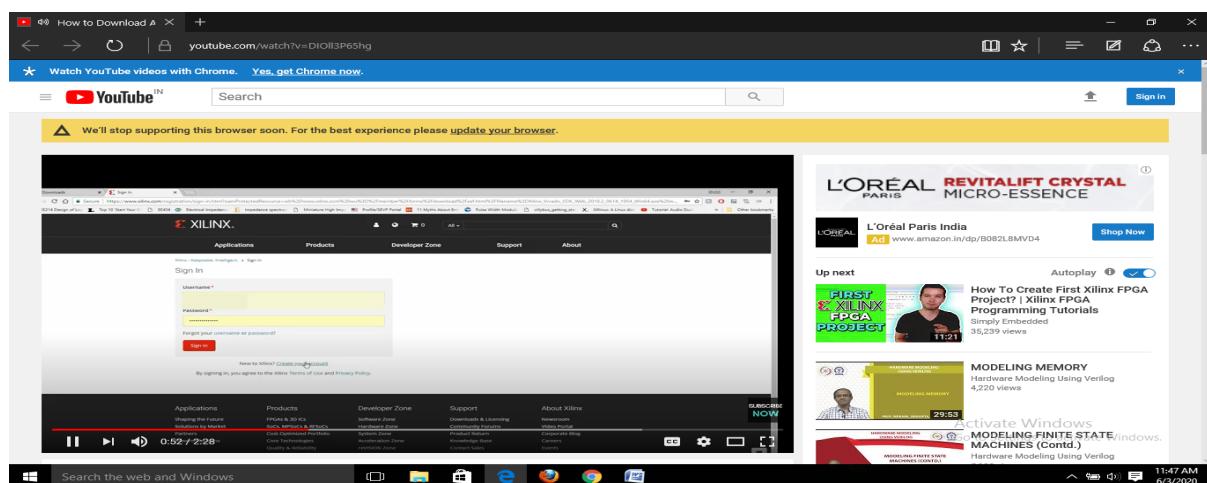
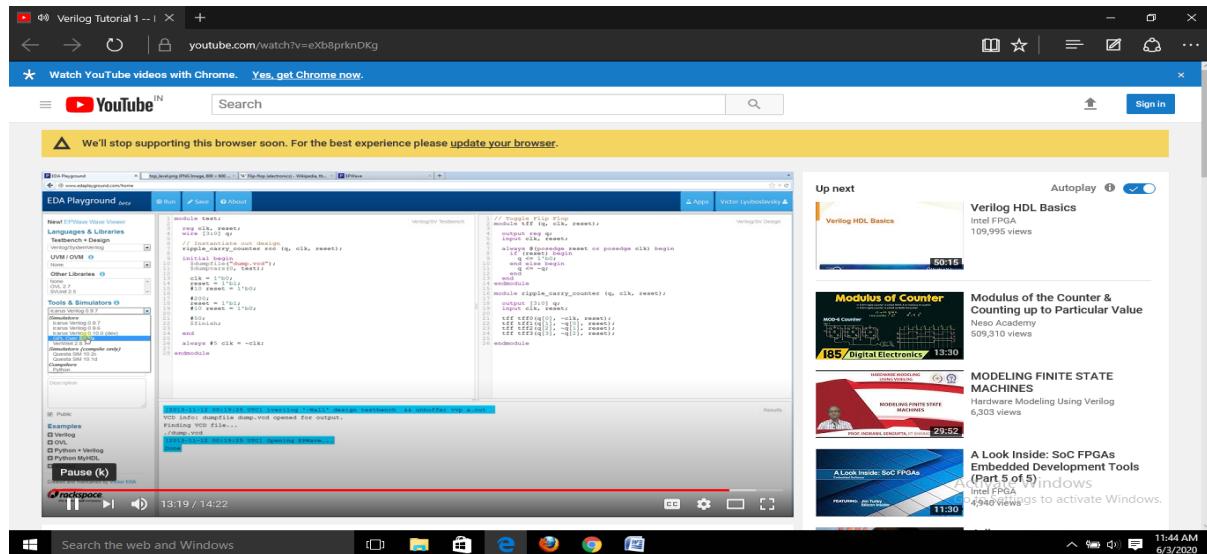
