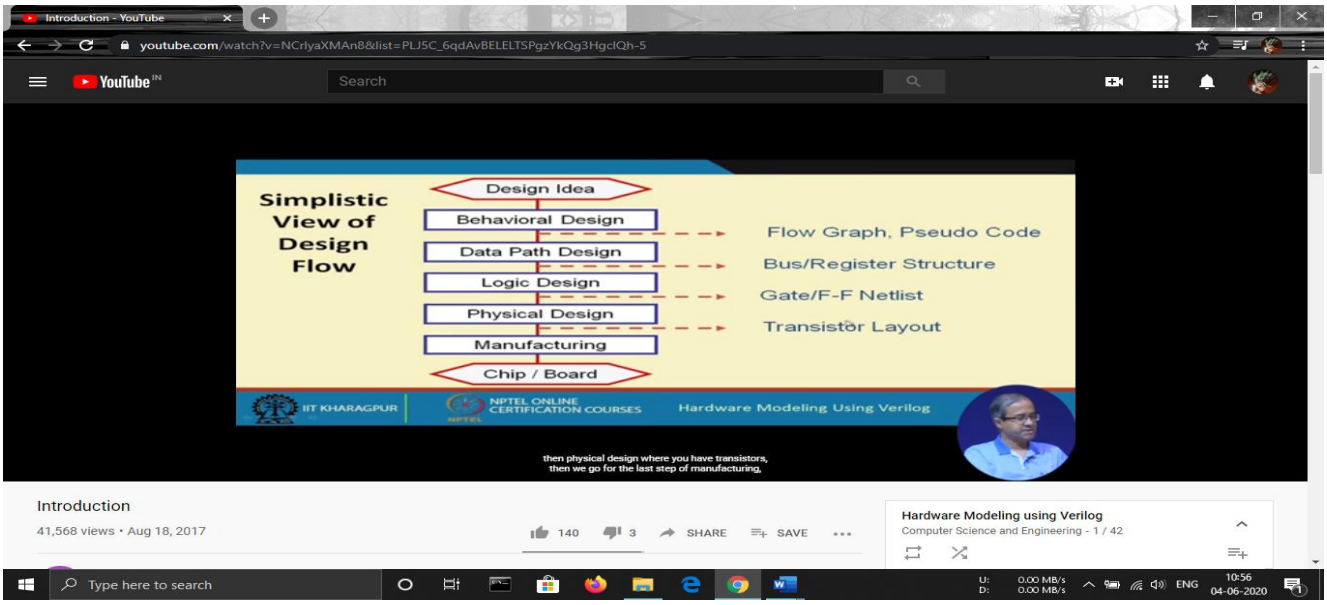
DAILY ASSESSMENT REPORT

Date:	**04/06/2020**	**Name:**	**Abhishek M Shastry K**
Subject:	**Digital Design Using HDL**	**USN:**	**4AL17EC002**
Topic:	**1] Hardware modelling using verilog** **2] FPGA and ASIC Interview questions**	**Semester & Section:**	**6th 'A'**
Github Repository:	**AbhishekShastry-Courses**		

FORENOON SESSION DETAILS

Image of session

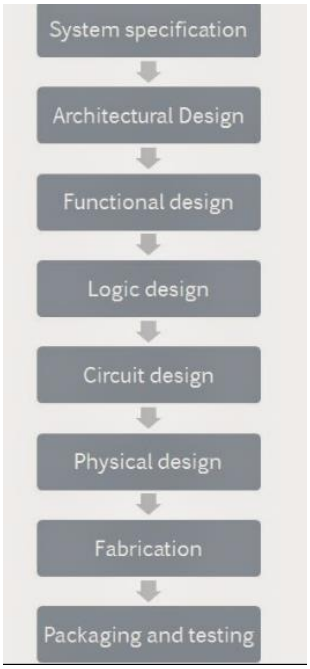


Report

VLSI Design

- Very-large-scale integration (VLSI) is the process of creating an integrated circuit (IC) by combining thousands of transistors into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device.
- The electronics industry has achieved a phenomenal growth over the last few decades, mainly due to the rapid advances in large scale integration technologies and system design applications. With the advent of very large-scale integration (VLSI) designs, the number of applications of integrated circuits (ICs) in high-performance computing, controls, telecommunications, image and video processing, and consumer electronics has been rising at a very fast pace.
- The current cutting-edge technologies such as high resolution and low bit-rate video and cellular communications provide the end-users a marvellous amount of applications, processing power and portability. This trend is expected to grow rapidly, with very important implications on VLSI design and systems design.

