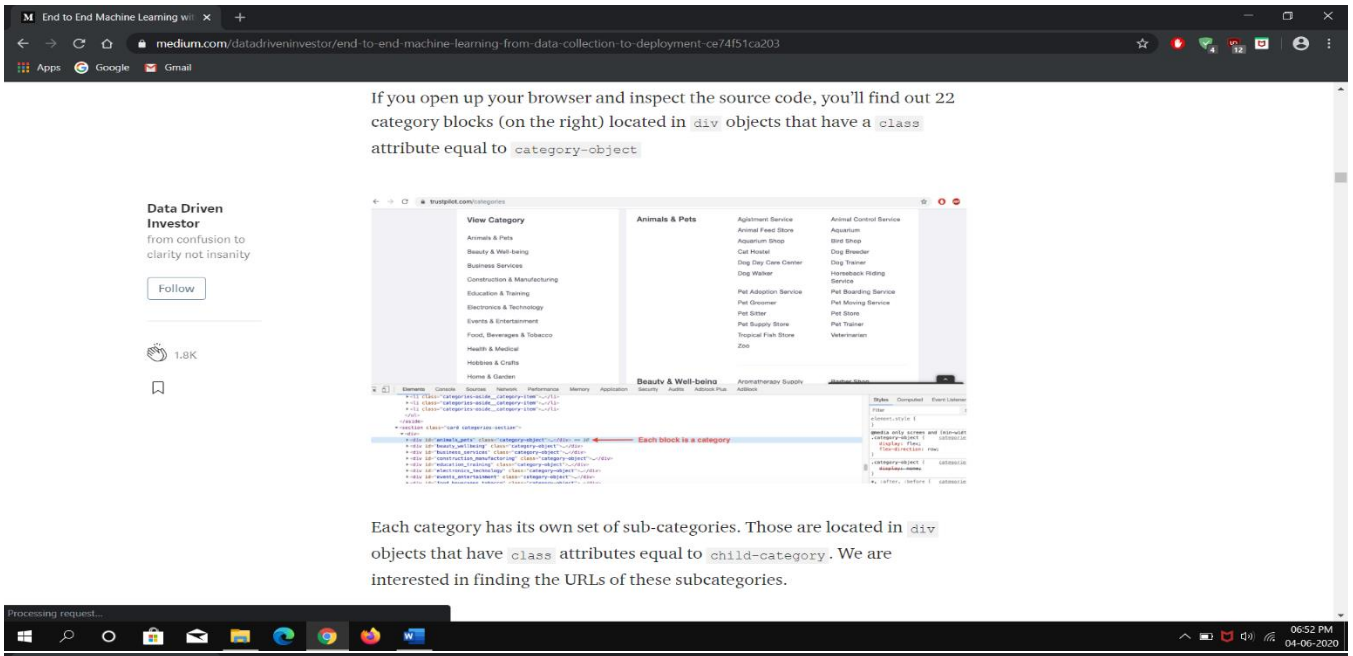
DAILY ASSESSMENT FORMAT

Date:	4th June 2020	**Name:**	Soundarya NA
Course:	UDEMY	**USN:**	4AL16EC077
Topic:	PYTHON: Application 10: Build a data collector web app with postgresql and flask	**Semester & Section:**	8th - B

