

# SPEECH BASED WEB BROWSER

## **Group Members:**

**Monika Khandelwal 184101026**

**Oza Jay 184101027**

**Shavi Gupta 184101033**

**Harsh Mehta 184101051**

## **Introduction:**

The system is used to control web browser through human voice.

Following functionalities, we have implemented with speech:

- 1.Open Browser.
- 2.Opens Facebook.
- 3.Opens News.
- 4.Read News.
- 5.Stop Narration of news.
- 6.Open Cricket Score and read it.
- 7.Opens Browsing History.
- 8.Opens Bookmarks.
- 9.Scroll Up.
- 10.Scroll Down.

## **Model Used for system development:**

Hidden Markov Model:

Elements:

1. Number of states: 5
2. Number of Observation symbol (Size of codebook): 32
3. Bakers model is used.

## **Recording:**

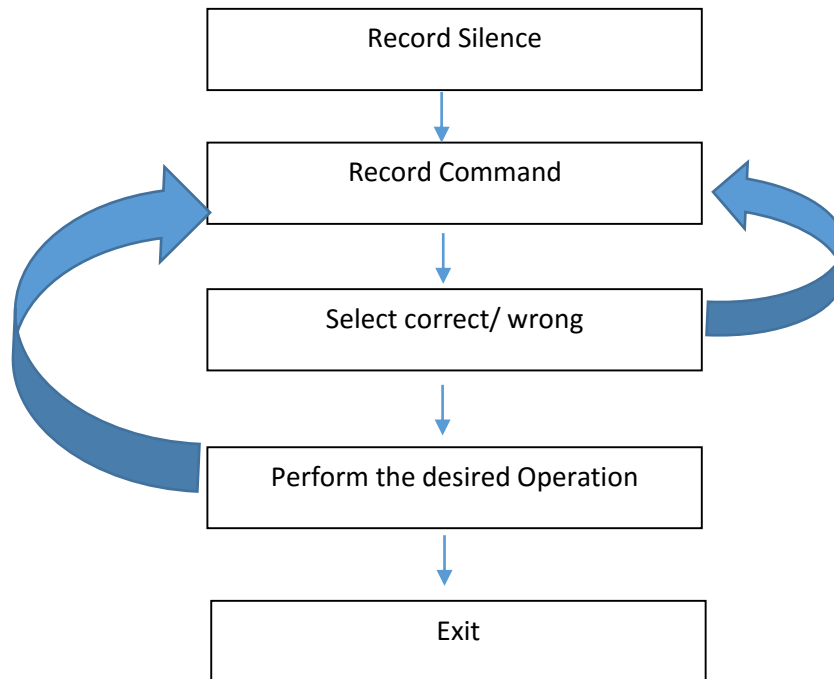
Following words are used to train and test the system. The recording is done using cool edit software. The sampling rate used was 16000. Each sample of size 16bit, mono channel.

- 1.Open.
- 2.News.
- 3.Read.
- 4.Stop.
- 5.Facebook.
- 6.Bookmark.
- 7.Cricket.
- 8.History.
- 9.Up
10. Down.
- 11.Close.

## **Codebook**

A 32\*12 codebook is used, generated using the cepstral coefficient which were generated from the pre-recorded utterance of different words. For generation of codebook, binary split method along with k-means clustering is used.

## High Level View of Model:



## Training Model:

The training was done with 30 utterances of each word. The data was pre-recorded with cool edit tool. The frame limit was set to 130 at max. The average of model was done for three iterations, taking model generated at previous iteration as base. The threshold value of 10 to the power of -30 (difference in  $P^*$  in consecutive iteration) was used to train the model for particular observation sequence.

## Steps for training:

1. Pre-recorded utterances of words are used to generate the observation sequences.
2. The obtained observation sequences are used to train the models for different colors.
3. The trained model are then used for testing purpose.

## **Live Training Module:**

For a new user we have included a module to train on user's voice.

To train first user need to reset all existing data, it can be reset by pressing 999.

After user will get options:

1.Record Word.

2.Train Model.

So first Record at least 10 words of each type and then train the model.

Use the generated model for testing on new user.

## **Testing the Model:**

On executing the program,

- 1) Silence File is recorded
- 2) The options are available:
  - a) Speak
  - b) Exit
- 3) On selecting option 1:

Record the command and the recorded command will be displayed.  
Enter c/w according to correct or wrong answer respectively.

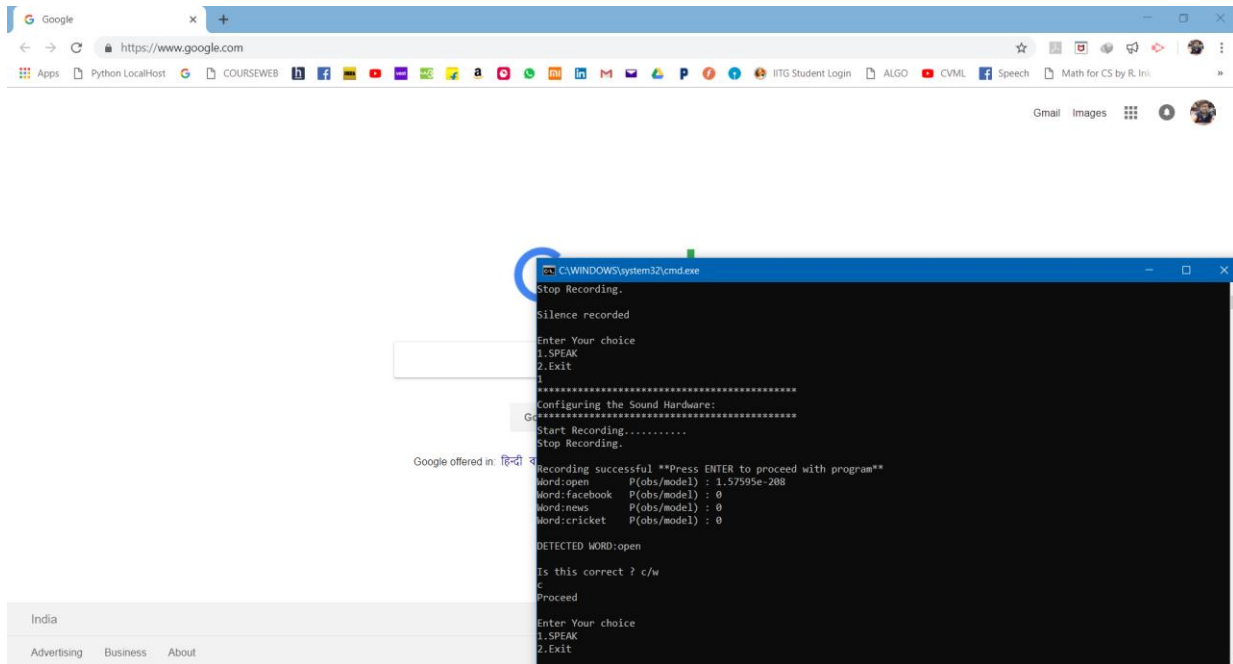
  - a) On selecting c it will perform the operation
  - b) If w is selected then it will ask to speak again.
- 4) On selecting option 2:

Simply exits the command line

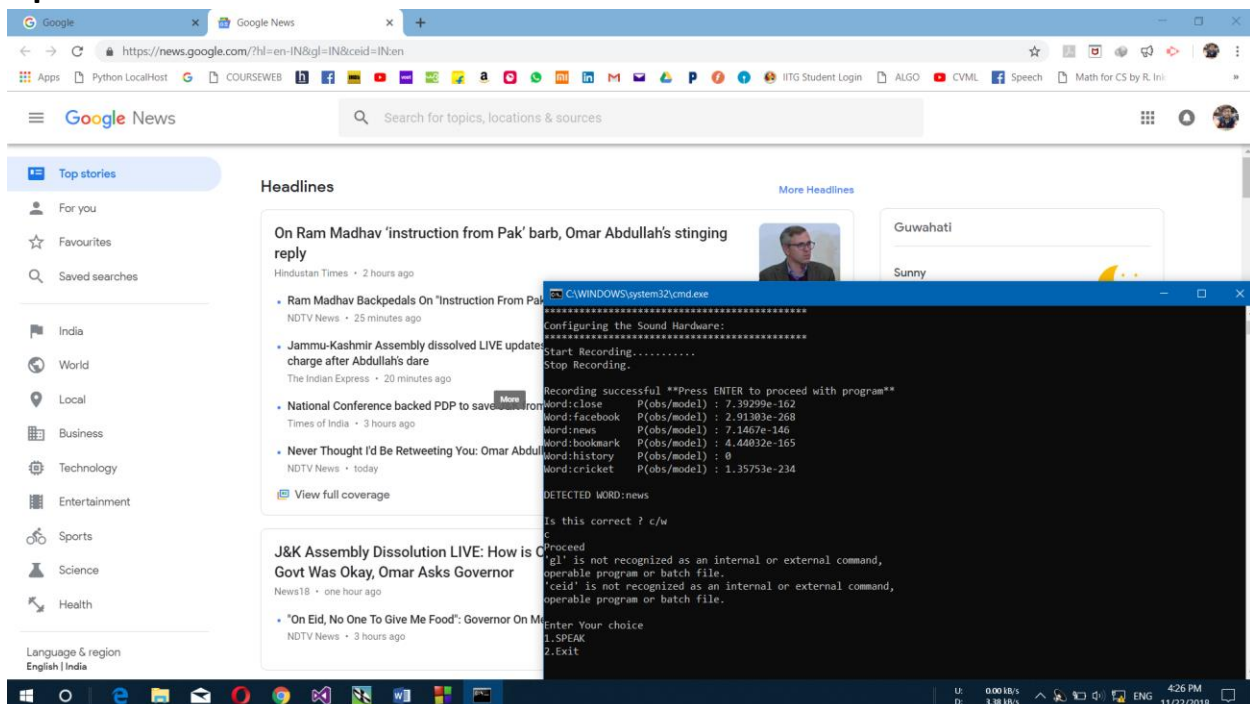
Output:

## TESTING MODULE SCREENSHOTS

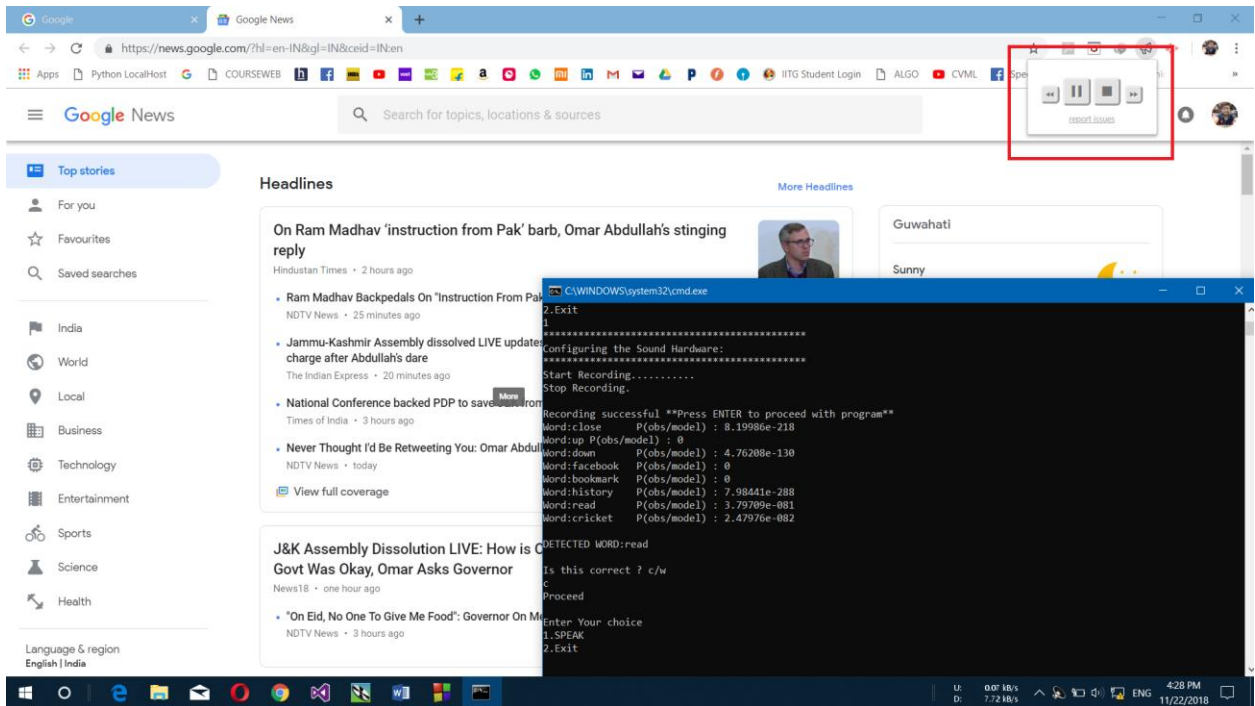
### 1) Open Browser



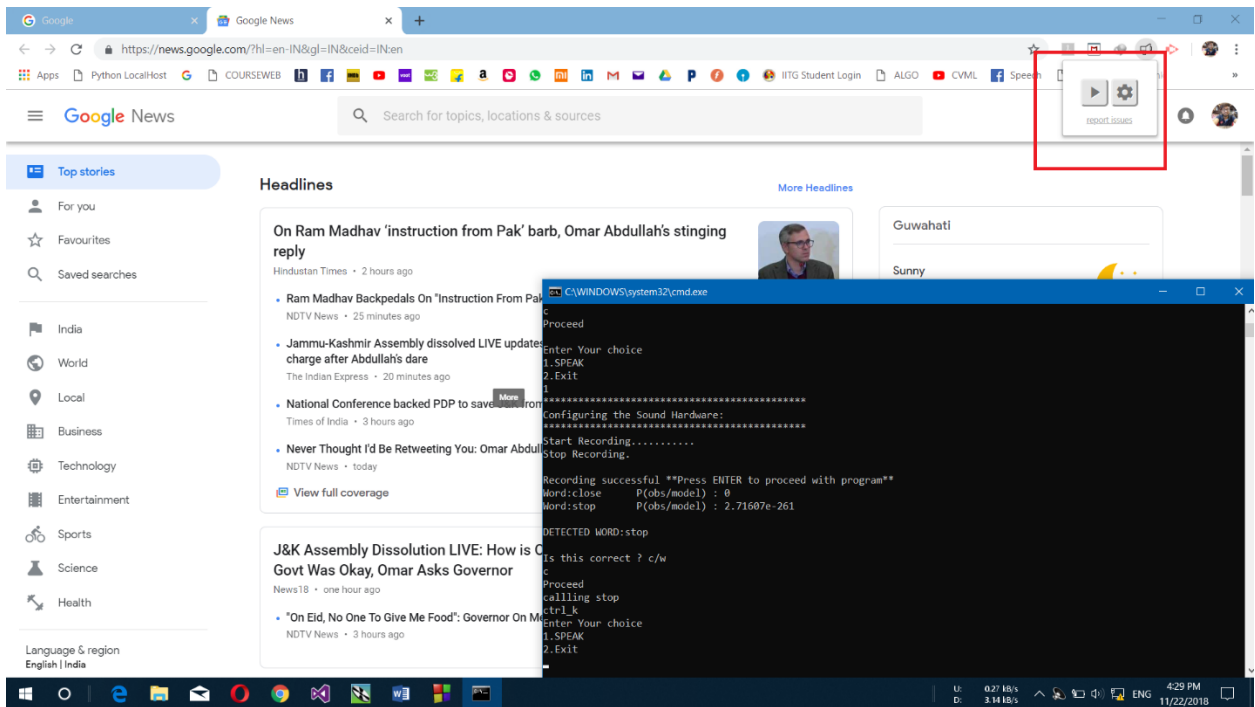