VLSI Design Flow



The VLSI design cycle starts with a formal specification of a VLSI chip, follows a series of steps, and eventually produces a packaged chip.

1. **System Specification:**

- The first step of any design process is to lay down the specifications of the system. System specification is a high-level representation of the system. The factors to be considered in this process include: performance, functionality, and physical dimensions (size of the die (chip)). The fabrication technology and design techniques are also considered.

2. **Architectural Design:**

- The basic architecture of the system is designed in this step. This includes, such decisions as RISC (Reduced Instruction Set Computer) versus CISC (Complex Instruction Set Computer), number of ALUs, Floating Point units, number and structure of pipelines, and size of caches among others.

3. **Behavioral or Functional Design:**

- In this step, main functional units of the system are identified. This also identifies the interconnect requirements between the units. The area, power, and other parameters of each unit are estimated.
- The behavioral aspects of the system are considered without implementation specific information.

4. **Logic Design:**

- In this step the control flow, word widths, register allocation, arithmetic operations, and logic operations of the design that represent the functional design are derived and tested.
- This description is called Register Transfer Level (RTL) description. RTL is expressed in a Hardware Description Language (HDL), such as VHDL or Verilog. This description can be used in simulation and verification.

5. **Circuit Design:**

- The purpose of circuit design is to develop a circuit representation based on the logic design. The Boolean expressions are converted into a circuit representation by taking into consideration the speed and power requirements of the original design. Circuit Simulation is used to verify the correctness and timing of each component.

6. **Physical Design:**

- In this step the circuit representation (or netlist) is converted into a geometric representation. As stated earlier, this geometric representation of a circuit is called a layout. Layout is created by converting each logic component (cells, macros, gates, transistors) into a geometric representation (specific shapes in multiple layers), which perform the intended logic function of the corresponding component. Connections between different components are also expressed as geometric patterns typically lines in multiple layers.

7. **Fabrication:**

- After layout and verification, the design is ready for fabrication. Since layout data is typically sent to fabrication on a tape, the event of release of data is called Tape Out. Layout data is converted (or fractured) into photo-lithographic masks, one for each layer. Masks identify spaces on the wafer, where certain materials need to be deposited, diffused or even removed. Silicon crystals are grown and sliced to produce wafers. Extremely small dimensions of VLSI devices require that the wafers be polished to near perfection. The fabrication process consists of several steps involving deposition, and diffusion of various materials on the wafer. During each step one mask is used. Several dozen masks may be used to complete the fabrication process.

8. **Packaging, Testing and Debugging:**

- Finally, the wafer is fabricated and diced into individual chips in a fabrication facility. Each chip is then packaged and tested to ensure that it meets all the design specifications and that it functions properly. Chips used in Printed Circuit Boards (PCBs) are packaged in Dual In-line Package (DIP), Pin Grid Array (PGA), Ball Grid Array (BGA), and Quad Flat Package (QFP). Chips used in Multi-Chip Modules (MCM) are not packaged, since MCMs use bare or naked chips.

Moore's Law

- Moore's Law refers to Moore's perception that the number of transistors on a microchip doubles every two years, though the cost of computers is halved. Moore's Law states that we can expect the speed and capability of our computers to increase every couple of years, and we will pay less for them. Another tenet of Moore's Law asserts that this growth is exponential.

