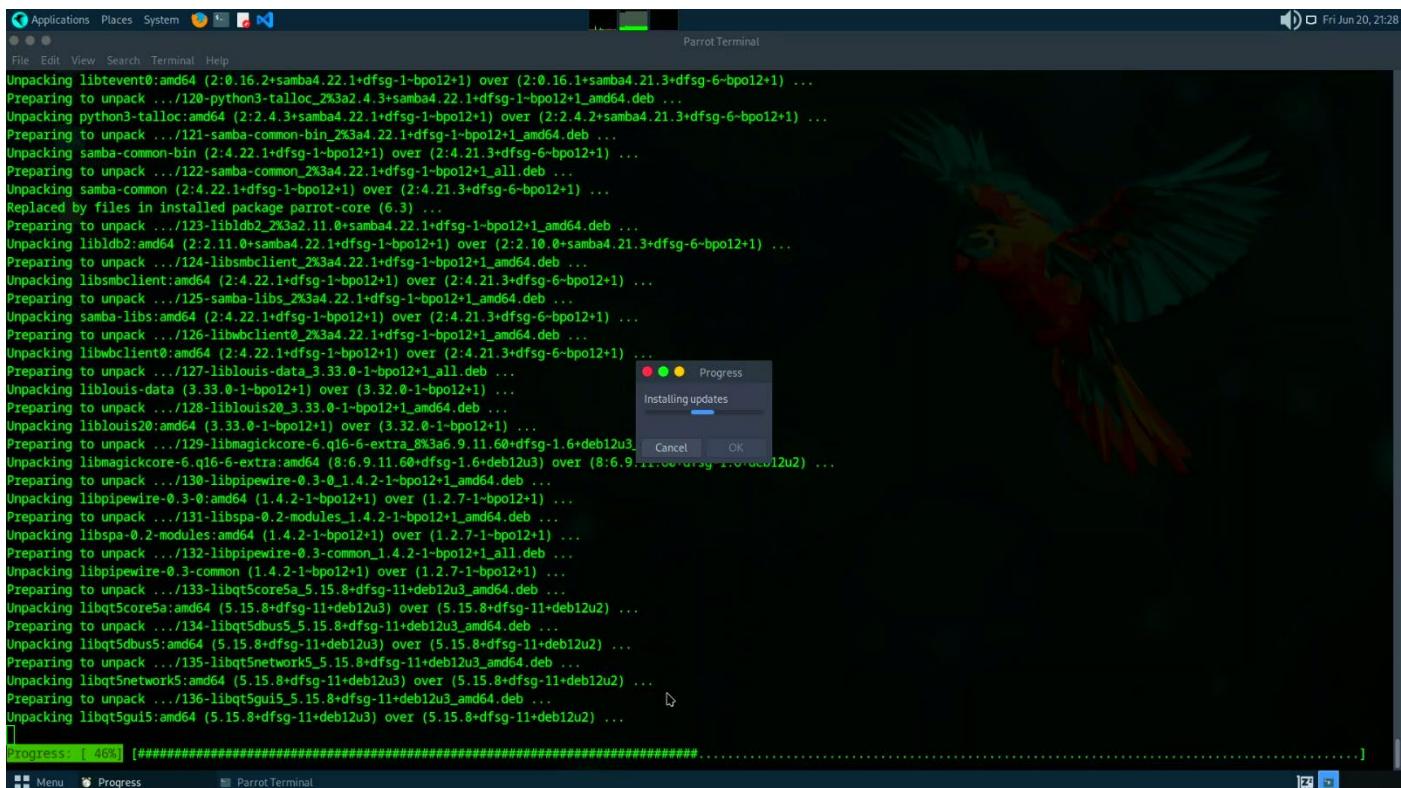
Step 1: Click on “Yes”.



Step 2: Enter your password and hit Enter.

