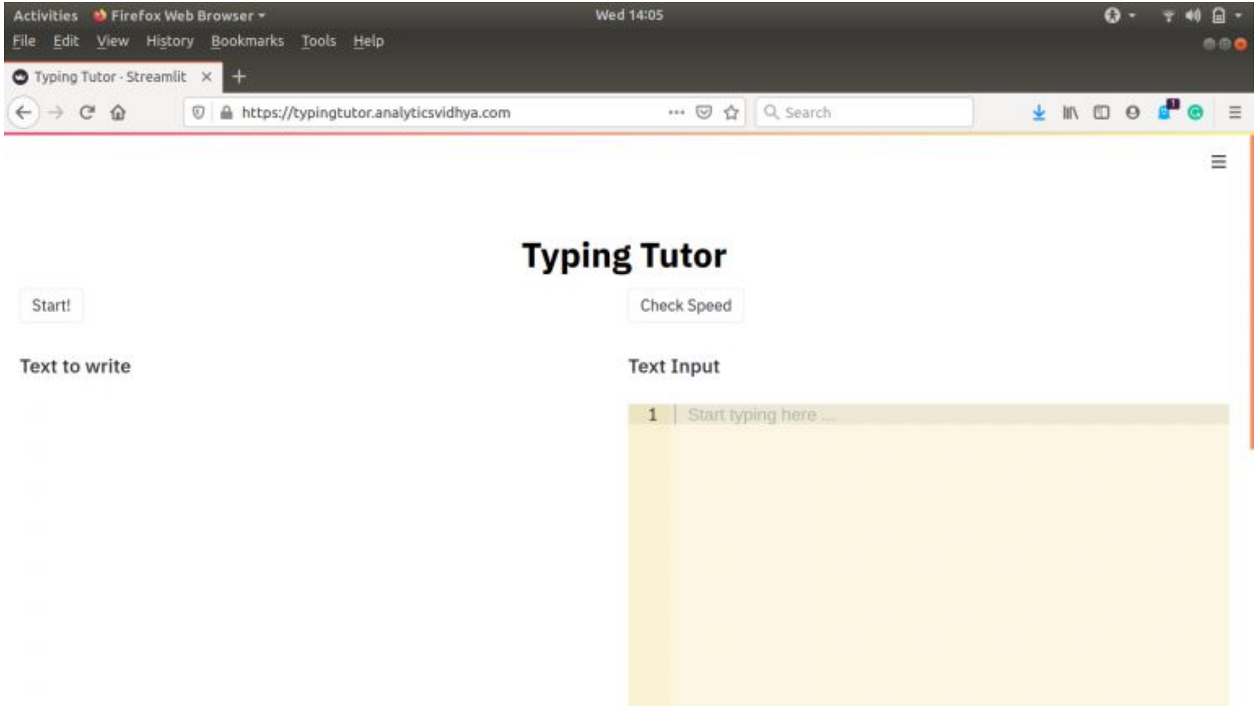
Creating the Front-end of Typing Tutor

What we will be covering in this module?

- Understand Text Generation Project
- Create Front-end of the Project
- Building a Text Generation model
- Deploying Text Generation model using Streamlit and AWS
- Setting up an accessible website

Problem Statement - Typing Tutor for Coders



Typing Tutor

Text to write

Text Input

```
1  Start typing here ...
```

Steps to create Typing Tutor

- Deploy the model using streamlit (Front-end)
    - Install required libraries
    - Setup DL model using streamlit
    - Deploy DL model on Colab

- Build a Text Generation model (Backend)
    - Install required libraries
    - Define deep learning model
    - Preprocess data and Get prediction
    - Deploy DL model on AWS

Blueprint of Typing Tutor

- class **TypingTutor:** Class for text generation in streamlit

- class **SessionState:** Class for maintaining session per user

Blueprint of Typing Tutor

- class TypingTutor

__init__()	Defines what happens when the website starts
_code_gen()	Function for text generation
_get_perf()	Function to get typing speed and accuracy
on_start_click()	Defines what happens when the start button is clicked
on_eval_click()	Defines what happens when the eval button is clicked