Task (DAY - 4)

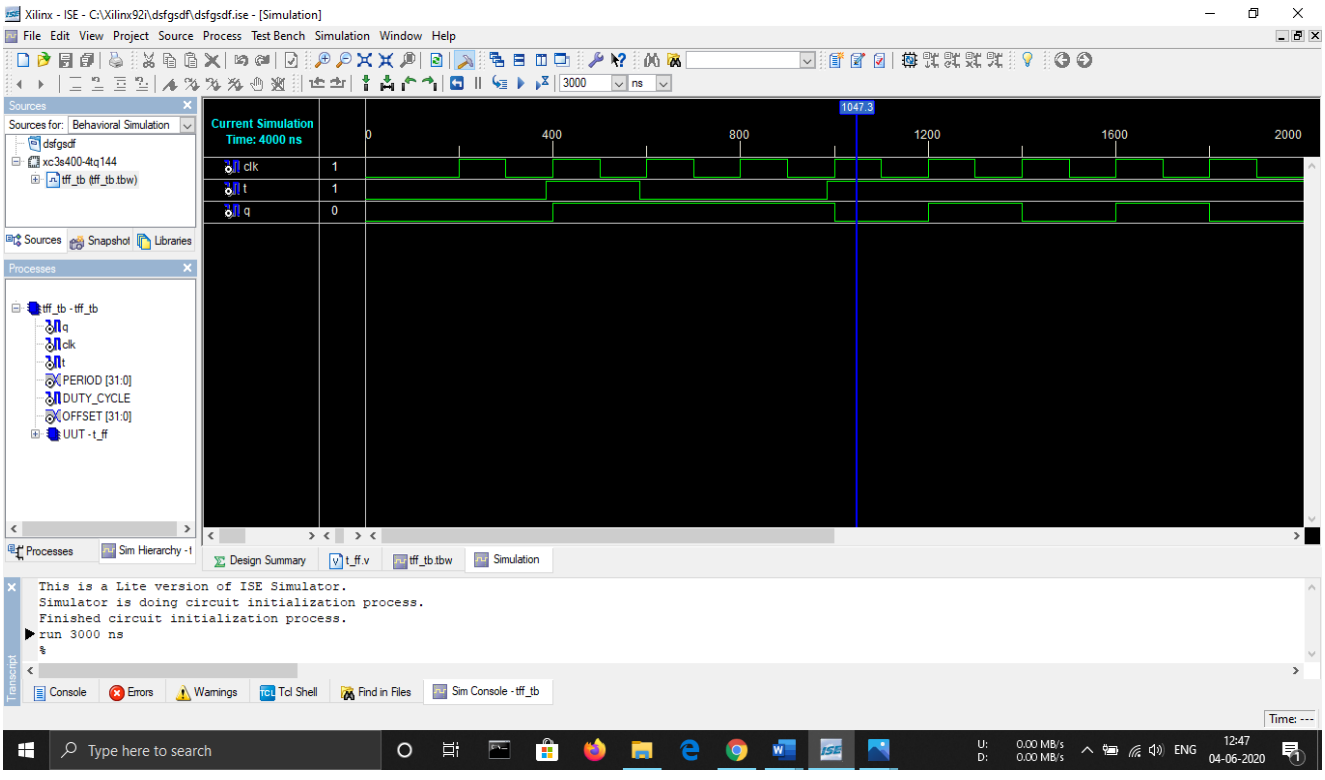
Implement a simple T Flipflop and test the module using a compiler.

Verilog Code:

```
module t_ff (t,q,clk);
input   t,clk;
output reg q = 0;

always @ (posedge clk)
begin
     if (t==1)
        begin
          q=~q;
        end
     else
        begin
          q=q;
        end
end
endmodule
```

