IIA PROJECT- Milestone - II

BIO.SYNC

-Tanishq Tiwari 2021496
-Siddharth Anand 2021494
-Aniket Panchal 2021448

QUESTION 1. What data/open tools/software is required?

We are using Flask framework for our backend integration, HTML, CSS, Tailwind, etc., for frontend Since We are using APIs and a single CSV file, ETL tools are unnecessary per our approach. We will use a Python script to select and correctly display the integrated information per our global schema for data persistence.

QUESTION 2. Describe how/from which sources (for example, data.gov.in) you have acquired or simulated the data.

We are using 3 data sources: Pubchem and Chembl to acquire the data via API and drugbank data from a CSV file

ChEMBL API PubChemPy API are the APIs supported locally in Python language.

ChEMBL API: We can access the ChEMBL API directly by sending HTTP requests to its endpoints. The ChEMBL API provides information about the drug and its properties.

```python
def get_drug_info_chembl(name):
    print('in names_chembl')
    try:
        from chembl_webresource_client.new_client import new_client
        drug_name = name
        molecule = new_client.molecule
        mols = molecule.filter(molecule_synonyms__molecule_synonym__iexact=drug_name).only('molecule_chembl_id')
        for mol in mols:
            print(f"ChEMBL ID for {drug_name}: {mol['molecule_chembl_id']}")
        chembl_id = mol['molecule_chembl_id']
        drug_info = molecule.get(chembl_id)
        print("Drug Information for ChEMBL ID:", chembl_id)
        return drug_info
    except Exception as e:
        return "Error: " + str(e)
```

For the Drug “Sildenafil”
Here the result of the the API query

Drug Information from ChEMBL

ChEMBL ID: CHEMBL192

Name: SILDENAFIL

ATC Classifications: G04BE03

Availability Type: 1

Biotherapeutic: None

Black Box Warning: 0

ChEBI Par ID: 9139

Chemical Probe: 0

Chirality: 2

Cross References:

- XRef ID: sildenafil%20citrate, XRef Name: sildenafil citrate, XRef Src: DailyMed
- XRef ID: 26748898, XRef Name: SID: 26748898, XRef Src: PubChem
- XRef ID: 50085897, XRef Name: SID: 50085897, XRef Src: PubChem

Dosed Ingredient: True

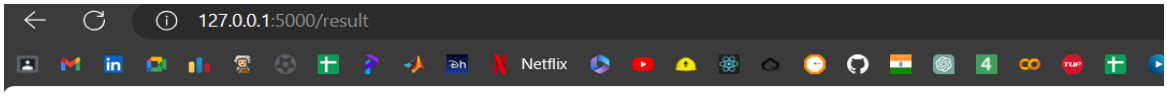
First Approval: 1998

First In Class: 0

Helm Notation: None

Indication Class: Impotence Therapy

Inorganic Flag: 0



Inorganic Flag: 0

Max Phase: 4.0

Molecule Type: Small molecule

Natural Product: 1

Oral: True

Parenteral: True

Polymer Flag: 0

Prodrug: 0

Structure Type: MOL

Therapeutic Flag: True

Topical: False

USAN Stem: -afil

USAN Stem Definition: phosphodiesterase type 5 inhibitors (PDE5) inhibitors: (PDE5) inhibitors containing a sulfonamide moiety

USAN Substem: -afil (denafil)

USAN Year: 1997

Withdrawn Flag: False

Molecule Properties:

- ALogP: 1.61
- Aromatic Rings: 3
- CX LogD: 1.16
- CX LogP: 1.23
- CX Most pKa: 7.63
- CX Most pKa: 5.98
- Full Molformula: C22H30N6O4S
- Full MWt: 474.59
- HBA: 8
- HBA Lipinski: 10
- HBD: 1
- HBD Lipinski: 1
- Heavy Atoms: 33
- Molecular Species: NEUTRAL
- MW Freebase: 474.59
- MW Monoisotopic: 474.2049
- NP Likeness Score: -1.51
- Num Lipinski RO5 Violations: 0
- Num RO5 Violations: 0
- PSA: 113.42
- QED Weighted: 0.55
- RO3 Pass: N
- RTB: 7

Molecule Structures:

- Canonical SMILES: CCCc1nn(C)c2c(=O)[nH]c(-c3cc(S(=O)(=O)N4CCN(C)CC4)ccc3OCC)nc12
- MOLFile: (RDKit 2D structure data)
- Standard InChI: InChI=1S/C22H30N6O4S/c1-5-7-17-19-20(27(4)25-17)22(29)24-21(23-19)16-14-15(8-9-18(16)32-6-2)33(30,31)28-12-10-26(3)11-13-28/h8-9,14H,5-7,10-13H2,1-4H3,(H,23,24,29)
- Standard InChI Key: BNRNXUUZRGQAQC-UHFFFAOYSA-N

Molecule Synonyms:

- Synonym: Aphrodil, Synonym Type: OTHER
- Synonym: HIP0908, Synonym Type: RESEARCH_CODE
- Synonym: HIP-0908, Synonym Type: RESEARCH_CODE
- Synonym: Nipatra, Synonym Type: TRADE_NAME
- Synonym: Patrex, Synonym Type: OTHER
- Synonym: Revatio, Synonym Type: TRADE_NAME
- Synonym: Sildenafil, Synonym Type: FDA
- Synonym: Sildenafil, Synonym Type: ATC
- Synonym: Sildenafil, Synonym Type: BAN
- Synonym: Sildenafil, Synonym Type: BNF
- Synonym: Sildenafil, Synonym Type: INN
- Synonym: Sildenafil, Synonym Type: MERCK_INDEX
- Synonym: Sildenafil, Synonym Type: OTHER
- Synonym: Sildenafil actavis, Synonym Type: OTHER
- Synonym: Sildenafil ratiopharm, Synonym Type: OTHER
- Synonym: Sildenafil teva, Synonym Type: OTHER
- Synonym: UK-92480, Synonym Type: RESEARCH_CODE
- Synonym: Viagra, Synonym Type: TRADE_NAME
- Synonym: Vizarsin, Synonym Type: TRADE_NAME

PubChemPy API: Installing the PubChemPy Python library (pip install PubChemPy) and using it to access PubChem data programmatically. PubChem provides information on chemical compounds, their structures, and properties.

```python
def get_drug_info_pubchem(name):
    print("in pubchem")
    try:
        compounds = pcp.get_compounds(name, 'name')
        if not compounds:
            print(f"No compound found for '{name}'")
            return
        drug_info_list = []
        for compound in compounds:
            compound_dict = compound.to_dict()
            drug_info_list.append(compound_dict)
        return drug_info_list
    except Exception as e:
        print("Error:", str(e))
```

For the same drug “Sildenafil”

Drug Information from Pubchem

ATOMS STEREO COUNT: 0

ATOMS:

- aid: 1, number: 16, element: S, y: -0.5, x: 5.4641
- aid: 2, number: 8, element: O, y: 0.366, x: 4.9641
- aid: 3, number: 8, element: O, y: -1.366, x: 5.9641
- aid: 4, number: 8, element: O, y: 1.5, x: 8.9282
- aid: 5, number: 8, element: O, y: -3, x: 9.7942
- aid: 6, number: 7, element: N, y: -1, x: 4.5981
- aid: 7, number: 7, element: N, y: -2, x: 2.866
- aid: 8, number: 7, element: N, y: -1.5, x: 8.9282
- aid: 9, number: 7, element: N, y: -1.8047, x: 11.6065
- aid: 10, number: 7, element: N, y: 0, x: 9.7942
- aid: 11, number: 7, element: N, y: -1, x: 12.1901
- aid: 12, number: 6, element: C, y: -0.5, x: 3.732
- aid: 13, number: 6, element: C, y: -2, x: 4.5981
- aid: 14, number: 6, element: C, y: -1, x: 2.866
- aid: 15, number: 6, element: C, y: -2.5, x: 3.732
- aid: 16, number: 6, element: C, y: 0, x: 6.3301
- aid: 17, number: 6, element: C, y: -2.5, x: 2
- aid: 18, number: 6, element: C, y: -0.5, x: 7.1962
- aid: 19, number: 6, element: C, y: 0, x: 8.0622
- aid: 20, number: 6, element: C, y: 1, x: 6.3301
- aid: 21, number: 6, element: C, y: 1, x: 8.0622
- aid: 22, number: 6, element: C, y: -0.5, x: 8.9282
- aid: 23, number: 6, element: C, y: 1.5, x: 7.1962
- aid: 24, number: 6, element: C, y: -0.5, x: 10.6603
- aid: 25, number: 6, element: C, y: -1.5, x: 10.6603
- aid: 26, number: 6, element: C, y: -0.1953, x: 11.6065
- aid: 27, number: 6, element: C, y: 0.7553, x: 11.9171
- aid: 28, number: 6, element: C, y: -2, x: 9.7942
- aid: 29, number: 6, element: C, y: 0.9615, x: 12.8956