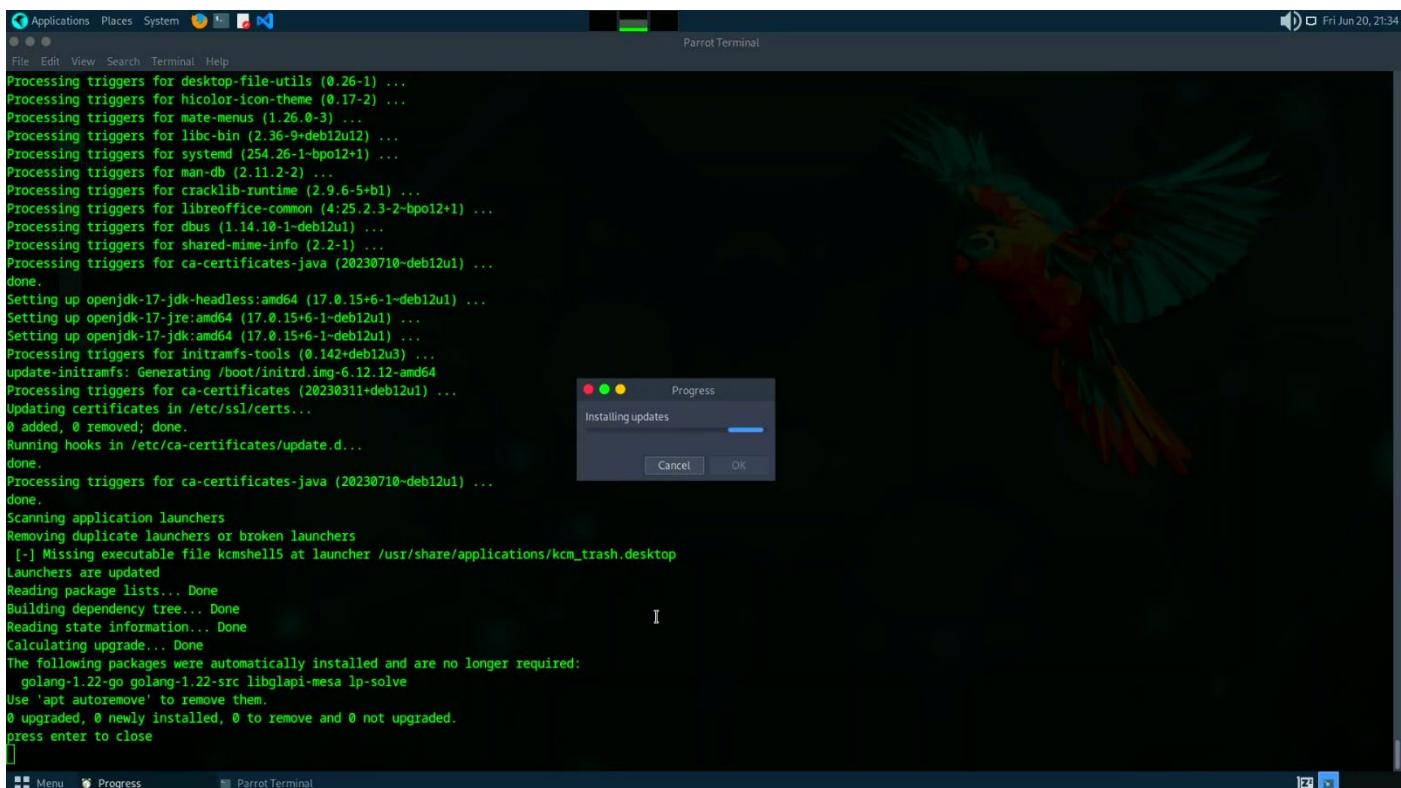


Step 3: Wait for the updates to get downloaded and installed.