FORENOON SESSION DETAILS

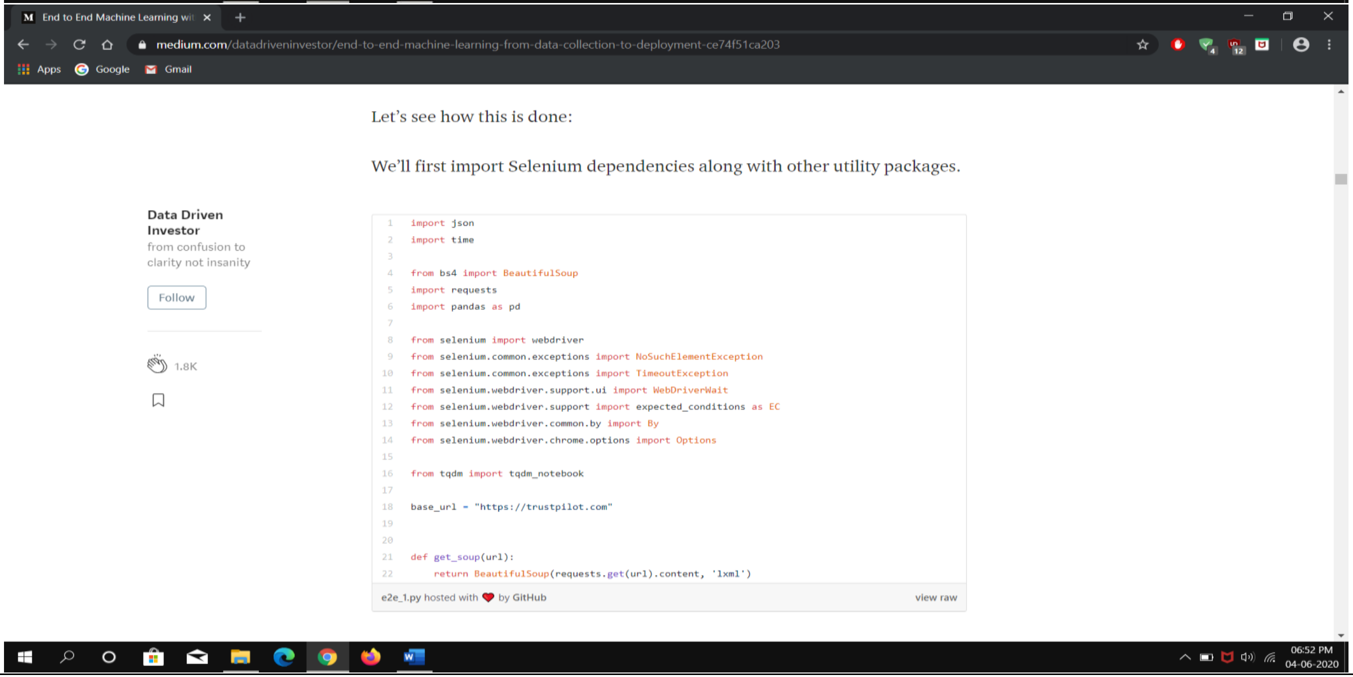
Image:



medium.com/datadriveninvestor/end-to-end-machine-learning-from-data-collection-to-deployment-ce74f51ca203

If you open up your browser and inspect the source code, you'll find out 22 category blocks (on the right) located in `div` objects that have a `class` attribute equal to `category-object`

Each category has its own set of sub-categories. Those are located in `div` objects that have `class` attributes equal to `child-category`. We are interested in finding the URLs of these subcategories.



medium.com/datadriveninvestor/end-to-end-machine-learning-from-data-collection-to-deployment-ce74f51ca203

Let's see how this is done:

We'll first import Selenium dependencies along with other utility packages.

```python
import json
import time

from bs4 import BeautifulSoup
import requests
import pandas as pd

from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import TimeoutException
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options

from tqdm import tqdm_notebook

base_url = "https://trustpilot.com"


def get_soup(url):
    return BeautifulSoup(requests.get(url).content, 'lxml')
```

e2e_1.py hosted with ❤ by GitHub view raw

Report:

Code:

```
basedir = os.path.abspath(os.path.dirname(__file__))

class Config(object):

    DEBUG = False

    TESTING = False

    CSRF_ENABLED = True

    SECRET_KEY = 'this-really-needs-to-be-changed'

    SQLALCHEMY_DATABASE_URI = os.environ['DATABASE_URL']

class ProductionConfig(Config):

    DEBUG = False

class StagingConfig(Config):

    DEVELOPMENT = True

    DEBUG = True

class DevelopmentConfig(Config):

    DEVELOPMENT = True

    DEBUG = True

class TestingConfig(Config):

    TESTING = True
```

Code:

```
from flask import Flask

from flask_sqlalchemy import SQLAlchemy

import os

app = Flask(__name__)

app.config.from_object(os.environ['APP_SETTINGS'])

app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

db = SQLAlchemy(app)

from models import Result

@app.route('/')

def hello():

    return "Hello World!"
```

```
@app.route('/<name>')
def hello_name(name):
    return "Hello {}!".format(name)
if __name__ == '__main__':
    app.run()
```

Data model:

```
from app import db
from sqlalchemy.dialects.postgresql import JSON
class Result(db.Model):
    __tablename__ = 'results'
    id = db.Column(db.Integer, primary_key=True)
    url = db.Column(db.String())
    result_all = db.Column(JSON)
    result_no_stop_words = db.Column(JSON)
    def __init__(self, url, result_all, result_no_stop_words):
        self.url = url
        self.result_all = result_all
        self.result_no_stop_words = result_no_stop_words
    def __repr__(self):
        return '<id {}>'.format(self.id)
```