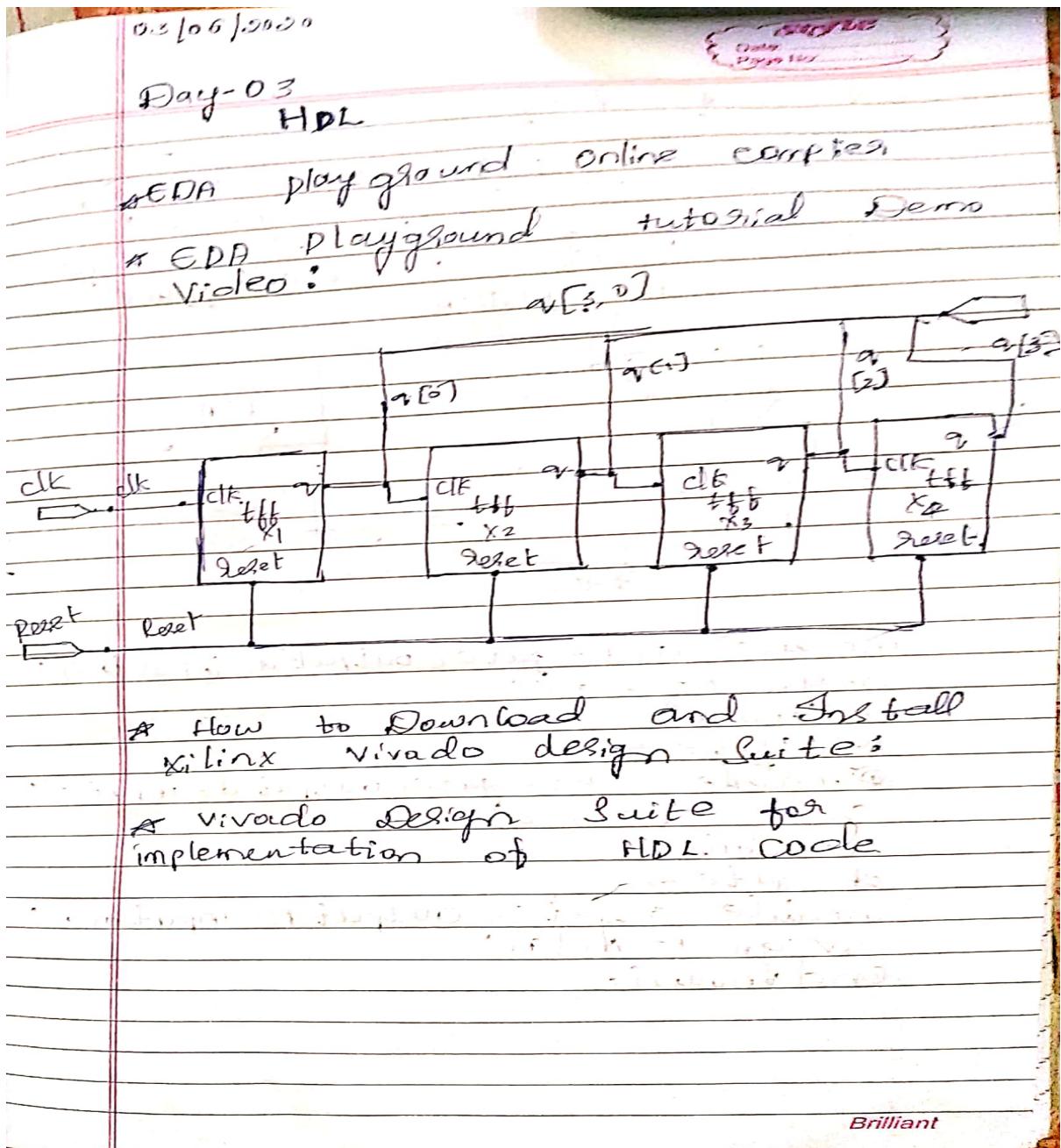
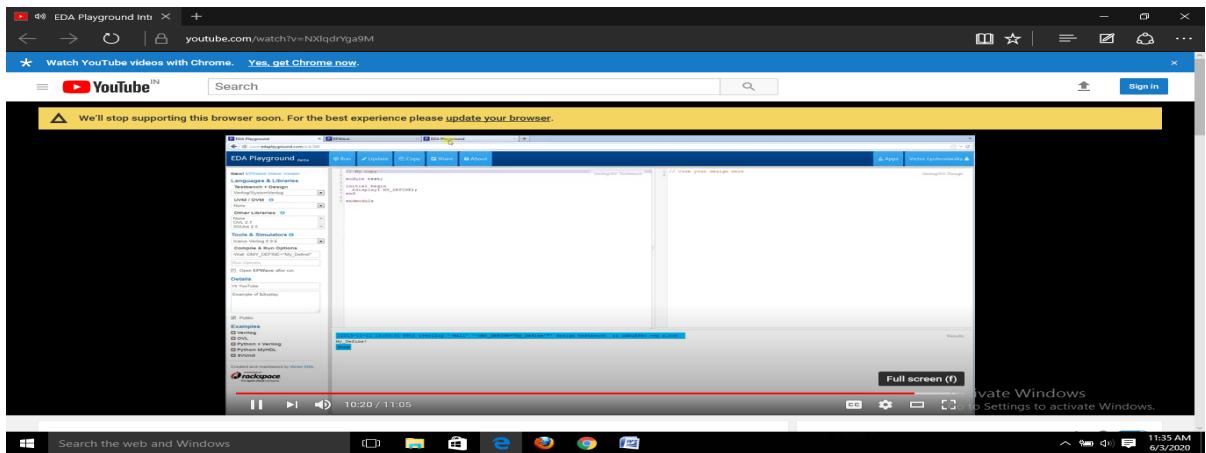