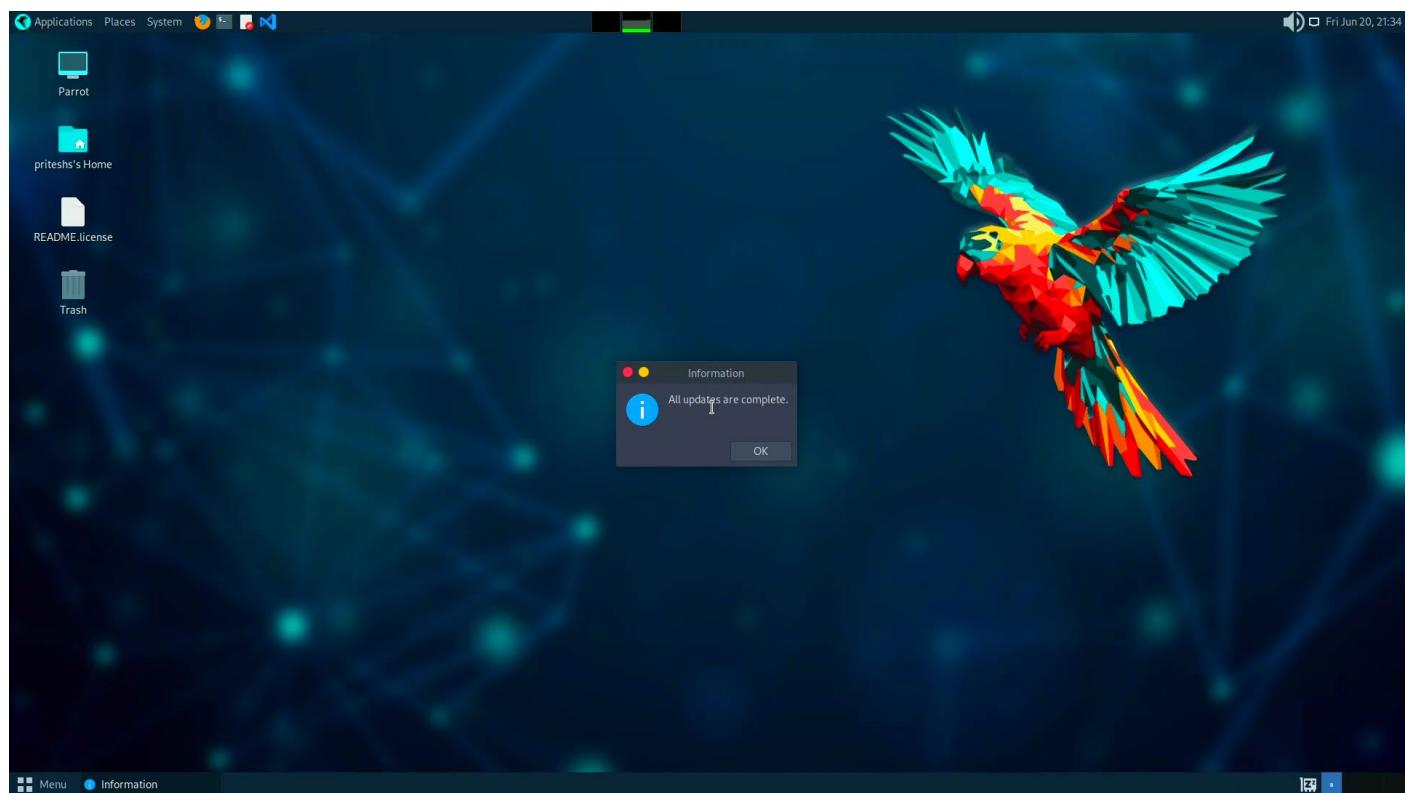
```
File Edit View Search Terminal Help
Unpacking libtevent0:amd64 (2:0.16.2+ubuntu4.22.1+dfsg-1-bpo12+1) over (2:0.16.1+samba4.21.3+dfsg-6-bpo12+1) ...
Preparing to unpack .../120-python3-talloc_2k3a2.4.3+samba4.22.1+dfsg-1-bpo12+_amd64.deb ...
Unpacking python3-talloc:amd64 (2:2.4.3+samba4.22.1+dfsg-1-bpo12+1) over (2:2.4.2+samba4.21.3+dfsg-6-bpo12+1) ...
Preparing to unpack .../121-samba-common-bin_2k3a4.22.1+dfsg-1-bpo12+_amd64.deb ...
Unpacking samba-common-bin (2:4.22.1+dfsg-1-bpo12+1) over (2:4.21.3+dfsg-6-bpo12+1) ...
Preparing to unpack .../122-samba-common_2k3a4.22.1+dfsg-1-bpo12+_all.deb ...
Unpacking samba-common (2:4.22.1+dfsg-1-bpo12+1) over (2:4.21.3+dfsg-6-bpo12+1) ...
Replaced by files in installed package parrot-core (6.3) ...
Preparing to unpack .../123-libldb2_2k3a2.11.0+samba4.22.1+dfsg-1-bpo12+_amd64.deb ...
Unpacking libldb2:amd64 (2:2.11.0+samba4.22.1+dfsg-1-bpo12+1) over (2:2.10.0+samba4.21.3+dfsg-6-bpo12+1) ...
Preparing to unpack .../124-libnmbclient_2k3a4.22.1+dfsg-1-bpo12+_amd64.deb ...
Unpacking libnmbclient:amd64 (2:4.22.1+dfsg-1-bpo12+1) over (2:4.21.3+dfsg-6-bpo12+1) ...
Preparing to unpack .../125-samba-libs_2k3a4.22.1+dfsg-1-bpo12+_amd64.deb ...
Unpacking samba-libs:amd64 (2:4.22.1+dfsg-1-bpo12+1) over (2:4.21.3+dfsg-6-bpo12+1) ...
Preparing to unpack .../126-libnmbclient0_2k3a4.22.1+dfsg-1-bpo12+_amd64.deb ...
Unpacking libnmbclient0:amd64 (2:4.22.1+dfsg-1-bpo12+1) over (2:4.21.3+dfsg-6-bpo12+1) ...
Preparing to unpack .../127-liblouis-data_3.33.0-1-bpo12+_all.deb ...
Unpacking liblouis-data (3.33.0-1-bpo12+1) over (3.32.0-1-bpo12+1) ...
Preparing to unpack .../128-liblouis0_3.33.0-1-bpo12+_amd64.deb ...
Unpacking liblouis0:amd64 (3.33.0-1-bpo12+1) over (3.32.0-1-bpo12+1) ...
Preparing to unpack .../129-libmagickcore-6.q16-6-extra_8k3a6.9.11.60+dfsg-1.6+deb12u3 ...
Unpacking libmagickcore-6.q16-6-extra:amd64 (8.6.9.11.60+dfsg-1.6+deb12u3) over (8.6.9.11.60+dfsg-1.0+deb12u2) ...
Preparing to unpack .../130-libpipedev-0.3-0.1.4.2-1-bpo12+_amd64.deb ...
Unpacking libpipedev-0.3-0:amd64 (1.4.2-1-bpo12+1) over (1.2.7-1-bpo12+1) ...
Preparing to unpack .../131-libspa-0.2-modules_1.4.2-1-bpo12+_amd64.deb ...
Unpacking libspa-0.2-modules:amd64 (1.4.2-1-bpo12+1) over (1.2.7-1-bpo12+1) ...
Preparing to unpack .../132-libpipedev-0.3-common_1.4.2-1-bpo12+_all.deb ...
Unpacking libpipedev-0.3-common (1.4.2-1-bpo12+1) over (1.2.7-1-bpo12+1) ...
Preparing to unpack .../133-libqt5core5a_5.15.8+dfsg-11+deb12u3 ...
Unpacking libqt5core5a:amd64 (5.15.8+dfsg-11+deb12u3) over (5.15.8+dfsg-11+deb12u2) ...
Preparing to unpack .../134-libqt5dbus5_5.15.8+dfsg-11+deb12u3 ...
Unpacking libqt5dbus5:amd64 (5.15.8+dfsg-11+deb12u3) over (5.15.8+dfsg-11+deb12u2) ...
Preparing to unpack .../135-libqt5network5_5.15.8+dfsg-11+deb12u3 ...
Unpacking libqt5network5:amd64 (5.15.8+dfsg-11+deb12u3) over (5.15.8+dfsg-11+deb12u2) ...
Preparing to unpack .../136-libqt5gui5_5.15.8+dfsg-11+deb12u3 ...
Unpacking libqt5gui5:amd64 (5.15.8+dfsg-11+deb12u3) over (5.15.8+dfsg-11+deb12u2) ...
Progress: [ 46% ] #####
[ Menu Progress Parrot Terminal ]
```