Local Migration:

```
import os
from flask_script import Manager
from flask_migrate import Migrate, MigrateCommand
from app import app, db
app.config.from_object(os.environ['APP_SETTINGS'])
migrate = Migrate(app, db)
manager = Manager(app)
manager.add_command('db', MigrateCommand)
```

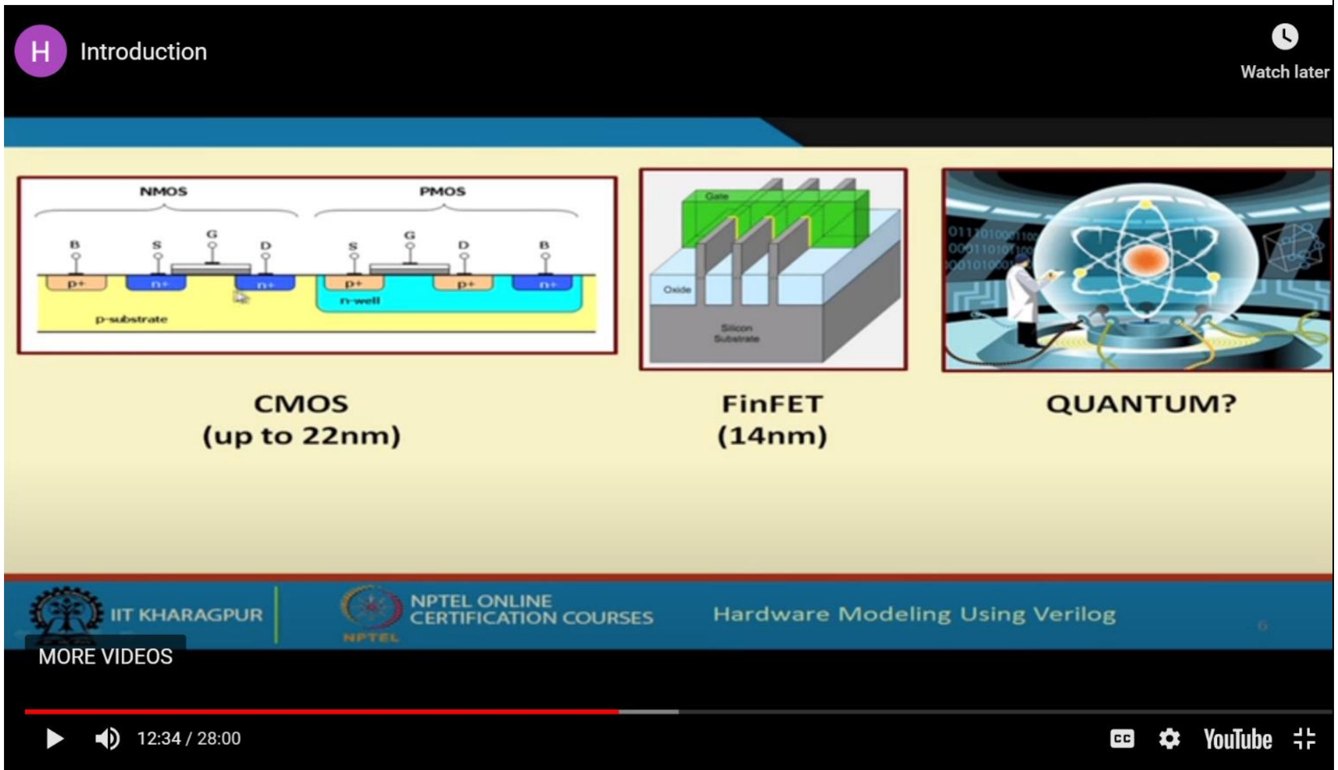
```
if __name__ == '__main__':
    manager.run()
```

SQL:

```
$ psql

# \c wordcount_dev

You are now connected to database "wordcount_dev" as user "michaelherman".

# \dt


              List of relations
 Schema |      Name       | Type  |     Owner
--------+-----------------+-------+---------------
 public | alembic_version | table | michaelherman
 public | results         | table | michaelherman
(2 rows)


# \d results
                          Table "public.results"
       Column        |       Type        |                      Modifiers
---------------------+-------------------+------------------------------------------------------
 id                  | integer           | not null default nextval('results_id_seq'::regclass)
 url                 | character varying |
 result_all          | json              |
 result_no_stop_words | json             |
Indexes:
    "results_pkey" PRIMARY KEY, btree (id)
```