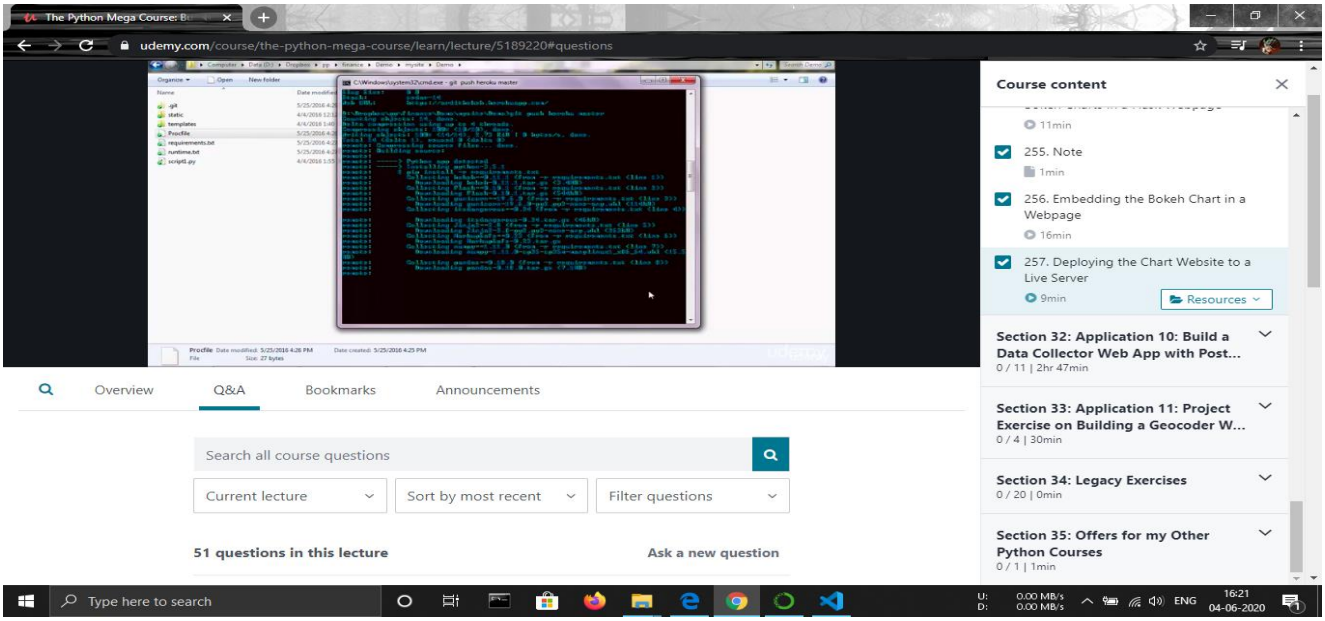
Compiler Output:

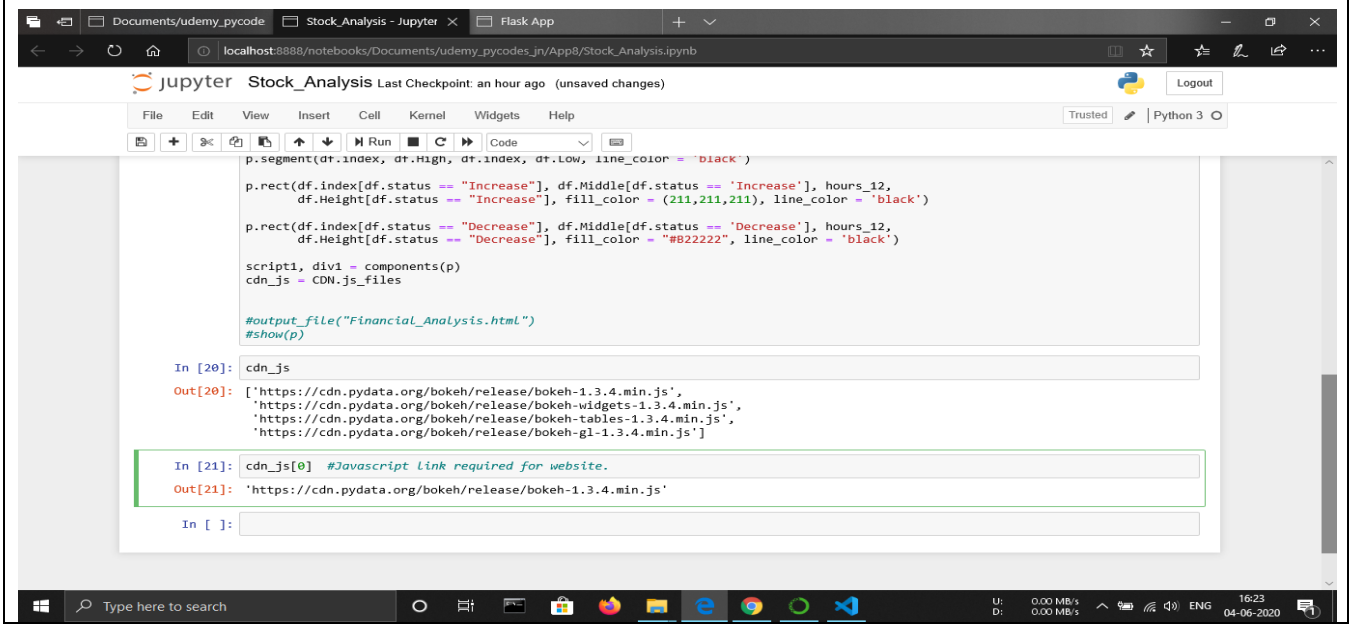


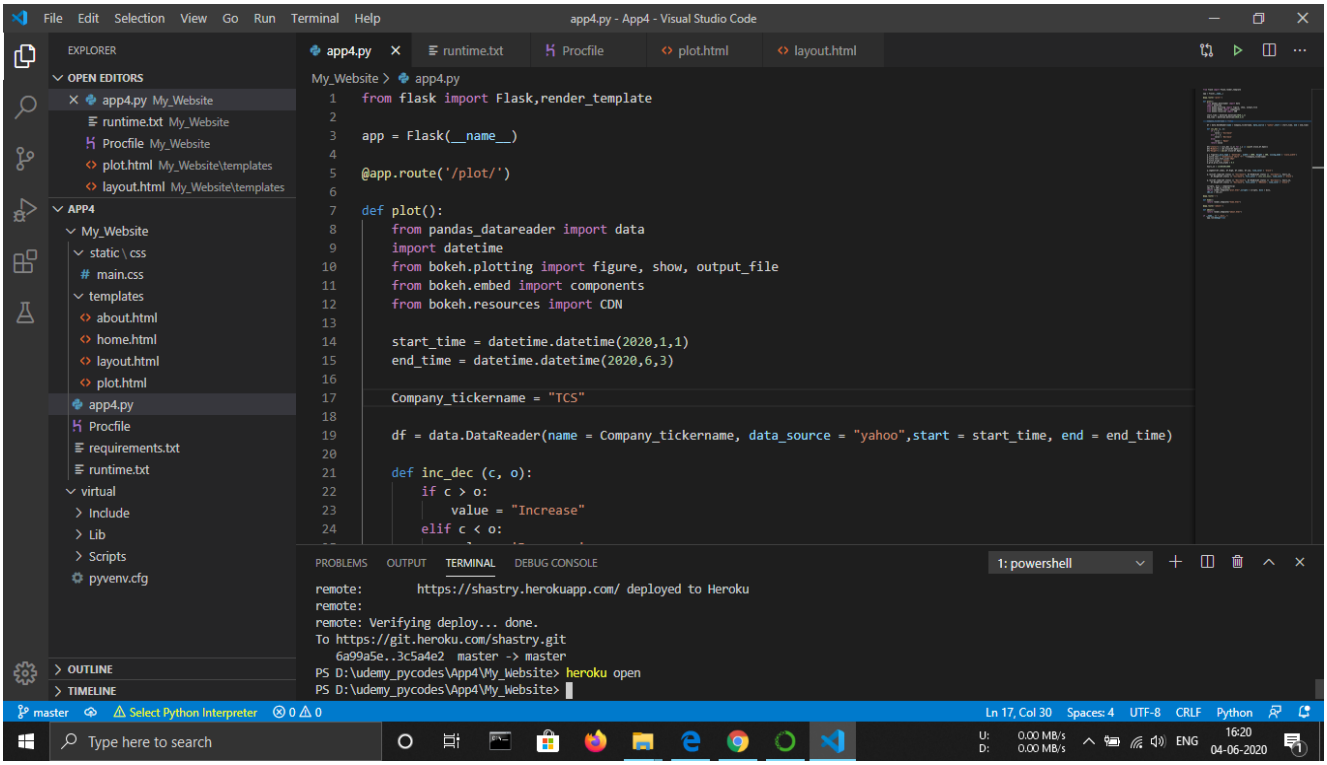
Date:	04/06/2020	Name:	Abhishek M Shastry K
Course:	The Python Mega Course: Build 10 Real World Applications	USN:	4AL17EC002
Topic:	1] Application 8: Build a Web-based Financial Graph	Semester & Section:	6th 'A'
Github Repository:	AbhishekShastry-Courses		