cactvs_fingerprint:
1110000001111011101110000000000000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
over

canonical_smiles: CCCC1=NN(C2=C1N=C(NC2=O)C3=C(C=CC(=C3)S(=O)(=O)N4CCN(CC4)C)OCC)C

charge: 0

cid: 135398744

complexity: 838

conformer_id_3d: None

conformer_rmsd_3d: None

coordinate_type: 2d

covalent_unit_count: 1

defined_atom_stereo_count: 0

defined_bond_stereo_count: 0

effective_rotor_count_3d: None

exact_mass: 474.20492463

feature_selfoverlap_3d: None

fingerprint:
00000371E07BB800400000000000000000000000001600000003C408000000000004001C000001E04184000000C0CE1DE0633C793C80402AA0327727470D

h_bond_acceptor_count: 8

h_bond_donor_count: 1

inchi: InChI=1S/C22H30N6O4S/c1-5-7-17-19-20(27(4)25-17)22(29)24-21(23-19)16-14-15(8-9-18(16)32-6-2)33(30,31)28-12-10-26(3)11-13-28/h8-9,14H,5-7,10

inchikey: BNRNXUUZRGQAQC-UHFFFAOYSA-N

These are just examples of snippets of the data acquired and we displayed them on webpage. The correct schemas of these local data sources are described in the next part with more data.

QUESTION 3. Clearly define and state your global and local schemas over these data sources/data stores created above (point 2). How will you access the data from these local data sources, whether through APIs or querying these data sources? If APIs, what are these APIs supported on the local data sources? How will you communicate through APIs from a global source (virtual/materialised information integration) to these local sources? If the data access is through sub-query execution at local sources, how will communication occur between global and local sources?
If you are federating and running a search query on the local sources, define the search criteria clearly with constraints, if any.

Local Schema of Pubchem

-- Table to store compound information

```
pubchem_attributes = [
    "cactvs_fingerprint",
    "canonical_smiles",
    "charge",
    "cid",
    "complexity",
    "conformer_id_3d",
    "conformer_rmsd_3d",
    "coordinate_type",
    "covalent_unit_count",
    "defined_atom_stereo_count",
    "defined_bond_stereo_count",
    "effective_rotor_count_3d",
    "exact_mass",
    "feature_selfoverlap_3d",
    "fingerprint",
    "h_bond_acceptor_count",
    "h_bond_donor_count",
    "inchi",
    "inchikey",
    "isomeric_smiles",
    "isotope_atom_count",
    "iupac_name",
    "mmff94_energy_3d",
    "mmff94_partial_charges_3d",
    "molecular_formula",
    "molecular_weight",
    "monoisotopic_mass",
    "multipoles_3d",
    "pharmacophore_features_3d",
    'charge',
    "rotatable_bond_count",
    "shape_fingerprint_3d",
    "shape_selfoverlap_3d",
    "tpsa",
    "undefined_atom_stereo_count",
    "undefined_bond_stereo_count",
    "volume_3d",
    "xlogp"
]
```

Local Schema of chembl

-- Create a table to store drug information

```
chembl_attributes = [
    "chembl_id",
    "pref_name",
    "atc_classifications",
```

"availability_type",
"biotherapeutic",
"black_box_warning",
"chebi_par_id",
"chemical_probe",
"chirality",
"cross_references",
"dosed_ingredient",
"first_approval",
"first_in_class",
"helm_notation",
"indication_class",
"inorganic_flag",
"max_phase",
"molecule_chembl_id",
"molecule_hierarchy",
"molecule_type",
"natural_product",
"oral",
"parenteral",
"polymer_flag",
"prodrug",
"structure_type",
"therapeutic_flag",
"topical",
"usan_stem",
"usan_stem_definition",
"usan_substem",
"usan_year",
"withdrawn_flag",