### 2) Open News



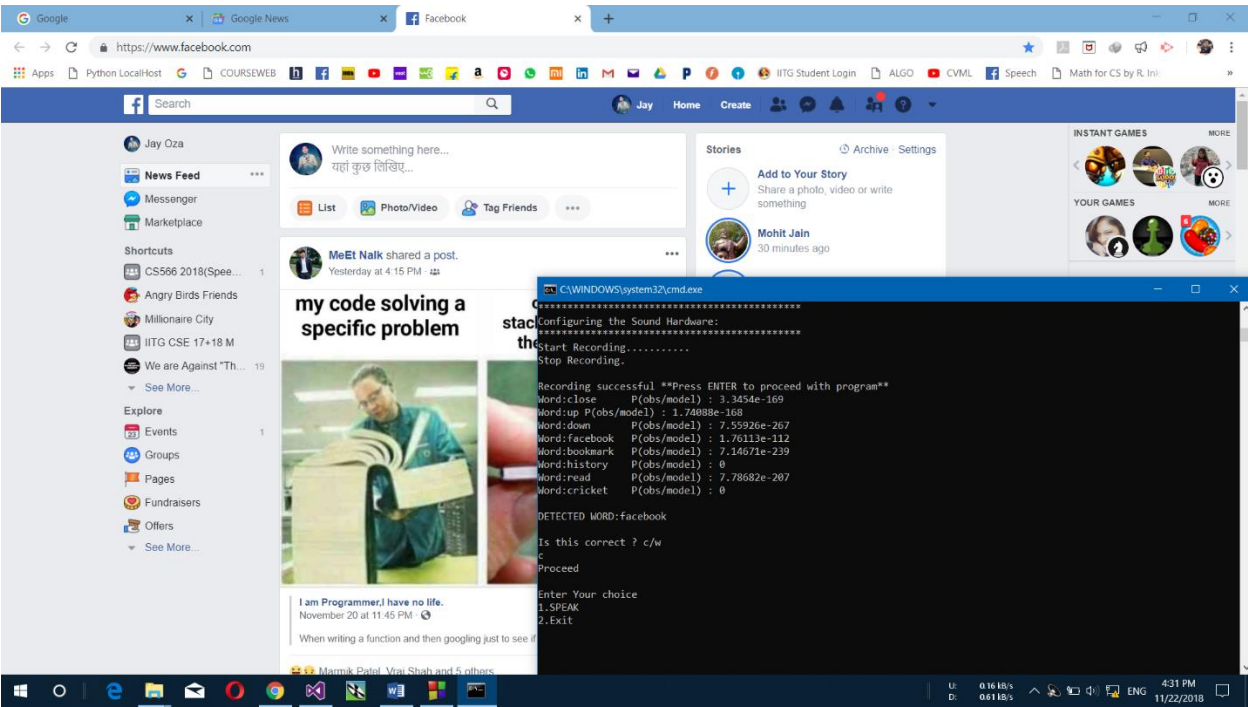
### 3) Read News



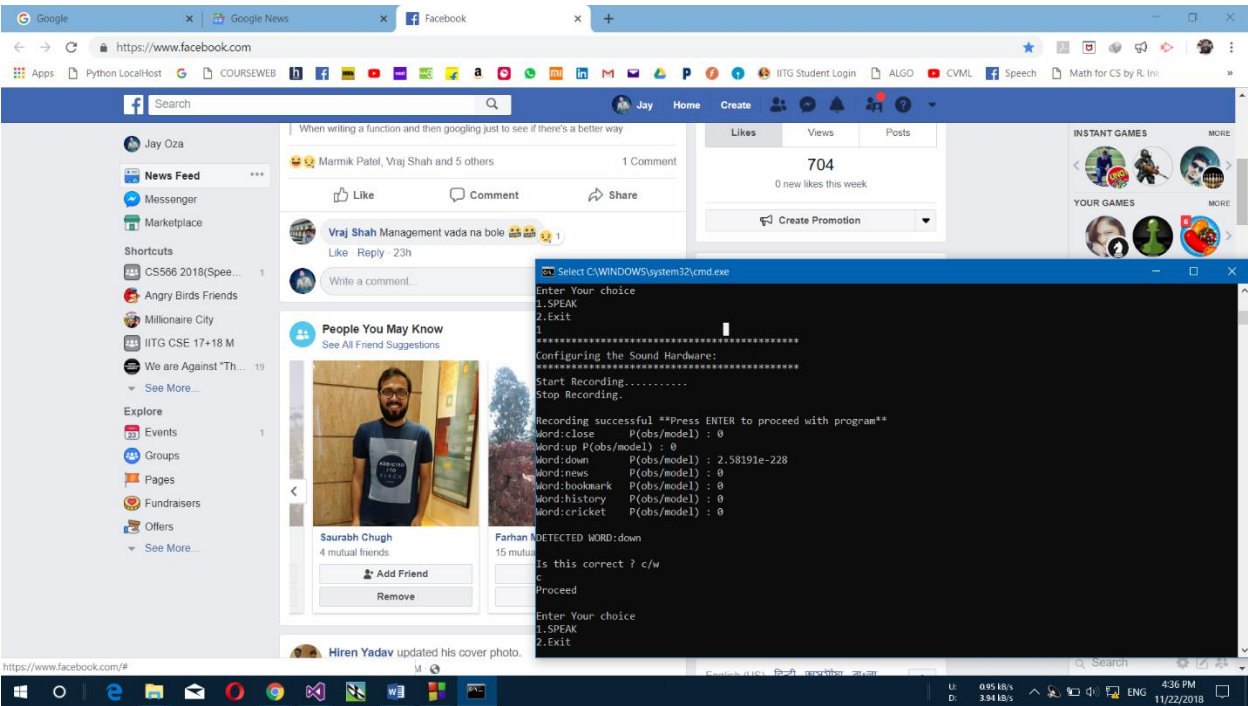
### 4) Stop Reading



5) Open Facebook