Step 4: Press enter to close the terminal.



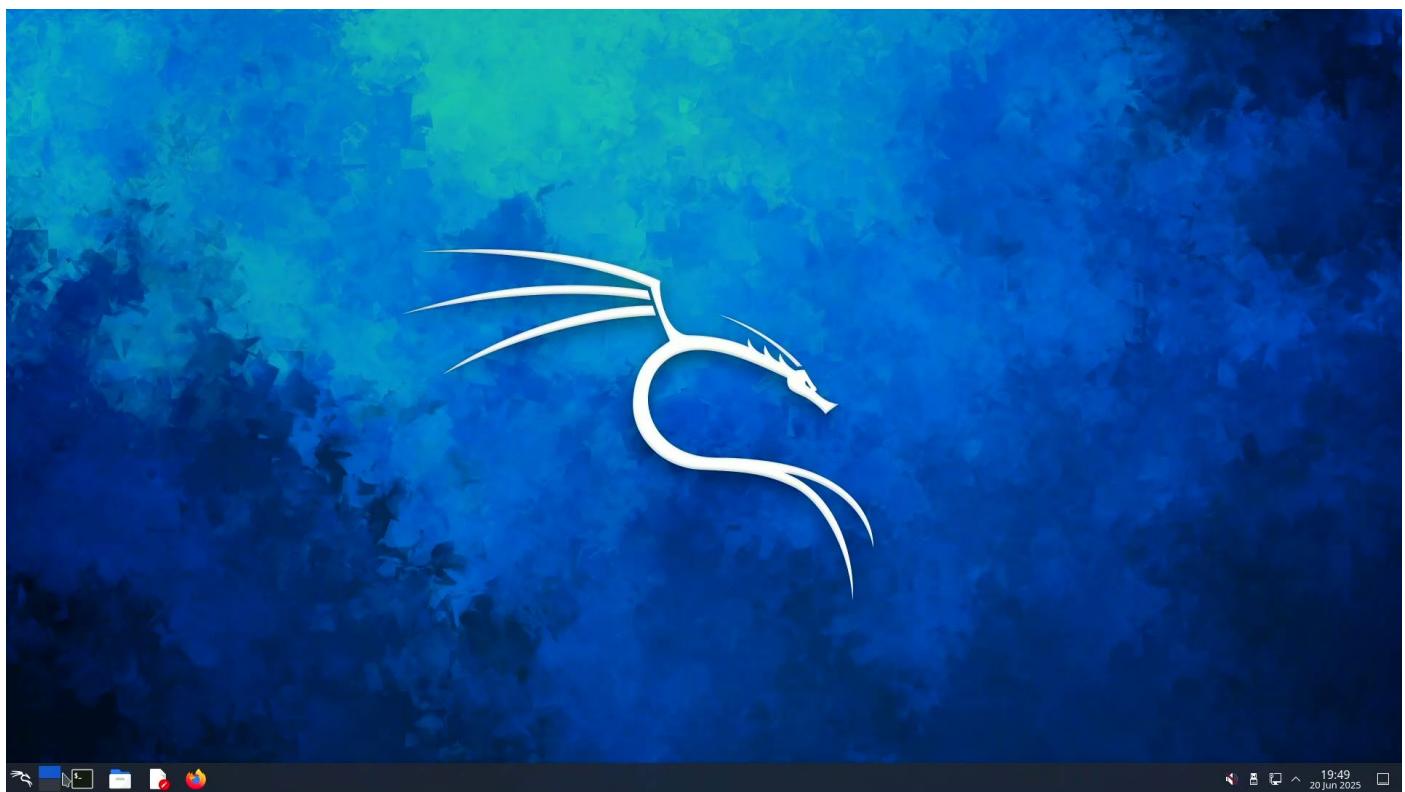
```
File Edit View Search Terminal Help
Processing triggers for desktop-file-utils (0.26-1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for mate-menus (1.26.0-3) ...
Processing triggers for libc-bin (2.36.9+deb12u12) ...
Processing triggers for systemd (254.26.1-bpo12+1) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for cracklib-runtime (2.9.6-5+b1) ...
Processing triggers for libreoffice-common (4:25.2.3-2-bpo12+1) ...
Processing triggers for dbus (1.14.10-1-deb12u1) ...
Processing triggers for shared-mime-info (2.2-1) ...
Processing triggers for ca-certificates-java (20230710-deb12u1) ...
done.
Setting up openjdk-17-jdk-headless:amd64 (17.0.15+6-1-deb12u1) ...
Setting up openjdk-17-jre:amd64 (17.0.15+6-1-deb12u1) ...
Setting up openjdk-17-jdk:amd64 (17.0.15+6-1-deb12u1) ...
Processing triggers for initramfs-tools (0.142+deb12u3) ...
update-initramfs: Generating /boot/initrd.img-6.12.12-amd64
Processing triggers for ca-certificates (20230311+deb12u1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for ca-certificates-java (20230710-deb12u1) ...
done.
Scanning application launchers
Removing duplicate launchers or broken launchers
[-] Missing executable file kcmshell15 at launcher /usr/share/applications/kcm_trash.desktop
Launchers are updated
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  golang-1.22-go golang-1.22-src libglapi-mesa lp-solve
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
press enter to close
[ Menu Progress Parrot Terminal ]
```