"molecule_synonym",
"syn_type",

"alogp",
"aromatic_rings",
"cx_logd",
"cx_logp",
"cx_most_pKa1",
"cx_most_pKa2",
"full_molformula",
"full_mwt",
"hba",
"hba_lipinski",
"hbd",
"hbd_lipinski",
"heavy_atoms",
"molecular_species",
"mw_freebase",
"mw_monoisotopic",
"np_likeness_score",
"num_lipinski_ro5_violations",
"num_ro5_violations",
"psa",
"qed_weighted",

```
    "ro3_pass",
    "rtb",

    "canonical_smiles",
    "molfile",
    "standard_inchi",
    "standard_inchi_key",


    # "xref_name"
    # "xref_src"
]
```

Local Schema of drugbank

```
drugbank_attributes = [
    "compound",
    "description",
    "state",
    "indication",
    "pharmacodynamics",
    "mechanism_of_action",
    "toxicity",
    "metabolism",
    "absorption",
    "half_life",
    "protein_binding",
    "route_of_elimination",
    "volume_of_distribution",
    "clearance"
]
```

Global Schema

```
global_attributes = [
    "drug_id", "name", "atc_classifications", "availability_type",
    "biotherapeutic", "black_box_warning", "chebi_par_id", "chemical_probe",
    "chirality", "cross_references", "dosed_ingredient", "first_approval",
    "first_in_class", "helm_notation", "indication_class", "inorganic_flag",
    "max_phase", "molecule_chembl_id", "molecule_hierarchy", "molecule_type",
    "natural_product", "oral", "parenteral", "polymer_flag", "prodrug",
    "structure_type", "therapeutic_flag", "topical", "usan_stem",
    "usan_stem_definition", "usan_substem", "usan_year", "withdrawn_flag",
    "molecule_synonym", "syn_type",  "alogp", "aromatic_rings",
    "cx_logd", "cx_logp", "cx_most_pka1", "cx_most_pka2", "full_molformula", "full_mwt",
    "hba", "hba_lipinski", "hbd", "hbd_lipinski", "heavy_atoms",
    "molecular_species", "mw_freebase", "mw_monoisotopic", "np_likeness_score",
    "num_lipinski_ro5_violations", "num_ro5_violations", "psa", "qed_weighted",
    "ro3_pass", "rtb", "rotatable_bond_count", "fingerprint", "fingerprint_type",
    "canonical_smiles", "isomeric_smiles", "molfile", "standard_inchi", "standard_inchi_key",
    "cactvs_fingerprint", "cid", "complexity", "conformer_id_3d", "conformer_rmsd_3d", "coordinate_type", "covalent_unit_count",
    "defined_atom_stereo_count", "defined_bond_stereo_count", "effective_rotor_count_3d", "feature_selfoverlap_3d", "fingerprint",
    "isotope_atom_count", "iupac_name", "mmff94_energy_3d", "mmff94_partial_charges_3d", "multipoles_3d", "pharmacophore_features_3d",
    'charge', "rotatable_bond_count", "shape_fingerprint_3d", "shape_selfoverlap_3d", "tpsa", "undefined_atom_stereo_count",
    "undefined_bond_stereo_count", "volume_3d", "xlogp" , "compound","description","state","indication","pharmacodynamics",
    "mechanism_of_action","toxicity","metabolism","absorption","half_life" "protein_binding","route_of_elimination",
    "volume_of_distribution","clearance"
]
```