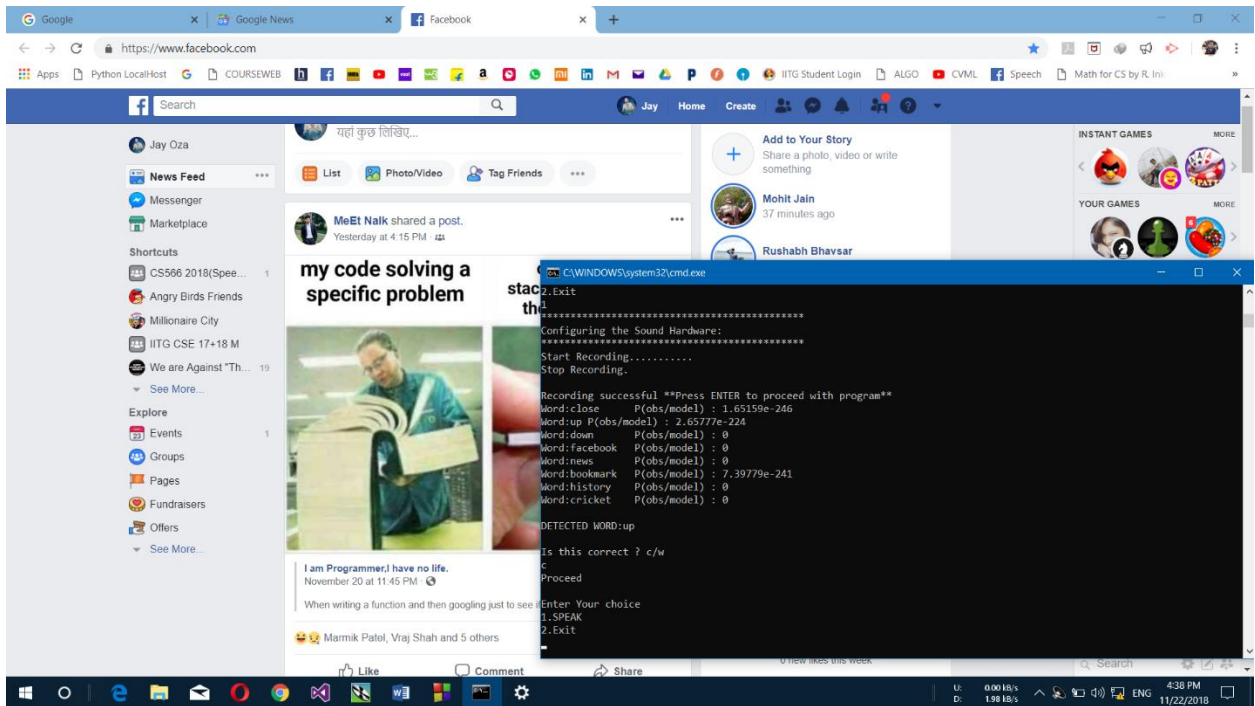


6) Scroll Down

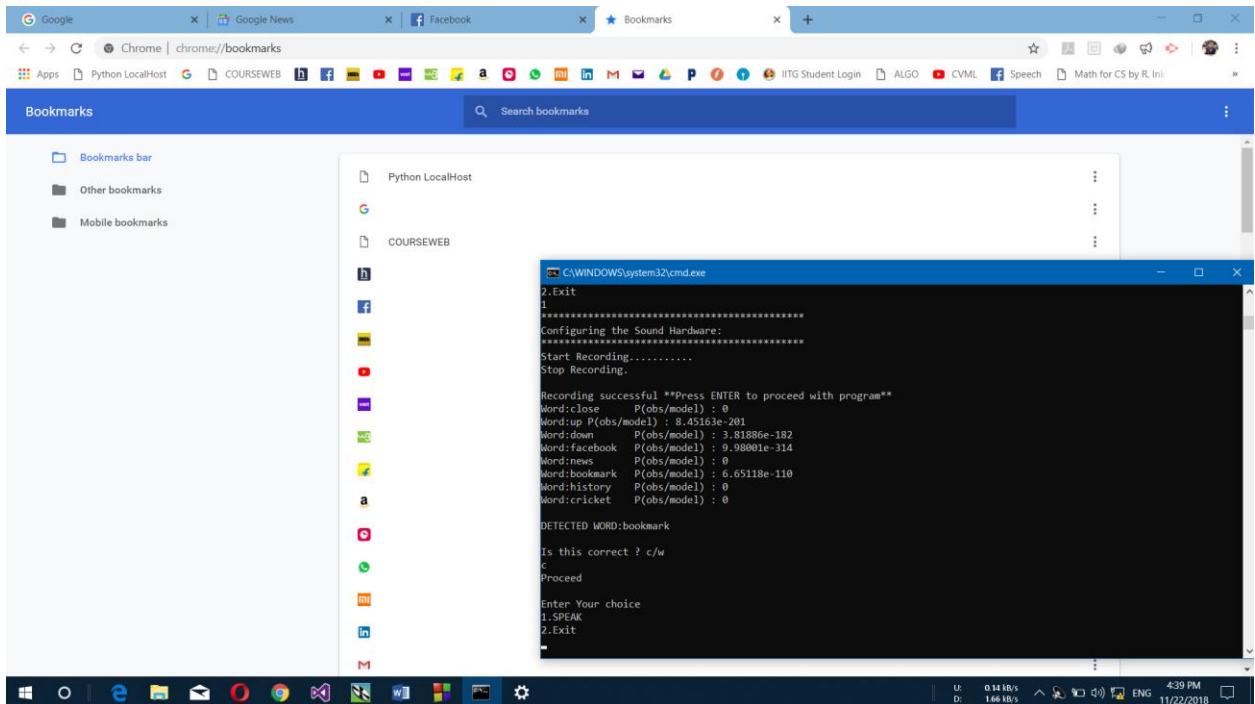




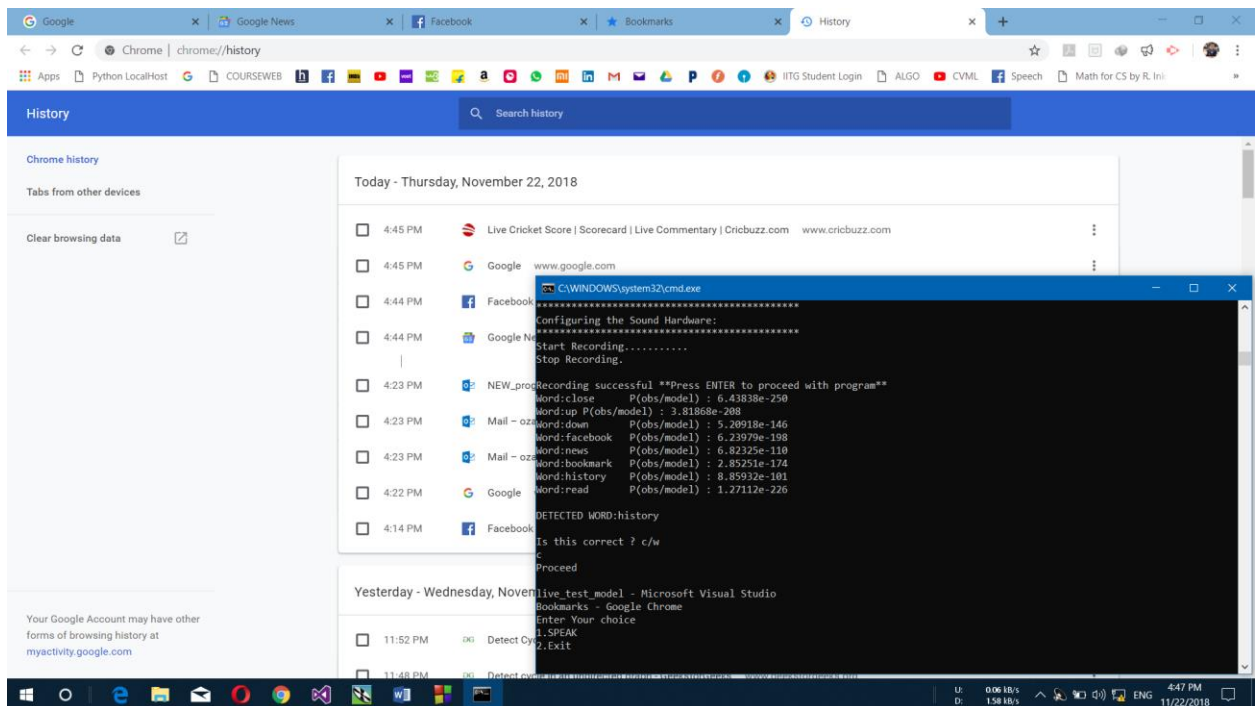
