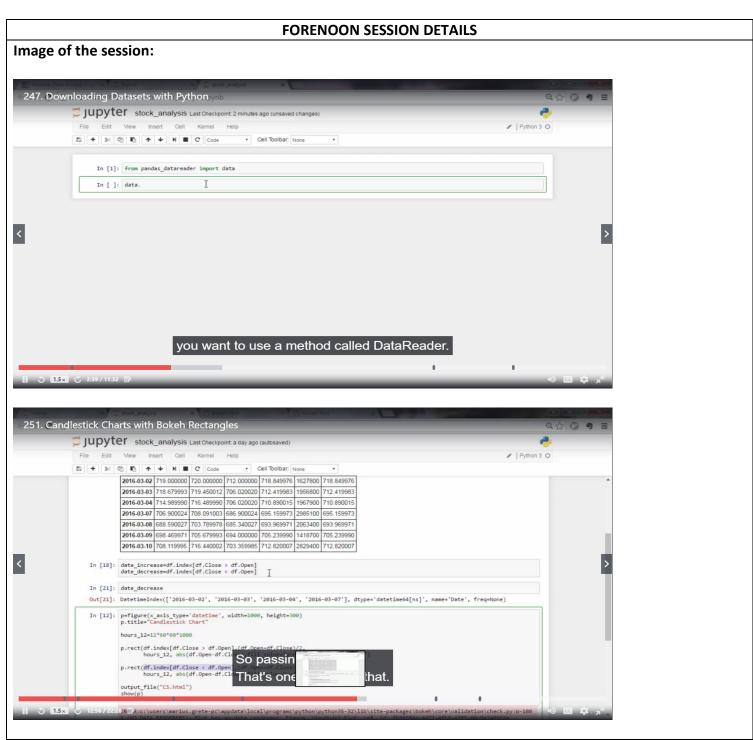
## **DAILY ASSESSMENT FORMAT**

Date:	04-06-2020	Name:	K B KUSHI
Course:	Python	USN:	4AL17EC107
Topic:	Application to Build a Web-based	Semester	6 & B
	Financial Graph	& Section:	
Github	https://github.com/alvas-		
Repository:	education-foundation/KUSHI-		
	COURSES.git		



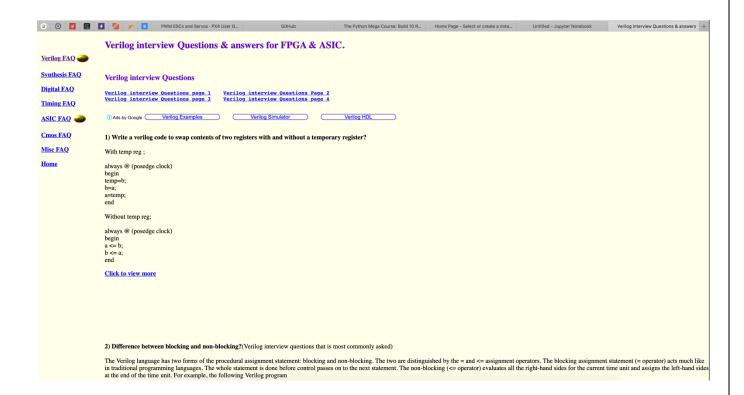


## Report – Report can be typed or hand written for up to two pages.

- In this section we learnt analyzing Stock Market Data, Plotting Stock Market Data Candlestick Charts,
   Updating Candlestick Charts with Bokeh Quadrants
- Learnt to plot Candlestick Charts with Bokeh Rectangles, Creating Candlestick Segments, Stylizing the obtained Chart
- Learnt the Concept Behind Embedding Bokeh, Sharing the Charts in a Flask Webpage
- Learnt how to Embed the Bokeh Chart in a Webpage and also learnt to Deploy the Chart Website to a Live Server.

Date:	04-06-2020	Name:	K B KUSHI	
Course:	DIGITAL DESIGN USING HDL	USN:	4AL17EC107	
Topic:	<ul> <li>Hardware modelling using         Verilog         FPGA and ASIC Interview         questions     </li> </ul>	Semester & Section:	6 <sup>тн</sup> & В	
Github Repository:	https://github.com/alvas-education- foundation/KUSHI-COURSES.git			

## Image of the session: Hardware Modeling using Verilog $\Box$ X =+ lardware Modeling Using Verilog Moore's Law **Design Representation** ware Modeling Using Verilog Getting Started with Verilog · Exponential growth Design complexity increases rapidly Automated tools are VLSI Design Styles (Part 2) essential Must follow well-VERILOG LANGUAGE FEATURES (PA... defined design flow 31:28 Hardware Modeling Using Verilog VERILOG LANGUAGE FEATURES (PA... GHOOMKETU I◀ II ►I ♠ 8:30 / 28:01 **■ • ■** □ ∷ Introduction Design Representation 41,568 views • 18 Aug 2017 1 140 4 3 → SHARE =+ SAVE ...



Report: In this section, we learnt about the Verilog hardware description language.

- The difference between behavioral and structural design styles.
- To write test benches and analyse simulation results.
- To model combinational and sequential circuits.

To distinguish between good and bad coding practices.
TASK: Implement a simple T Flipflop and test the module using a compiler.
Design:
module tff ( input clk, input rstn,
input t,
output reg q);
always O (necedes ally) havin
always @ (posedge clk) begin if (!rstn)
q <= 0;
else
if (t)
q <= ~q; else
q <= q;
end
endmodule
<u>Testbench</u>
module tb;
reg clk;
reg rstn; reg t;
leg t,
tff u0 ( .clk(clk),
.rstn(rstn),
.t(t), .q(q));
·4\4 <i>II</i>
always #5 clk = ~clk;

```
initial begin
{rstn, clk, t} <= 0;

$monitor ("T=%0t rstn=%0b t=%0d q=%0d", $time, rstn, t, q);
repeat(2) @(posedge clk);
rstn <= 1;

for (integer i = 0; i < 20; i = i+1) begin
reg [4:0] dly = $random;
#(dly) t <= $random;
end
#20 $finish;
end
endmodule</pre>
```