Step 5: Click “OK”.

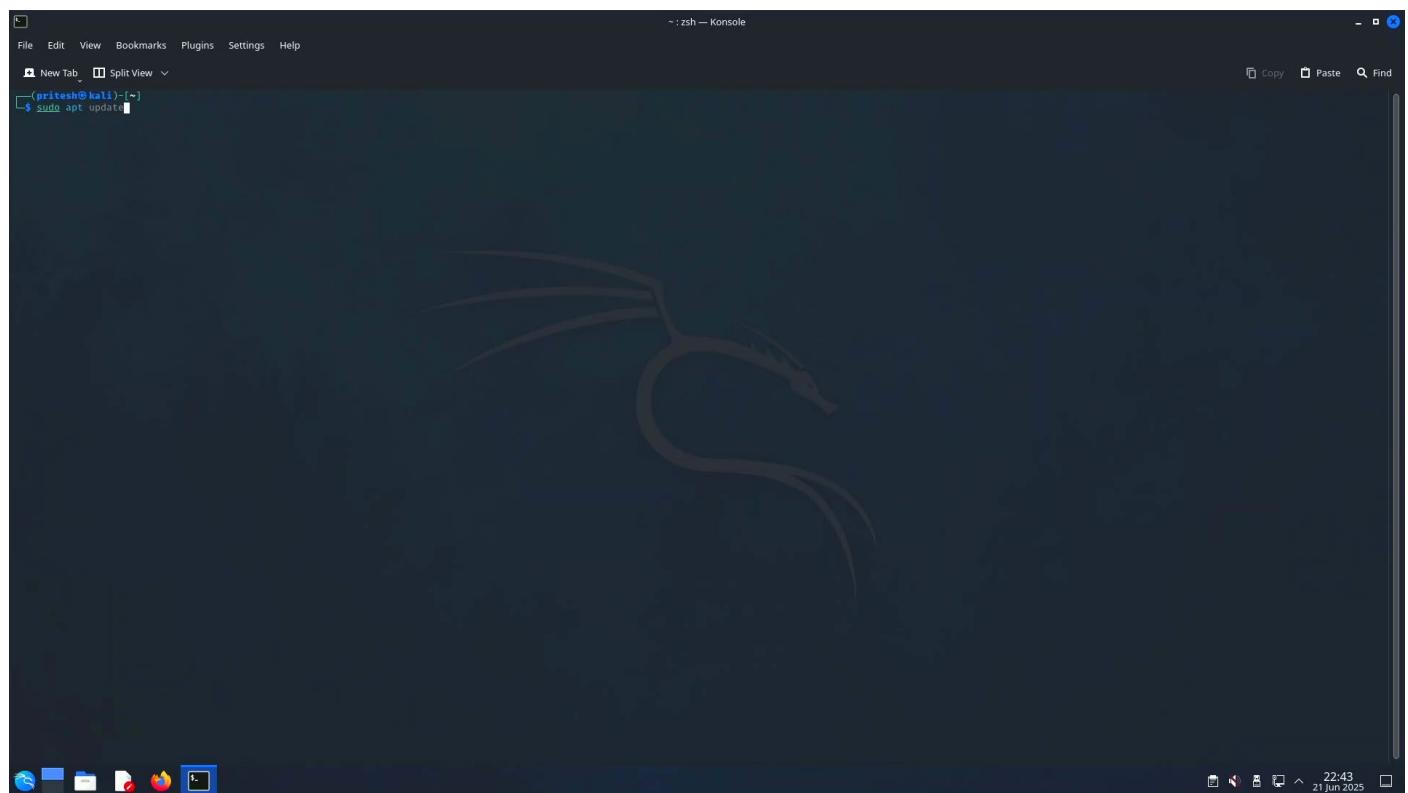


Kali Linux:

Step 1: Open Terminal by pressing “CTRL + ALT + T”.