Date:	4th June 2020	**Name:**	Soundarya NA
Course:	Digital Design using HDL	**USN:**	4AL16EC077
Topic:	Interview FAQ on digital system and HDL	**Semester & Section:**	8th - B

Image:

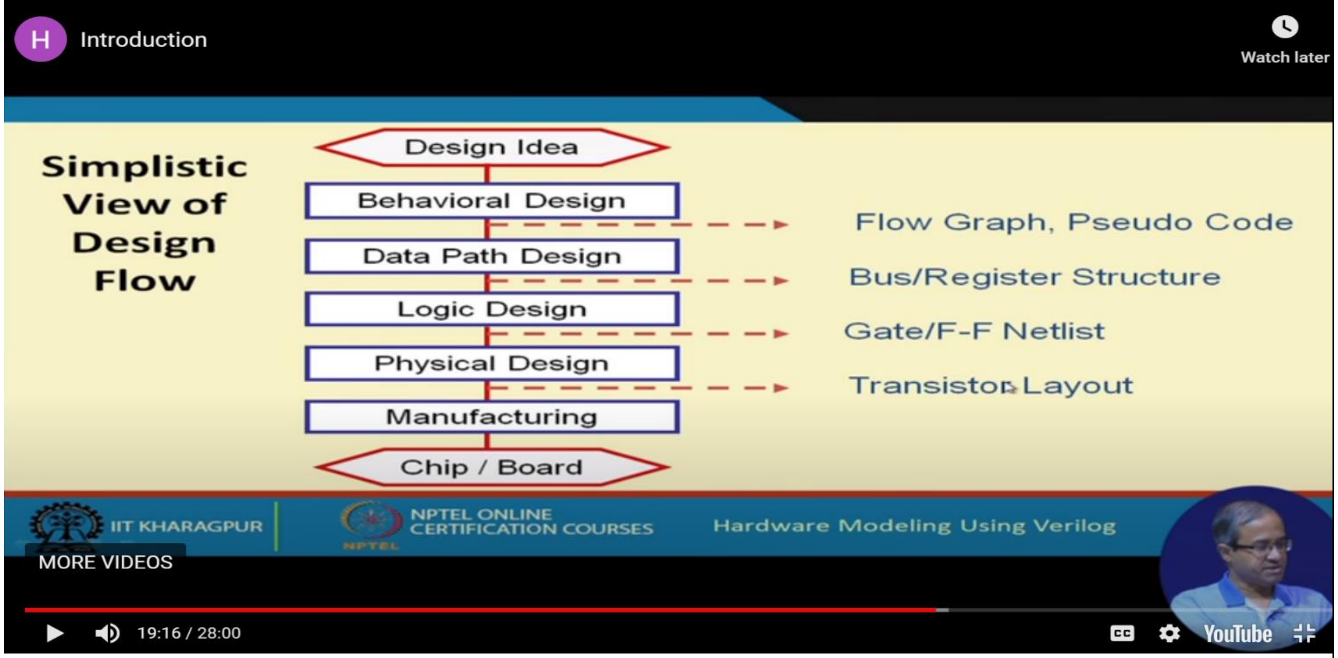


CMOS (up to 22nm)

FinFET (14nm)

QUANTUM?

Hardware Modeling Using Verilog



Simplistic View of Design Flow

Design Idea → Behavioral Design → Data Path Design → Logic Design → Physical Design → Manufacturing → Chip / Board