Date:	03-06-2020	Name:	Yamunashree N
Course:	HDL	USN:	4AL17EC097
Topic:	EDA Playground Online complier, EDA Playground Tutorial Demo Video, How to Download And Install Xilinx Vivado Design Suite, Vivado Design Suite for implementation of HDL code & Task Day-3: Implement 4 to 1 MUX using two 2 to 1 MUX using structural modelling style and test the module in online/offline compiler.	Semester and section	6 th sem 'B' section
Github repository:	yamunashree-course		



Edit with WPS Office



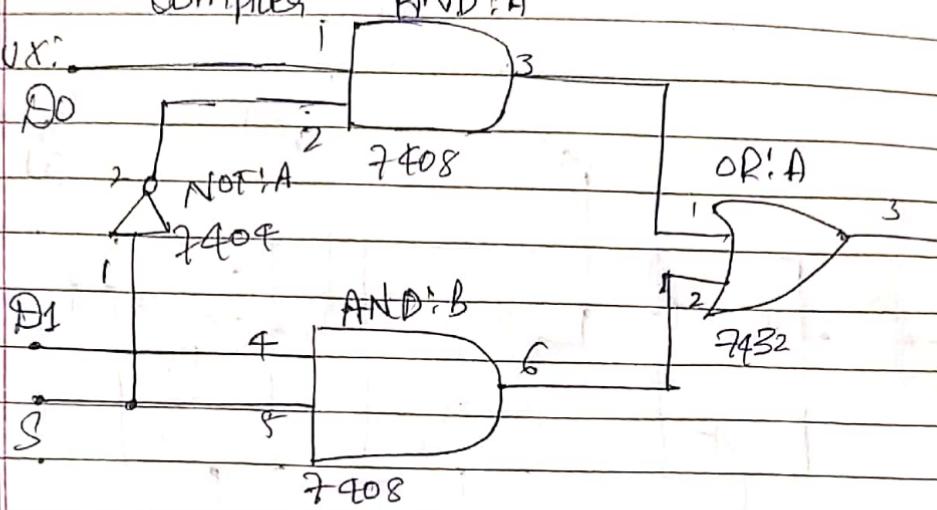
Edit with WPS Office

Tasks for Day 3

Implement 4 to 1 Mux using
2 to 1 Mux using structural
modelling style and test
the module in online/offline

Compiler AND:A

2:1 MUX:



```
module and_gate(output a, input b,c);  
assign a = b & c;  
endmodule
```

NOT gate \Rightarrow

```
@module not_gate(output d, input e);  
assign d = ~ e;  
endmodule
```

OR gate \Rightarrow

```
module orgate( output t, input m,n);  
assign t= m | n;  
endmodule
```

4.1 MUX :

module and_gate(output a, input b, c, d);

assign a = b & c & d;

endmodule

NOT gate =>

module not_gate(output f, input e);

assign e = ~f;

endmodule

OR-gate =>

module OR_gate(output l, input m, n, o, p);

assign l = m | n | o | p;

endmodule

Date:	03-06-2020	Name:	Yamunashree N
Course:	Python programming	USN:	4AL17EC097
Topic:	Application 8: scrape real estate property data from the web.	Semester and section:	6 th sem and B sec

The screenshot shows a browser window displaying a Udemy course titled "The Python Mega Course: Build 10 Real World Applications". The main content area shows a Jupyter notebook interface with two code cells. The first cell imports requests and beautifulsoup4, and makes a GET request to a real estate website. The second cell uses BeautifulSoup to parse the response and extract specific property details. To the right of the notebook, a sidebar titled "Course content" lists 24 video lectures, each with a thumbnail, title, and duration. Below the sidebar, sections for "Section 31: Application 9: Build a Web-based Financial Graph" and "Section 32: Application 10: Build a" are visible. At the bottom of the browser window, the Windows taskbar is visible with icons for search, file explorer, task manager, and other applications.