Blueprint of Typing Tutor

- class SessionState

start_time	The time when the user starts writing the code
end_time	The time when the user has written the code
num_chars	Number of characters to write
text	The overall code to be written
content	The code written by the user

Flow Chart of the website

- Website starts

Flow Chart of the website

- Website starts → create instance of class *TypingTutor*

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
    - Creates a unique session for the user
    - Creates the front-end for the website
    - Initializes and loads DL model

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which
  - Internally calls **_code_gen()** to generate code

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
    - Creates a unique session for the user
    - Creates the front-end for the website
    - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which
    - Internally calls **_code_gen()** to generate code
    - Modifies session state variables (**start_time, text, num_chars**)

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which
  - Internally calls **_code_gen()** to generate code
  - Modifies session state variables (**start_time, text, num_chars**)
  - Updates the front-end accordingly

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which
  - Internally calls **_code_gen()** to generate code
  - Modifies session state variables (**start_time, text, num_chars**)
  - Updates the front-end accordingly

- Click on “Check Speed” button

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which
  - Internally calls **_code_gen()** to generate code
  - Modifies session state variables (**start_time, text, num_chars**)
  - Updates the front-end accordingly

- Click on “Check Speed” button → calling **on_eval_click()**, which

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which
  - Internally calls **_code_gen()** to generate code
  - Modifies session state variables (**start_time, text, num_chars**)
  - Updates the front-end accordingly

- Click on “Check Speed” button → calling **on_eval_click()**, which
  - Internally calls **_get_perf()** to calculate performance

Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which
  - Internally calls **_code_gen()** to generate code
  - Modifies session state variables (**start_time, text, num_chars**)
  - Updates the front-end accordingly

- Click on “Check Speed” button → calling **on_eval_click()**, which
  - Internally calls **_get_perf()** to calculate performance
  - Modifies session state variables (**end_time, content**)

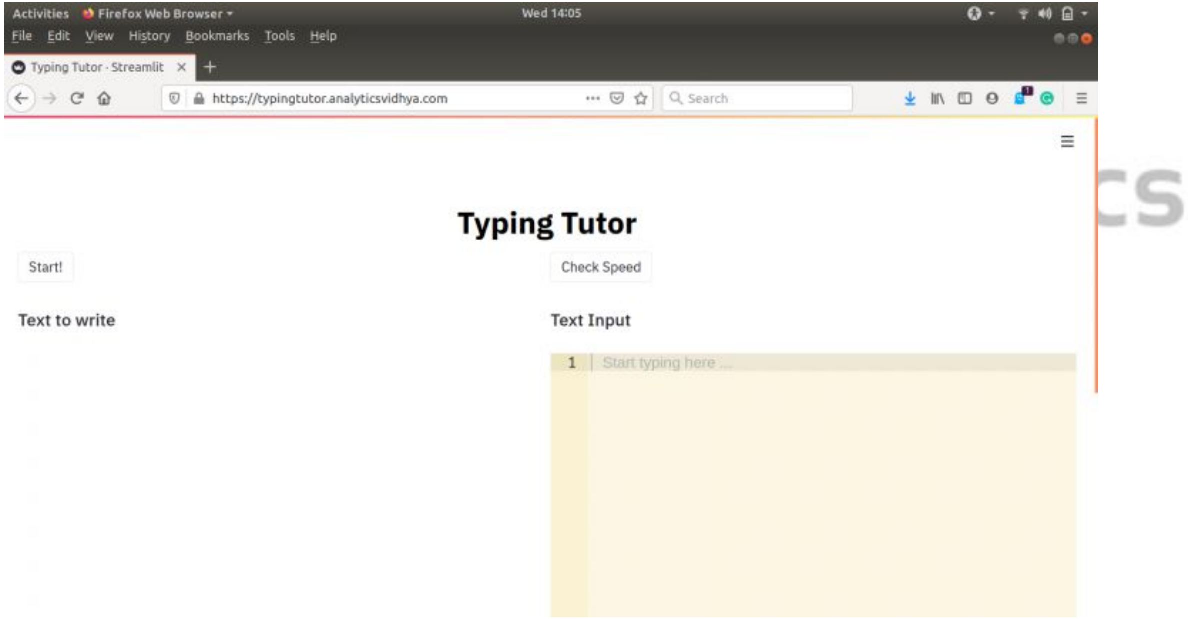
Flow Chart of the website

- Website starts → create instance of class *TypingTutor* → calls **__init__()**
  - Creates a unique session for the user
  - Creates the front-end for the website
  - Initializes and loads DL model