## 7) Scroll Up



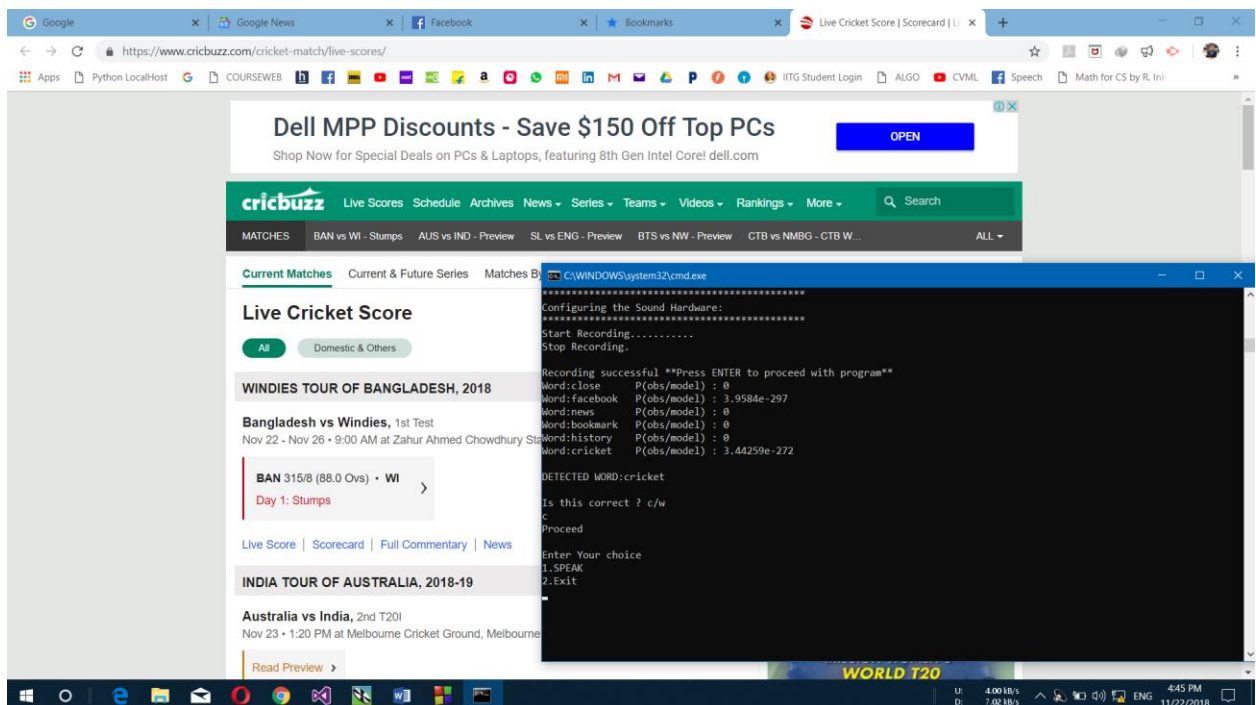
## 8) Bookmark



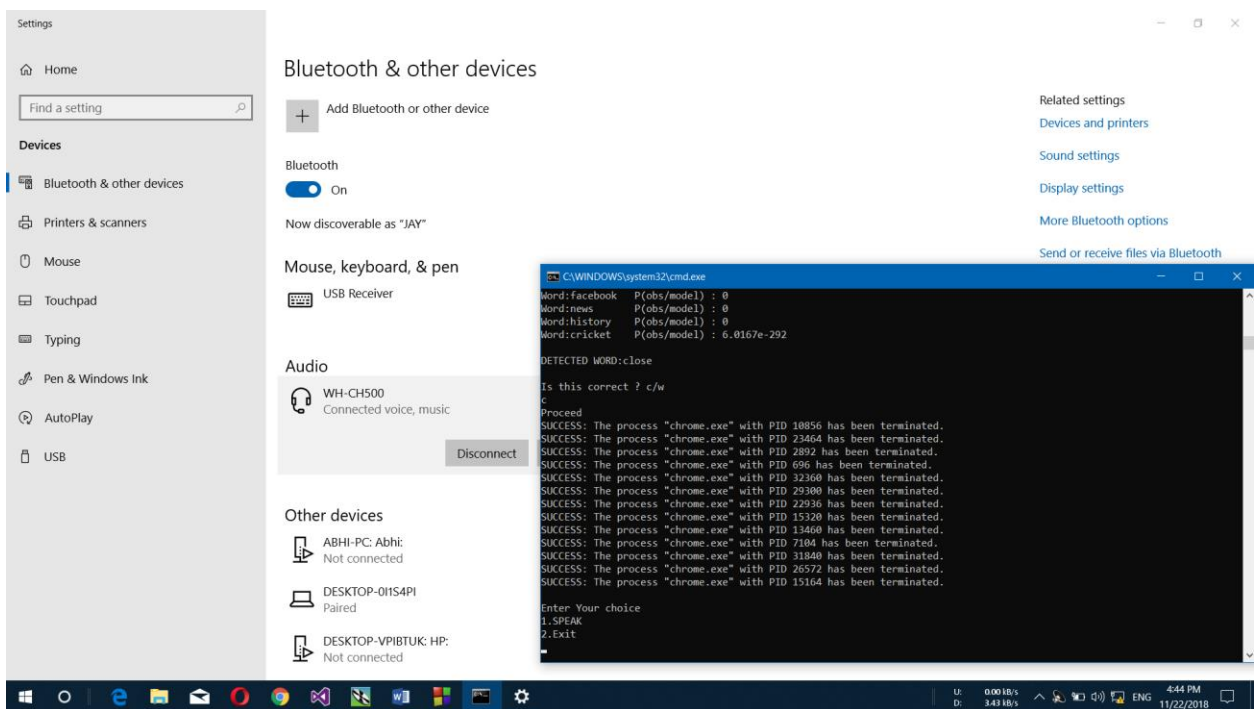
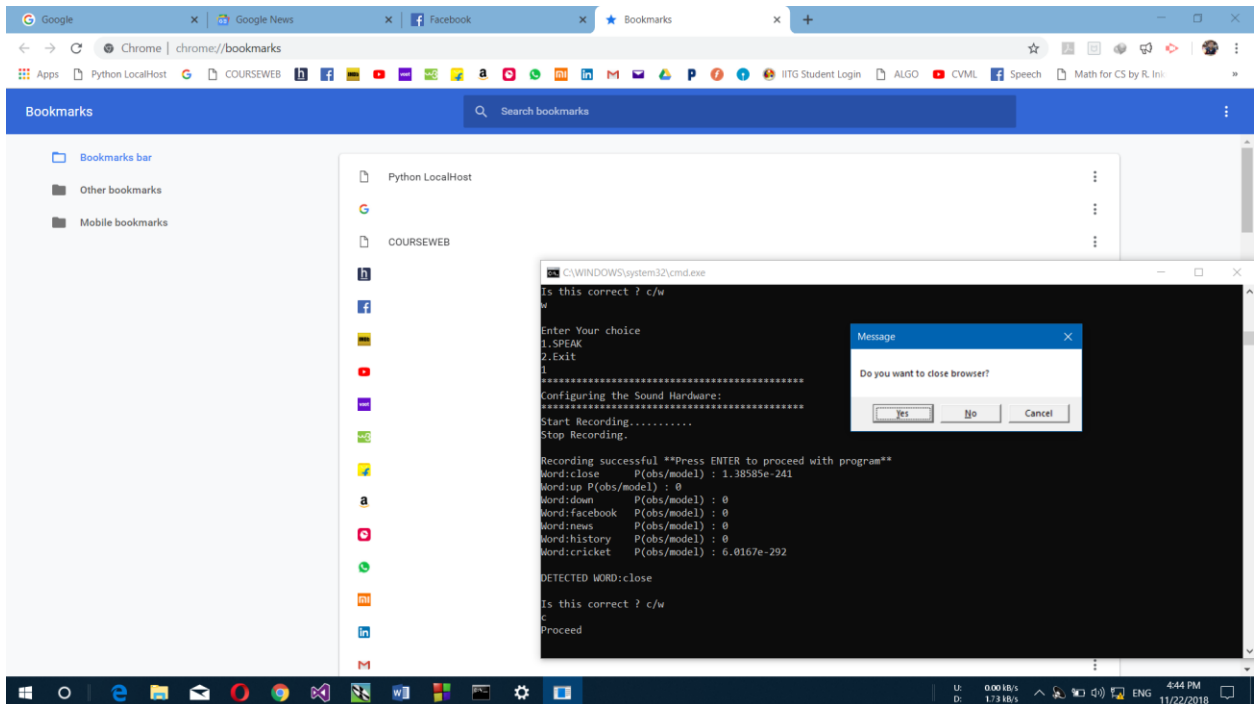
## 9) History