This is the ENTITY MAPPING

```
global_mapping = {
    'drug_id': ['chembl_id', 'compound_id', None],
    'name': ['pref_name', 'name', None],
    'atc_classifications': ['atc_classifications', None, None],
    'availability_type': ['availability_type', None, None],
    'biotherapeutic': ['biotherapeutic', None, None],
    'black_box_warning': ['black_box_warning', None, None],
    'chebi_par_id': ['chebi_par_id', None, None],
    'chemical_probe': ['chemical_probe', None, None],
    'chirality': ['chirality', None, None],
    'cross_references': ['cross_references', None, None],
    'dosed_ingredient': ['dosed_ingredient', None, None],
    'first_approval': ['first_approval', None, None],
    'first_in_class': ['first_in_class', None, None],
    'helm_notation': ['helm_notation', None, None],
    'indication_class': ['indication_class', None, None],
    'inorganic_flag': ['inorganic_flag', None, None],
    'max_phase': ['max_phase', None, None],
    'molecule_chembl_id': ['molecule_chembl_id', None, None],
    'molecule_hierarchy': ['molecule_hierarchy', None, None],
    'molecule_type': ['molecule_type', None, None],
    'natural_product': ['natural_product', None, None],
    'oral': ['oral', None, None],
    'parenteral': ['parenteral', None, None],
    'polymer_flag': ['polymer_flag', None, None],
    'prodrug': ['prodrug', None, None],
    'structure_type': ['structure_type', None, None],
    'therapeutic_flag': ['therapeutic_flag', None, None],
    'topical': ['topical', None, None],
    'usan_stem': ['usan_stem', None, None],
    'usan_stem_definition': ['usan_stem_definition', None, None],
    'usan_substem': ['usan_substem', None, None],
    'usan_year': ['usan_year', None, None],
    'withdrawn_flag': ['withdrawn_flag', None, None],
    'molecule_synonym': ['molecule_synonym', None, None],
    'syn_type': ['syn_type', None, None],
    'alogp': ['alogp', None, None],
    'aromatic_rings': ['aromatic_rings', None, None],
    'cx_logd': ['cx_logd', None, None],
    'cx_logp': ['cx_logp', None, None],
    'cx_most_pka1': ['cx_most_pka1', None, None],
    'cx_most_pka2': ['cx_most_pka2', None, None],
    'full_molformula': ['full_molformula', 'formula', None],
    'full_mwt': ['full_mwt', 'molecular_weight', None],
    'hba': ['hba', 'h_bond_acceptor_count', None],
    'hba_lipinski': ['hba_lipinski', None, None],
    'hbd': ['hbd', 'h_bond_donor_count', None],
    'hbd_lipinski': ['hbd_lipinski', None, None],
    'heavy_atoms': ['heavy_atoms', 'heavy_atom_count', None],
```

'molecular_species': ['molecular_species', None, None],
'mw_freebase': ['mw_freebase', None, None],
'mw_monoisotopic': ['mw_monoisotopic', None, None],
'np_likeness_score': ['np_likeness_score', None, None],
'num_lipinski_ro5_violations': ['num_lipinski_ro5_violations', None, None],
'num_ro5_violations': ['num_ro5_violations', None, None],
'psa': ['psa', None, None],
'qed_weighted': ['qed_weighted', None, None],
'ro3_pass': ['ro3_pass', None, None],
'rtb': ['rtb', None, None],
'canonical_smiles': ['canonical_smiles', 'canonical_smiles', None],
'isomeric_smiles': [None, 'isomeric_smiles', None],
'molfile': ['molfile', None, None],
'standard_inchi': ['standard_inchi', 'inchi', None],
'standard_inchi_key': ['standard_inchi_key', 'inchikey', None],
'cactvs_fingerprint': [None, 'cactvs_fingerprint', None],
'cid': [None, 'cid', None],
'complexity': [None, 'complexity', None],
'conformer_id_3d': [None, 'conformer_id_3d', None],
'conformer_rmsd_3d': [None, 'conformer_rmsd_3d', None],
'coordinate_type': [None, 'coordinate_type', None],
'covalent_unit_count': [None, 'covalent_unit_count', None],
'defined_atom_stereo_count': [None, 'defined_atom_stereo_count', None],
'defined_bond_stereo_count': [None, 'defined_bond_stereo_count', None],
'effective_rotor_count_3d': [None, 'effective_rotor_count_3d', None],
'feature_selfoverlap_3d': [None, 'feature_selfoverlap_3d', None],
'fingerprint': [None, 'fingerprint', None],
'isotope_atom_count': [None, 'isotope_atom_count', None],
'iupac_name': [None, 'iupac_name', None],
'mmff94_energy_3d': [None, 'mmff94_energy_3d', None],
'mmff94_partial_charges_3d': [None, 'mmff94_partial_charges_3d', None],
'multipoles_3d': [None, 'multipoles_3d', None],
'pharmacophore_features_3d': [None, 'pharmacophore_features_3d', None],
'charge': [None, 'charge', None],
'rotatable_bond_count': [None, 'rotatable_bond_count', None],
'shape_fingerprint_3d': [None, 'shape_fingerprint_3d', None],
'shape_selfoverlap_3d': [None, 'shape_selfoverlap_3d', None],
'tpsa': [None, 'tpsa', None],
'undefined_atom_stereo_count': [None, 'undefined_atom_stereo_count', None],
'undefined_bond_stereo_count': [None, 'undefined_bond_stereo_count', None],
'volume_3d': [None, 'volume_3d', None],
'xlogp': [None, 'xlogp', None],