Step 2: Type “sudo apt update” and hit enter.





Step 3: Enter your password.

```
File Edit View Bookmarks Plugins Settings Help
New Tab Split View ~
(pritesh㉿kali):~$ sudo apt update
[sudo] password for pritesh:
```

Step 4: Type “sudo apt upgrade” and hit Enter.

```
File Edit View Bookmarks Plugins Settings Help
New Tab Split View ~
(pritesh㉿kali):~$ sudo apt update
Hit:1 http://http.kali.org/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1371 packages can be upgraded. Run 'apt list --upgradable' to see them.
(pritesh㉿kali):~$ sudo apt upgrade
```



Step 5: Press “y” and hit enter.

Step 6: Wait for the updates to get downloaded and installed.

```
- : sudo apt — Konsole

File Edit View Bookmarks Plugins Settings Help
New Tab Split View Copy Paste Find

Unpacking libkf6screenpms8:amd64 (4:6.3.4-1) ...
Selecting previously unselected package liblayershellqtinterface6:amd64.
Preparing to unpack .../56-liblayershellqtinterface6_0.3.4-1_amd64.deb ...
Unpacking liblayershellqtinterface6:amd64 (6.3.4-1) ...
Preparing to unpack .../57-libkwin-common:amd64 (4:5.27.9-1) ...
Unpacking libkdeconfig5:amd64 (4:5.3.1) over (5.27.9-1) ...
dpkg: libkscreenclocker5:amd64 dependency problems, but removing anyway as you requested:
  plasma-workspace depends on libkscreenclocker5 (> 5.27.9-).
  kwin-common depends on libkscreenclocker5 (> 5.27.9-).

(Reading database ... 467364 files and directories currently installed.)
Removing libkscreenclocker5:amd64 (5.27.9-1) ...
Selecting previously unselected package libkscreenclocker6:amd64.
(Reading database ... 467357 files and directories currently installed.)
Preparing to unpack .../60-libkscreenclocker6_6.3.5-1_amd64.deb ...
Unpacking libkscreenclocker6:amd64 (6.3.5-1) ...
Preparing to unpack .../61-kwayland-data:amd64 ...
Unpacking kwayland-data (4:5.3.4-1) ...
Selecting previously unselected package libkwaylandclient6:amd64.
Preparing to unpack .../2-libkwaylandclient6_4x3a6.3.4-1_amd64.deb ...
Unpacking libkwaylandclient6:amd64 (4x3a6.3.4-1) ...
Preparing to unpack .../libglx:amd64 ...
Unpacking libglx:amd64 (6.8.2-3) ...
Preparing to unpack .../4-libwayland-server0_1.23.1-3_amd64.deb ...
Unpacking libwayland-server0:amd64 (1.23.1-3) over (1.22.0-2.1) ...
Selecting previously unselected package libkwin5.
Preparing to unpack .../5-libkwin5:amd64_4x3a6.3.5-1_amd64.deb ...
Unpacking libkwin5 (4:6.1.5-1) ...
Selecting previously unselected package libqaccessibilityclient-qt6:amd64.
Preparing to unpack .../6-libqaccessibilityclient-qt6-0_0.6.0-3_amd64.deb ...
Unpacking libqaccessibilityclient-qt6-0:amd64 (0.6.0-3) ...
Preparing to unpack .../7-kwin-x11:amd64 ...
Unpacking kwin-x11 (4:5.27.9-1) ...
Preparing to unpack .../8-kwin-wayland_4x3a6.3.5-1_amd64.deb ...
Unpacking kwin-wayland (4:6.3.5-1) over (4:5.27.9-1) ...
Preparing to unpack .../9-kde-config-gtk-style:amd64 (4:6.3.4-1) over (4:5.27.9-1) ...
dpkg: libdecorations2-sv5:amd64 dependency problems, but removing anyway as you requested:
  kwin-style-breeze depends on libdecorations2-sv5 (> 4:5.27.9-).
  kwin-common depends on libdecorations2-sv5 (> 4:5.27.9-).

(Reading database ... 467394 files and directories currently installed.)
Removing libdecorations2-sv5:amd64 (4:5.27.9-1) ...
Selecting previously unselected package libkwin-connections3-6:amd64.
(Reading database ... 467254 files and directories currently installed.)
Preparing to unpack .../00-libkwin-connections3-6_4x3a6.3.4-1_amd64.deb ...
Unpacking libkwin-connections3-6:amd64 (4:6.3.4-1) ...
Preparing to unpack .../01-kwin-style-breeze_4x3a6.3.5-1_amd64.deb ...
Unpacking kwin-style-breeze (4:6.3.5-1) over 4:5.27.9-1 ...
Preparing to unpack .../02-kde-decoration:amd64 (4:6.3.5-1) over (4:5.27.9-1) ...
Unpacking kde-decorator-portal-kde (6.3.5-1) over (5.27.9-1) ...
Preparing to unpack .../03-plasma-workspace-data_4x3a6.3.5-1_all.deb ...
Unpacking plasma-workspace-data (4:6.3.5-1) over (4:5.27.9-1) ...

Progress: [====] 9% [=====] 100%
```