Edit with WPS Office

Request Headers

Note

Whenever I use this code in the next videos:

```
r = requests.get("http://www.pythonhow.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/")
```

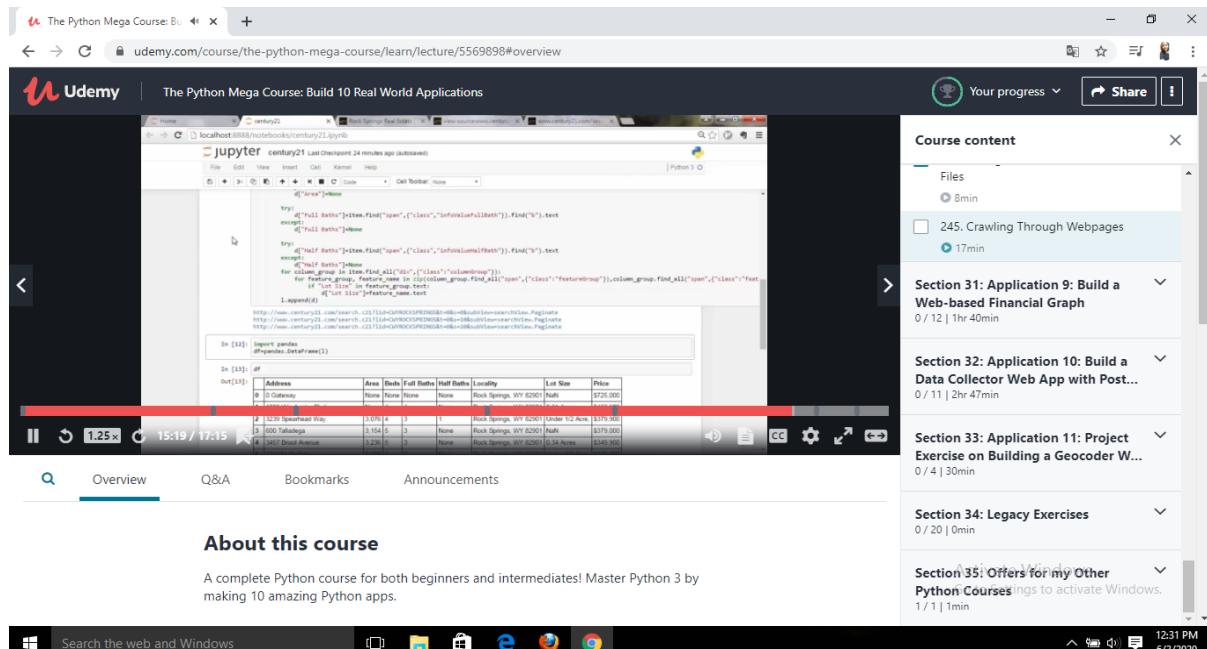
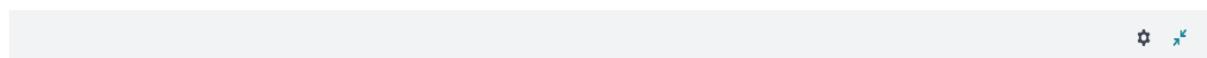
please use this instead:

```
r = requests.get("http://www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/", headers={'User-agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0'})
```

The rest of the code stays the same.

So, we're just changing the domain name from *pythonhow* to *pyclass* and we're adding a *header* argument. Some webpages don't like scripts sometimes, so adding a header allows the script to impersonate a web browser.

Activate Windows
Go to Settings to activate Windows.



The screenshot shows a browser window for the Udemy course 'The Python Mega Course: Build 10 Real World Applications'. The main content area displays a Jupyter notebook interface with several code cells and their outputs. One cell shows the creation of a DataFrame from a CSV file, and another shows a table of real estate data. The sidebar on the right lists course content sections, each with a thumbnail, title, and duration. The top of the browser window shows the Udemy logo and the course title.

About this course

A complete Python course for both beginners and intermediates! Master Python 3 by making 10 amazing Python apps.

Course content sections listed on the right:

- Section 31: Application 9: Build a Web-based Financial Graph
- Section 32: Application 10: Build a Data Collector Web App with Post...
- Section 33: Application 11: Project Exercise on Building a Geocoder W...
- Section 34: Legacy Exercises
- Section 35: Offers for my Other Python Courses



Edit with WPS Office

03/06/2020

Day - 15

Application 7: Scrape Real Estate Property Data from the web

- * Scrapped Website Data - How the output will look like
- * Request Headers
- * Loading the web page in Python
- * Extracting "div" tags
- * Extracting Address and property details
- * Extracting elements without unique identifiers
- * Saving the extracted data in CSV files
- * Drawing through webpages



Edit with WPS Office