## 10) Cricket Score



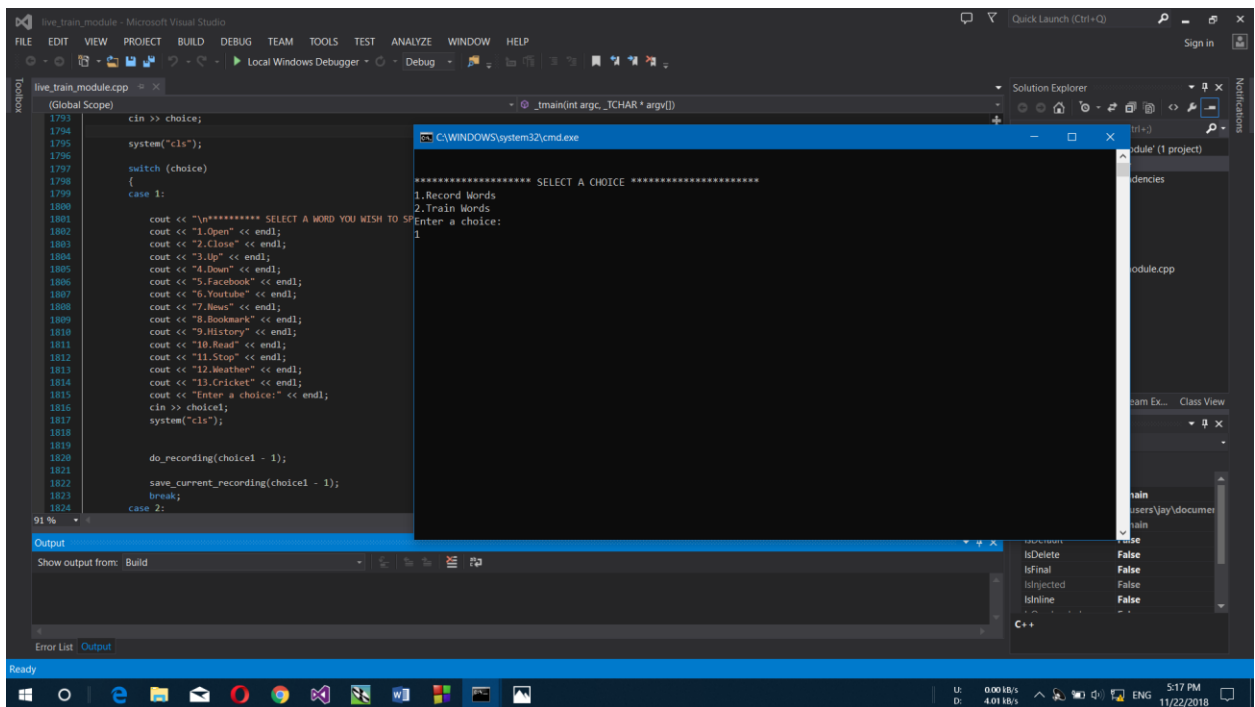
## 11) Close



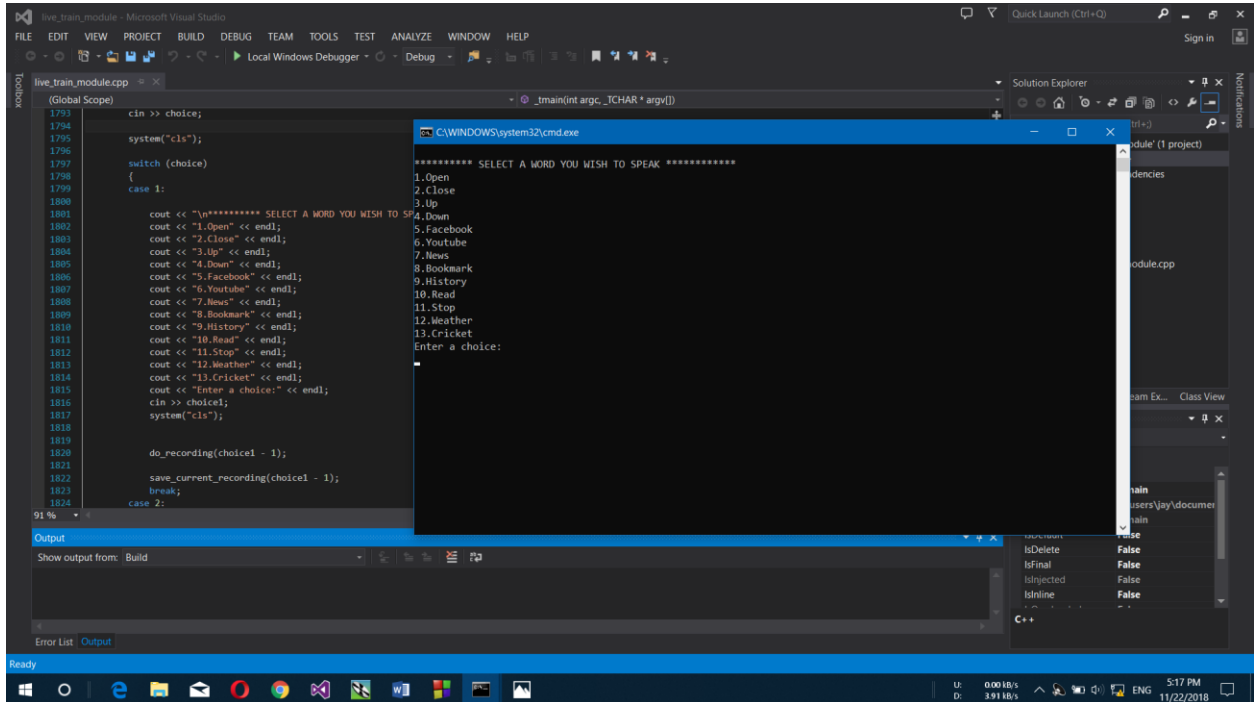
## Live Training Module Screenshots:

- Choose whether to record or train.
- Select a word for which you want to record.
- Data will be saved automatically into Folder hierarchy “saved\_recordings/words\_name/”.
- Counter files for each of the recorded words is kept to keep track of current iteration and file is saved at : “saved\_recordings/iteration\_wordname.txt”.
- Counter for all files can be reset using 999 as input choice in main menu to start recording fresh and saved data gets overwritten.