Step 7: Type “sudo apt update” to verify there no new updates.

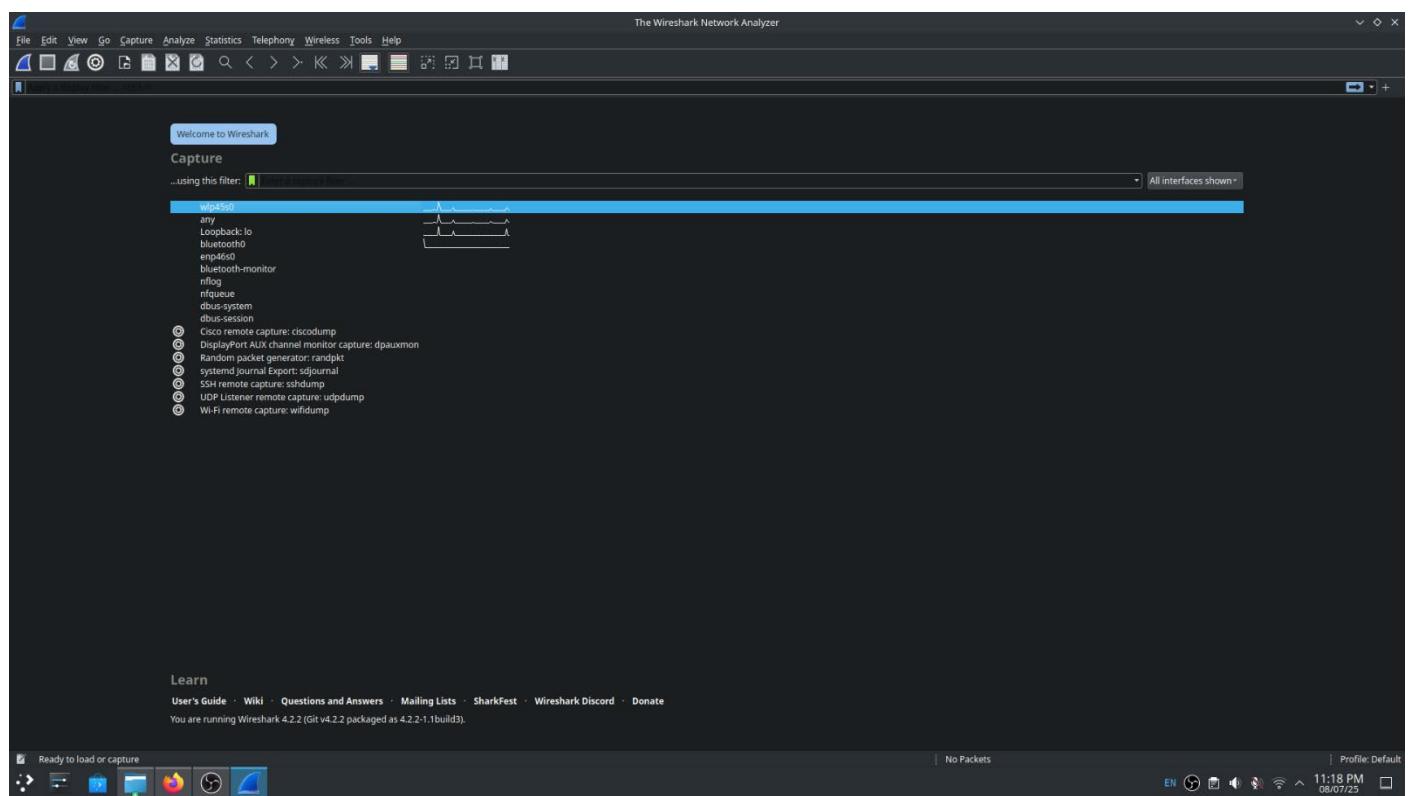
```
(pritesh㉿kali)-[~]
$ sudo apt update
Hit:1 http://http.kali.org/kali kali-rolling InRelease
All packages are up to date.

(pritesh㉿kali)-[~]
```

## Practical – 9

Aim: Monitor network traffic and detect suspicious or malicious activity

Step 1: Select and open “wlp4s0”.