'compound': [None, None, 'compound'],
'description': [None, None, 'description'],
'state': [None, None, 'state'],
'indication': [None, None, 'indication'],
'pharmacodynamics': [None, None, 'pharmacodynamics'],
'mechanism_of_action': [None, None, 'mechanism_of_action'],
'toxicity': [None, None, 'toxicity'],
'metabolism': [None, None, 'metabolism'],
'absorption': [None, None, 'absorption'],
'half_life': [None, None, 'half_life'],
'protein_binding': [None, None, 'protein_binding'],
'route_of_elimination': [None, None, 'route_of_elimination'],

```
    'volume_of_distribution': [None, None, 'volume_of_distribution'],
    'clearance': [None, None, 'clearance']

}
```

The local data (pubchem and chembyl) sources have been extracted by using APIs and

VIRUALISED VIA APIs

1) Pubchem API -> PUBCHEMPY package in python.
2) Chembl - > Chemblwebresourceclient package supported in python

MATERIALISED ON OUR LOCAL SYSTEM

3) Drug Bank -> from the CSV file.

Below is the stepwise workflow for your solution:

- Fetching the data via APIs.
- Storing the fetched data temporarily in the local schema since there, in this case, the local schema is a Python dictionary where the Keyvalue is the attribute of that data and values are the corresponding data of that attribute/column of our local schema.
- We made a global schema and its entity mapping, For the global schema per all the local schemas, the format of the mapping is mentioned above; the value of the global data dictionary consists of a list where the 1st attribute represents the attributes of Chembl schema (local schema) while the 2nd attribute represents the corresponding attribute of Pubchem (local schema), 3rd attribute represents the attribute mapping with drugbank schema(local schema).
- Now we will fill the data to the global schema by python script sub-querying, this is where we will communicate between local and global schema by sub-querying.

```
for mapping, (chembl_key, pubchem_key , drugbank_key) in global_mapping.items():
    if chembl_key is None:
        global_values[mapping] = pubchem_values[pubchem_key]
    elif pubchem_key is None:
        global_values[mapping] = chembl_values[chembl_key]
    elif chembl_key is None and pubchem_key is None:
        global_values[mapping] = drugbank_values[drugbank_key]
    else:
        global_values[mapping] = chembl_values[chembl_key]
```

- Then we will display the values from the GLOBAL schema of Flask webpage. For example (tentative for now):

```
<h1>Drug Information from GlobaL schema</h1>


<p>Drug ID: {{ global_values['drug_id'] }}</p>
<p>Name: {{ global_values['name'] }}</p>
<p>ATC Classifications: {{ global_values['atc_classifications'] }}</p>
<p>Availability Type: {{ global_values['availability_type'] }}</p>
<p>Biotherapeutic: {{ global_values['biotherapeutic'] }}</p>
<p>Black Box Warning: {{ global_values['black_box_warning'] }}</p>
<p>CHEBI PAR ID: {{ global_values['chebi_par_id'] }}</p>
<p>Chemical Probe: {{ global_values['chemical_probe'] }}</p>
<p>Chirality: {{ global_values['chirality'] }}</p>
<p>Cross References: {{ global_values['cross_references'] }}</p>
<p>Dosed Ingredient: {{ global_values['dosed_ingredient'] }}</p>
```

- We can search the data on the external databases, which are our local data sources, via drug name, chembl _id , pubchem_id, or smiles(canonical smiles) of a drug; these will be our search constraints if we want to search for any drug.