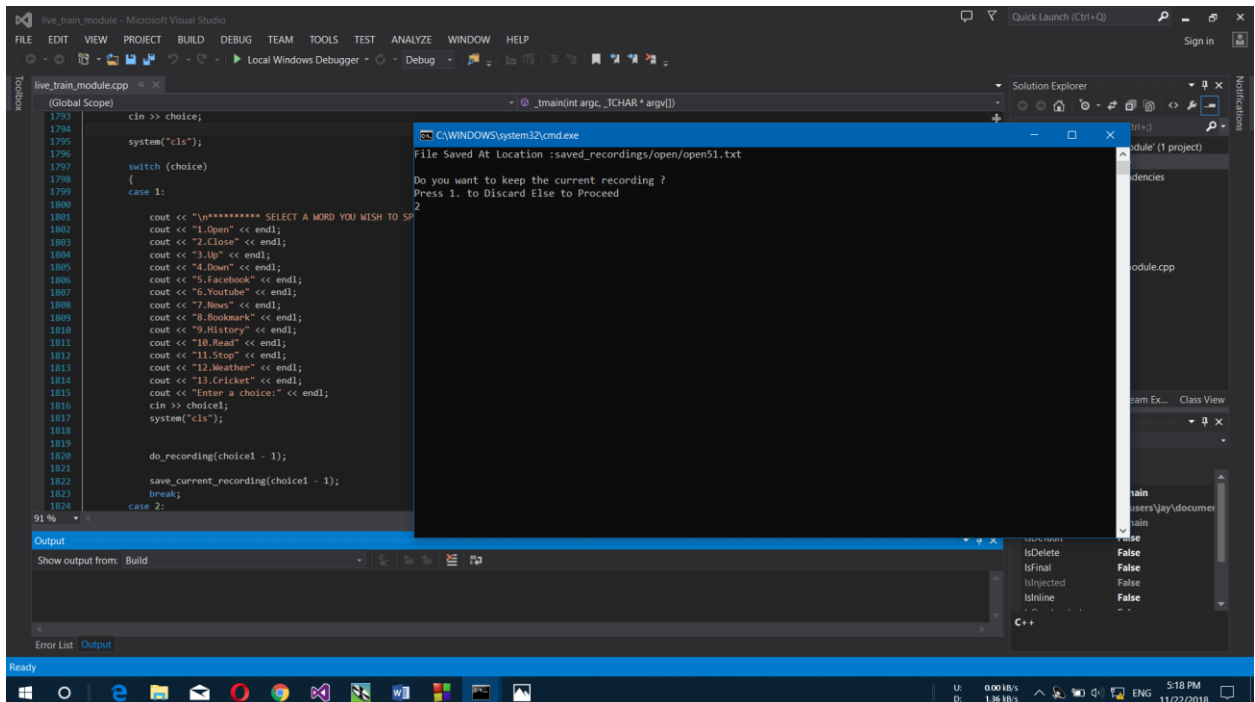
### 1) Main menu: Record words or Train with recorded Data



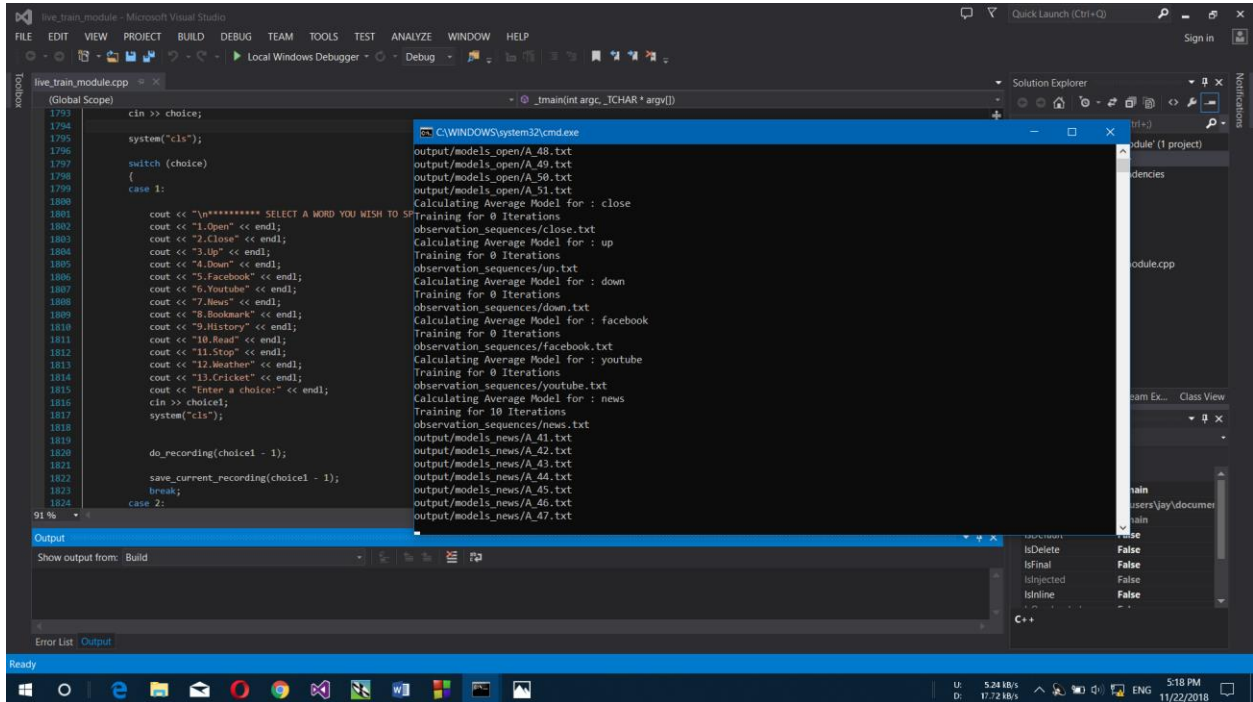
## 2) Recording from list of pre-defined Words



## 3) Saving or Discarding



#### 4) Training all the saved data.



### Implementation Details

- All the links and websites are triggered using system commands.
- The activities of switching tab, opening and closing of add-ons, scroll up and scroll down are controlled using virtual keyboard inputs.
- Add on used for reading web pages: “Read Aloud – A text to speech voice reader”.
- Scan code for different keys used are as below:

```
static const WORD TAB_SCANCODE = 0x0f;  
static const WORD LEFT_ALT_SCANCODE = 0x38;  
static const WORD LCTRL_SCANCODE = 0x1d;  
static const WORD H_SCANCODE = 0x23;  
static const WORD B_SCANCODE = 0x30;  
static const WORD J_SCANCODE = 0x24;  
static const WORD K_SCANCODE = 0x25;  
static const WORD O_SCANCODE = 0x18;
```



```
static const WORD P_SCANCODE = 0x19;  
static const WORD NUMPAD_5 = 0x4c;  
static const WORD LSHIFT_SCANCODE = 0x2a;  
static const DWORD keypress_delay_ms = 500;
```

- Currently the default call is to Chrome browser which can be extended to other browsers as well.
- Since there are 11 words in total we have cut down the list of recognized words at any particular moment by cutting down some of the command which cannot be triggered at that moment and ignoring them for calculations.

E.g: Once the browser is Open then only it can be "Closed " so before opening browser we can ignore checking P(O/model) for Close.

## **Future Scope**

- Adding more websites like YouTube and Gmail which are more frequent can be added.
- Weather Forecasting can be added which belongs to text to speech category and be narrated.
- Narrator in Windows can be used to gain more control over which text to read.
- Continuous recording module can be added to minimize keyboard interaction to almost none.
- Websites for playing songs can be added and playback can be managed using Speech.

## **Result and Accuracy:**

The accuracy of the system for prerecorded file is approximately 97%

For live testing the accuracy is achieved up to 94%.

For a new user the accuracy achieved is up to 90%.