Behavioral Design → Flow Graph, Pseudo Code

Data Path Design → Bus/Register Structure

Logic Design → Gate/F-F Netlist

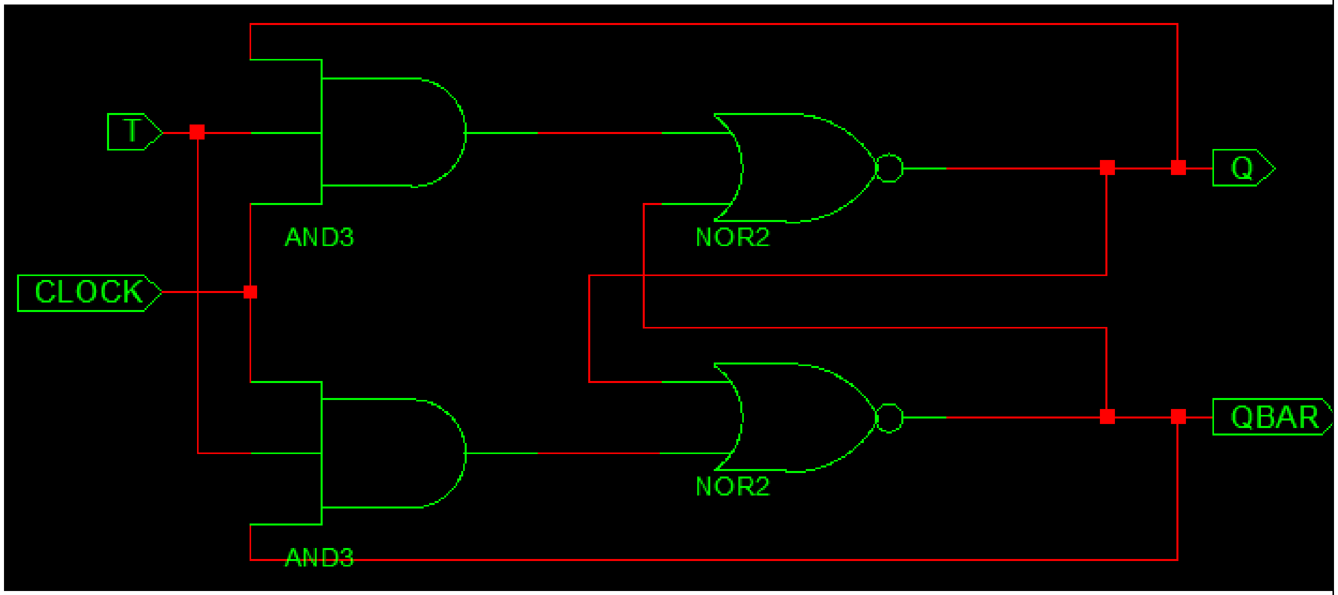
Physical Design → Transistor Layout

Hardware Modeling Using Verilog

Report:

Implement a simple T flipflop and test the module using a compiler:

The T flip-flop is a single input version of the JK flip-flop. As shown in figure, the T flip-flop is obtained from the JK type if both inputs are tied together. The output of the T flip-flop “toggles” with each clock pulse.



Truth table:

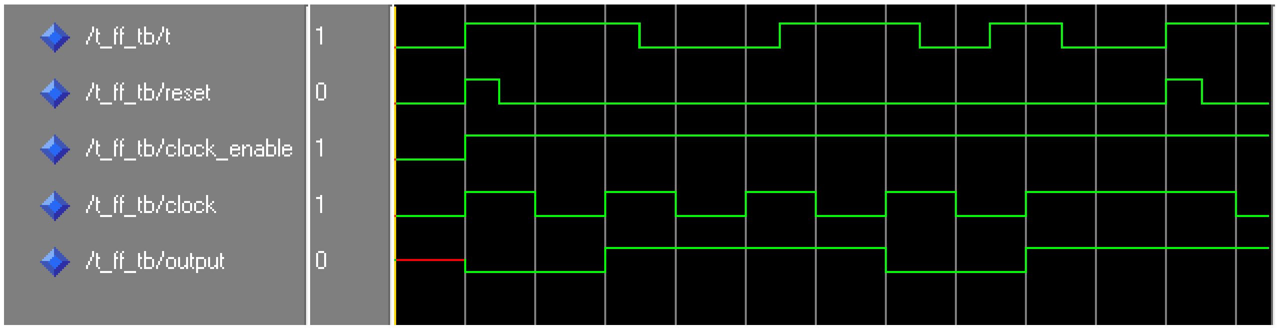
Q	T	Q(T+1)
0	0	0
0	1	1
1	0	1
1	1	0

Code:

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;


entity t_trigger is
```

```
  port (T,Reset,CLK,CLK_enable: in std_logic;
          Q: out std_logic);
end t_trigger;

architecture beh_t_trigger of t_trigger is

begin
   process (Reset,CLK)
   variable  temp: std_logic;
   begin
      if (rising_edge(CLK)) then    --sometimes you need to include a package for rising_edge, you can
use CLK'EVENT AND CLK = '1' instead
         if Reset='1' then
            temp := '0';
         elsif CLK_enable ='1' then
            temp := T xor temp;
         end if;
      end if;
   Q <= temp;
   end process;
end beh_t_trigger;
```

Simulated Results:


/t_ff_tb/t 1
/t_ff_tb/reset 0
/t_ff_tb/clock_enable 1
/t_ff_tb/clock 1
/t_ff_tb/output 0