- Click on “Start” button → calling **on_start_click()**, which
  - Internally calls **_code_gen()** to generate code
  - Modifies session state variables (**start_time, text, num_chars**)
  - Updates the front-end accordingly

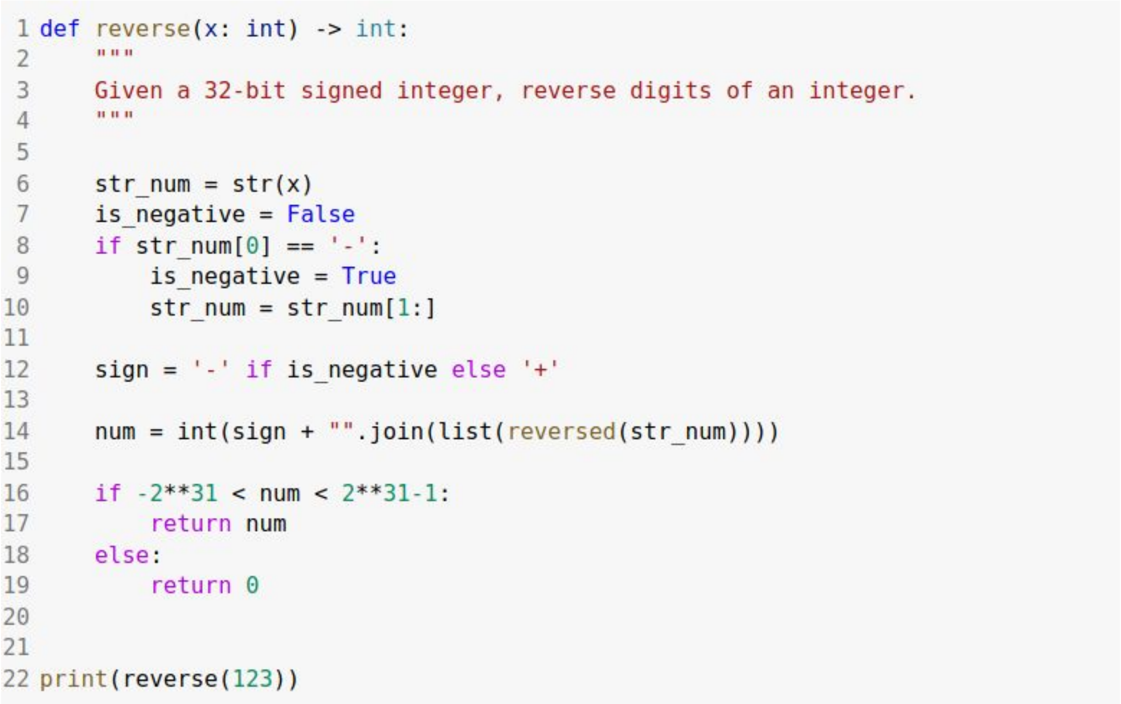
- Click on “Check Speed” button → calling **on_eval_click()**, which
  - Internally calls **_get_perf()** to calculate performance
  - Modifies session state variables (**end_time, content**)
  - Updates the front-end accordingly

Problem Statement - Typing Tutor for Coders



Front-end

Problem Statement - Typing Tutor for Coders



```python
def reverse(x: int) -> int:
    """
    Given a 32-bit signed integer, reverse digits of an integer.
    """

    str_num = str(x)
    is_negative = False
    if str_num[0] == '-':
        is_negative = True
        str_num = str_num[1:]

    sign = '-' if is_negative else '+'

    num = int(sign + "".join(list(reversed(str_num))))

    if -2**31 < num < 2**31-1:
        return num
    else:
        return 0


print(reverse(123))
```

example_code.py

Thank you