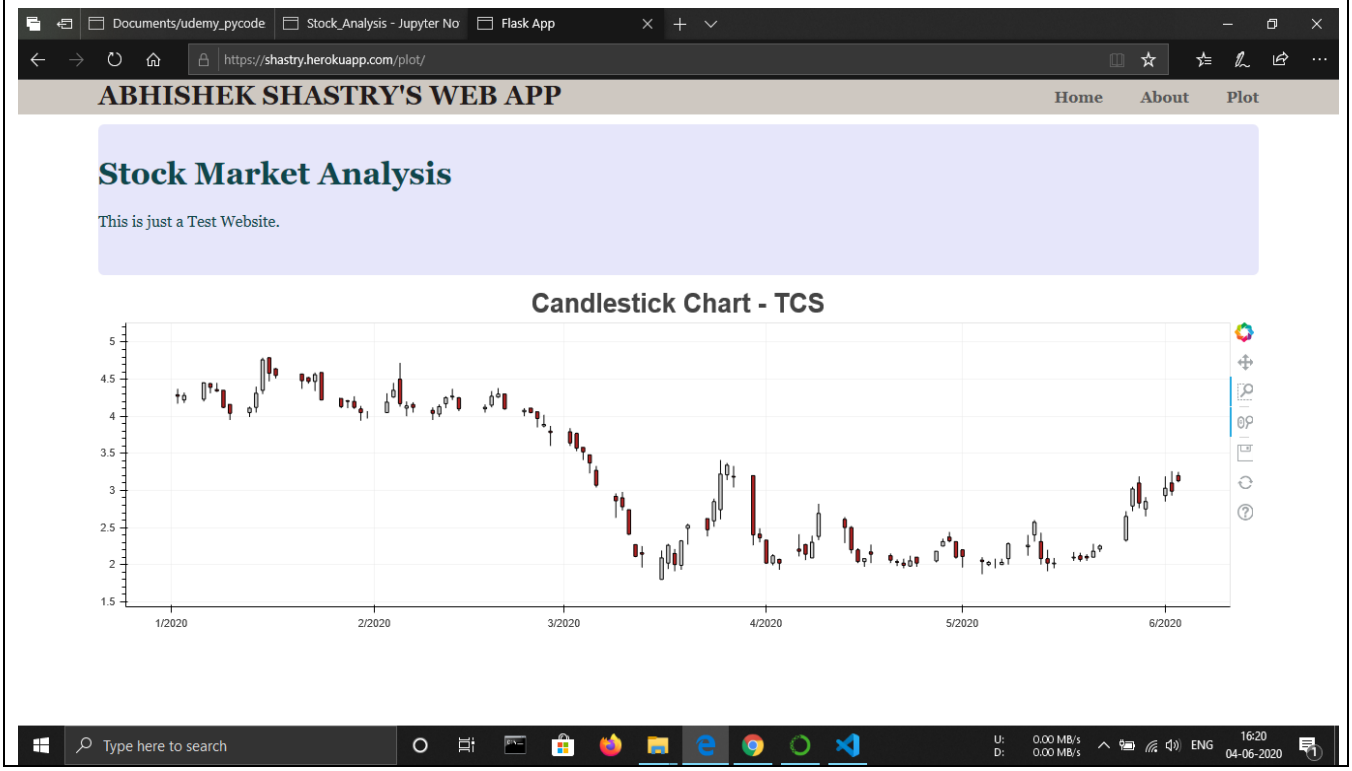
AFTERNOON SESSION DETAILS

Image of session






File Edit Selection View Go Run Terminal Help
app4.py - App4 - Visual Studio Code
EXPLORER
OPEN EDITORS
app4.py My_Website
runtime.txt My_Website
Procfile My_Website
plot.html My_Website\templates
layout.html My_Website\templates
APP4
My_Website
static\css
main.css
templates
about.html
home.html
layout.html
plot.html
app4.py
Procfile
requirements.txt
runtime.txt
virtual
Include
Lib
Scripts
pyvenv.cfg
app4.py
runtime.txt
Procfile
plot.html
layout.html
My_Website > app4.py
from flask import Flask,render_template
app = Flask(__name__)
@app.route('/plot/')
def plot():
from pandas_datareader import data
import datetime
from bokeh.plotting import figure, show, output_file
from bokeh.embed import components
from bokeh.resources import CDN
start_time = datetime.datetime(2020,1,1)
end_time = datetime.datetime(2020,6,3)
Company_tickername = "TCS"
df = data.DataReader(name = Company_tickername, data_source = "yahoo",start = start_time, end = end_time)
def inc_dec (c, o):
if c > o:
value = "Increase"
elif c < o:
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: powershell
remote: https://shastry.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/shastry.git
6a99a5e..3c5a4e2 master -> master
PS D:\udemy_pycodes\App4\My_Website> heroku open
PS D:\udemy_pycodes\App4\My_Website>
OUTLINE
TIMELINE
master
Select Python Interpreter
Ln 17, Col 30 Spaces: 4 UTF-8 CRLF Python
Type here to search
ENG
16:20
04-06-2020


Documents/udemy_pycode
Stock_Analysis - Jupyter No
Flask App
https://shastry.herokuapp.com/plot/
ABHISHEK SHASTRY'S WEB APP
Home
About
Plot
Stock Market Analysis
This is just a Test Website.
Candlestick Chart - TCS
5
4.5
4
3.5
3
2.5
2
1.5
1/2020
2/2020
3/2020
4/2020
5/2020
6/2020
Type here to search
ENG
16:20
04-06-2020