## Step 2: Start network capturing.

The screenshot shows a NetworkMiner capture from interface wlp4s0. The timeline lists numerous TCP and UDP sessions between 192.168.0.1 and 192.168.0.202. Key interactions include:

- Session 328: 19475575 to 35.244.180.134 (HTTP traffic, ACK=144, Seq=112, Len=1045).
- Session 327: 195615372 to 35.244.180.134 (HTTP traffic, ACK=144, Seq=112, Len=42).
- Session 326: 195615372 to 35.244.180.134 (HTTP traffic, ACK=144, Seq=112, Len=42).
- Session 329: 206420554 to 192.168.0.202 (HTTP traffic, ACK=144, Seq=112, Len=42).
- Session 338: 297537384 to 35.244.180.134 (HTTP traffic, ACK=144, Seq=112, Len=42).
- Session 331: 988248381 to 192.168.0.202 (HTTP traffic, ACK=144, Seq=112, Len=42).
- Session 332: 988248381 to 172.217.194.84 (HTTP traffic, ACK=144, Seq=112, Len=42).
- Session 333: 8899655567 to 192.168.0.202 (TCP traffic, ACK=2727, Seq=2277, Win=62464, Len=9, TSval=3462935388, TSecr=1225959497).
- Session 334: 99.099914959 to 192.168.0.291 (TCP traffic, ACK=204, Seq=2277, Win=62464, Len=9, TSval=3462935388, TSecr=1225959497).
- Session 335: 99.099914959 to 192.168.0.291 (TCP traffic, ACK=204, Seq=2277, Win=62464, Len=9, TSval=3462935388, TSecr=1225959497).
- Session 336: 92.053396270 to 142.250.183.42 (HTTP traffic, ACK=144, Seq=112, Len=42).
- Session 337: 92.053396270 to 192.168.0.202 (HTTP traffic, ACK=144, Seq=112, Len=42).
- Session 338: 93.08652165 to 172.217.194.84 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 339: 95.027968675 to 192.168.0.201 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 340: 95.027968675 to 192.168.0.201 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 341: 98.612409678 to 192.168.0.291 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 342: 98.612409678 to 192.168.0.291 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 343: 99.94386133 to 192.168.0.201 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 344: 100.639749654 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 345: 100.639749654 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 346: 101.629174268 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 347: 101.629174268 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 348: 101.629174268 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 349: 101.629174268 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 350: 101.629174268 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 351: 101.992942701 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 352: 102.076184827 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 353: 102.076184827 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 354: 102.115489638 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 355: 103.647959795 to 192.168.0.292 (DNS traffic, ACK=256, OPT=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 356: 103.647959795 to 192.168.0.292 (DNS traffic, ACK=256, OPT=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 357: 103.647959795 to 192.168.0.292 (DNS traffic, ACK=256, OPT=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 358: 103.689379133 to 192.168.0.1 (DNS traffic, ACK=256, OPT=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 359: 103.689379133 to 192.168.0.1 (DNS traffic, ACK=256, OPT=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 360: 103.689379133 to 192.168.0.1 (DNS traffic, ACK=256, OPT=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 361: 103.689379133 to 192.168.0.1 (DNS traffic, ACK=256, OPT=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 362: 103.708947989 to 142.250.183.42 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 363: 103.708947989 to 142.250.183.42 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 364: 103.708947989 to 142.250.183.42 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 365: 103.708947989 to 142.250.183.42 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 366: 103.708947989 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 367: 103.708947989 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 368: 103.708947989 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 369: 103.708947989 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 370: 103.708947989 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 371: 103.708947989 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 372: 103.708947989 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 373: 104.245510959 to 142.251.42.238 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 374: 104.245510959 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 375: 104.245510959 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 376: 107.996959344 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 377: 107.997199856 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 378: 107.997199856 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 379: 107.997199856 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).
- Session 380: 107.997199856 to 192.168.0.292 (HTTP traffic, ACK=144, Seq=112, Len=45, TSval=3462935388, TSecr=1225959497).

Packets: 380 - Displayed: 380 (100.0%) | Profile: Default | EN | 11:20 PM | 08/07/25

## Step 3: Open "<http://testphp.vulnweb.com/login.php>" in browser.

The screenshot shows the Acunetix Web Vulnerability Scanner interface. A test request is being sent to the URL <http://testphp.vulnweb.com/login.php>. The payload injected is: `?username=1 OR 1=1`.

The response status is "Success" with a response time of 0.0000 ms. The response content shows a login form with fields for "Username" and "Password". Below the form, a warning message states: "Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more."

At the bottom, the browser toolbar shows various icons for navigation, search, and file operations.



#### Step 4: Enter Username and Password.



#### Step 5: In filters type “ http contains “uname” “ and hit Enter.



**DRS. KIRAN & PALLAVI PATEL GLOBAL UNIVERSITY**

Established Under Gujarat Private Universities (Amendment) Act, 2021 (Gujarat Act No. 15 of 2021)

KRISHNA SCHOOL OF EMERGING TECHNOLOGY & APPLIED RESEARCH (KSET)

**KPGU**  
Vadodara

## Practical – 10

Aim: Study AI related attacks on AI Devices.