Report

Application 8: Build a Web-based Financial Graph

- Python script to plot stock market data using bokeh library and deploy the bokeh plot to a live website.
- A ticker symbol or stock symbol is an abbreviation used to uniquely identify publicly traded shares of a particular stock on a particular stock market. A stock symbol may consist of letters, numbers or a combination of both. "Ticker symbol" refers to the symbols that were printed on the ticker tape of a ticker tape machine.
- Some of the examples are:
    - ✓ NYSE (New York Stock Exchange) uses the ticker symbol with 3 letters or few – such as 'NYT' for the New York Times Co. or 'T' for AT&T.
    - ✓ Symbols with 4 or more letters generally denote securities traded on the American stock exchange and NASDAQ.
    - ✓ Those ending in 'X' indicate mutual funds.
    - ✓ There are also certain symbols that denote specific status or type of security say, tickers ending in 'Q' indicate issuers which are under bankruptcy and letter 'Y' denotes security is an ADR.
- Some of the functions used under bokeh library:
    - ✓ It is possible to ask Bokeh to return the individual components of a standalone document for individual embedding using the **components ()** function under **bokeh.embed** module. This function returns a <script> that contains the data for your plot, together with an accompanying tag that the plot view is loaded into. These tags can be used in HTML documents however you like.
    - ✓ The **resources** module provides the Resources class for easily configuring how BokehJS code and CSS resources should be located, loaded, and embedded in Bokeh documents.
    - ✓ Additionally, functions for retrieving Sub resource Integrity hashes for Bokeh JavaScript files are provided here.
    - ✓ Content delivery network (CDN): Load minified BokehJS from CDN.