**COMP 851 Final Project**

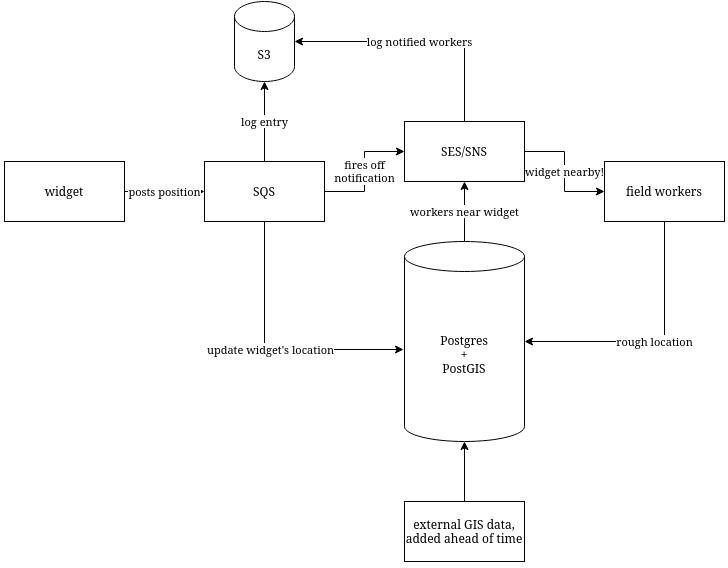
**Ullas Uppalanchi**

GitHub Link: <https://github.com/ullas22/COMP851-final-project.git>

**Project Topic: Data Stream 2**

The second integration has to do with a product. The PTWC Widgets will be deployed into the field and communicate their GPS position. In order to prepare field operations, we will need to establish a database which can determine the proximity of Widgets to county and township locations where field operators may be stationed, or sent. In order to do this, we must deploy a GIS database, called PostGIS and ingest the city lat / long positions. In addition, we must notify and record the ingest of these positions in preparation for the location of the field operators and Widget positions. We must do so using the AWS SQS, and SNS/SES interfaces in order to send the notifications and emails, and finally deposit log entries on s3.

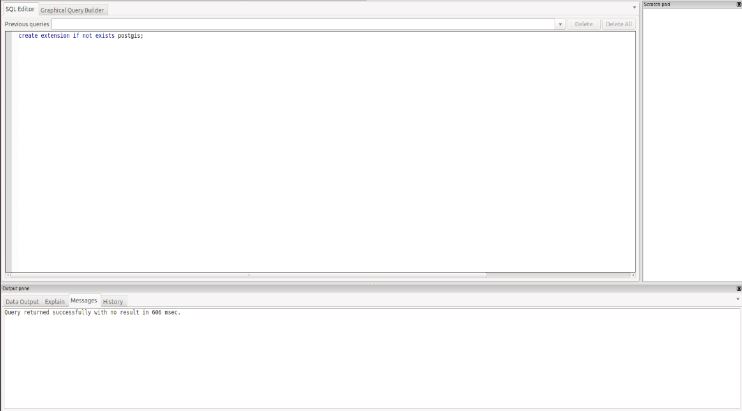
**Design:**



**Implementing using PostGres GUI:**

**Step 1: create PostGIS extension**

create extension if no exists postgis;



**Step 2: Creating the table landmarks and index landmarks\_the\_geom\_gist**

CREATE TABLE landmarks

(

gid serial NOT NULL,

name character varying(50),

address character varying(50),

date\_built character varying(10),

architect character varying(50),

landmark character varying(10),

latitude double precision,

longitude double precision,

the\_geom geometry,

CONSTRAINT landmarks\_pkey PRIMARY KEY (gid),

CONSTRAINT enforce\_dims\_the\_geom CHECK (st\_ndims(the\_geom) = 2),

CONSTRAINT enforce\_geotype\_geom CHECK (geometrytype(the\_geom) = 'POINT'::text OR the\_geom IS NULL),

CONSTRAINT enforce\_srid\_the\_geom CHECK (st\_srid(the\_geom) = 4326)

);

CREATE INDEX landmarks\_the\_geom\_gist

ON landmarks

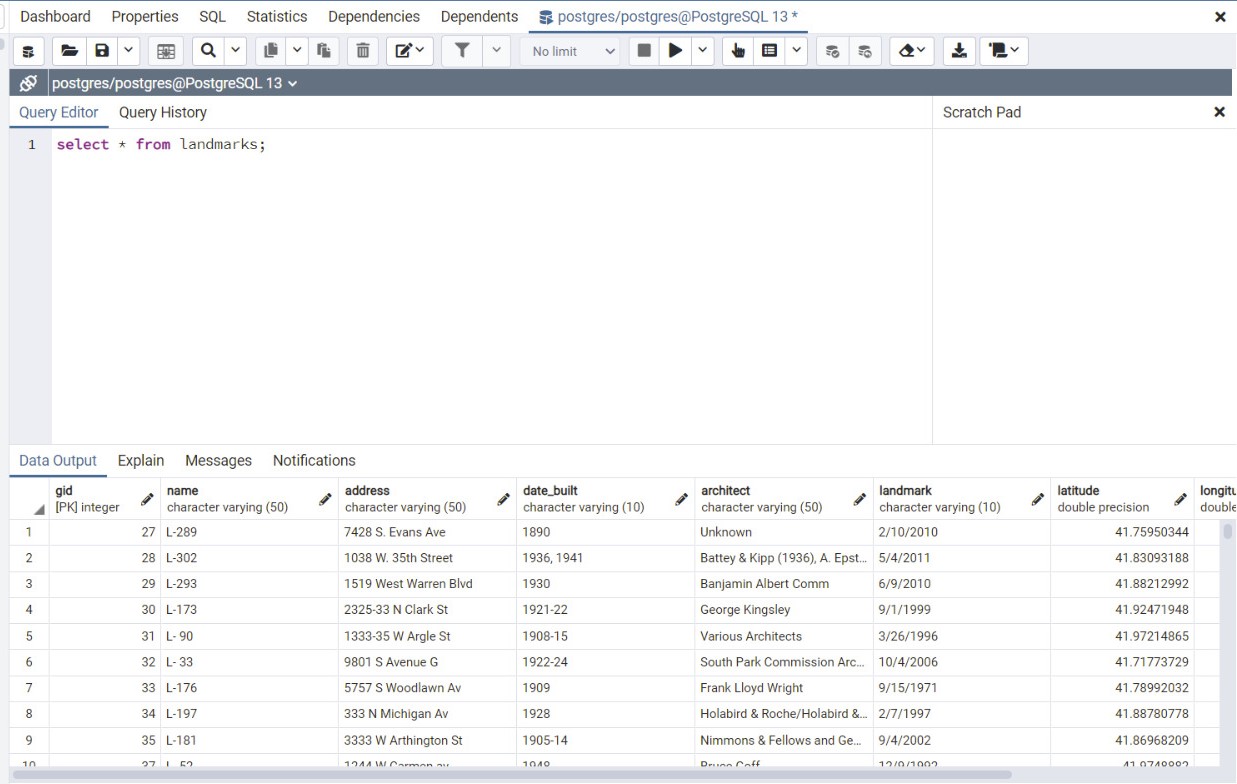
USING gist

(the\_geom );

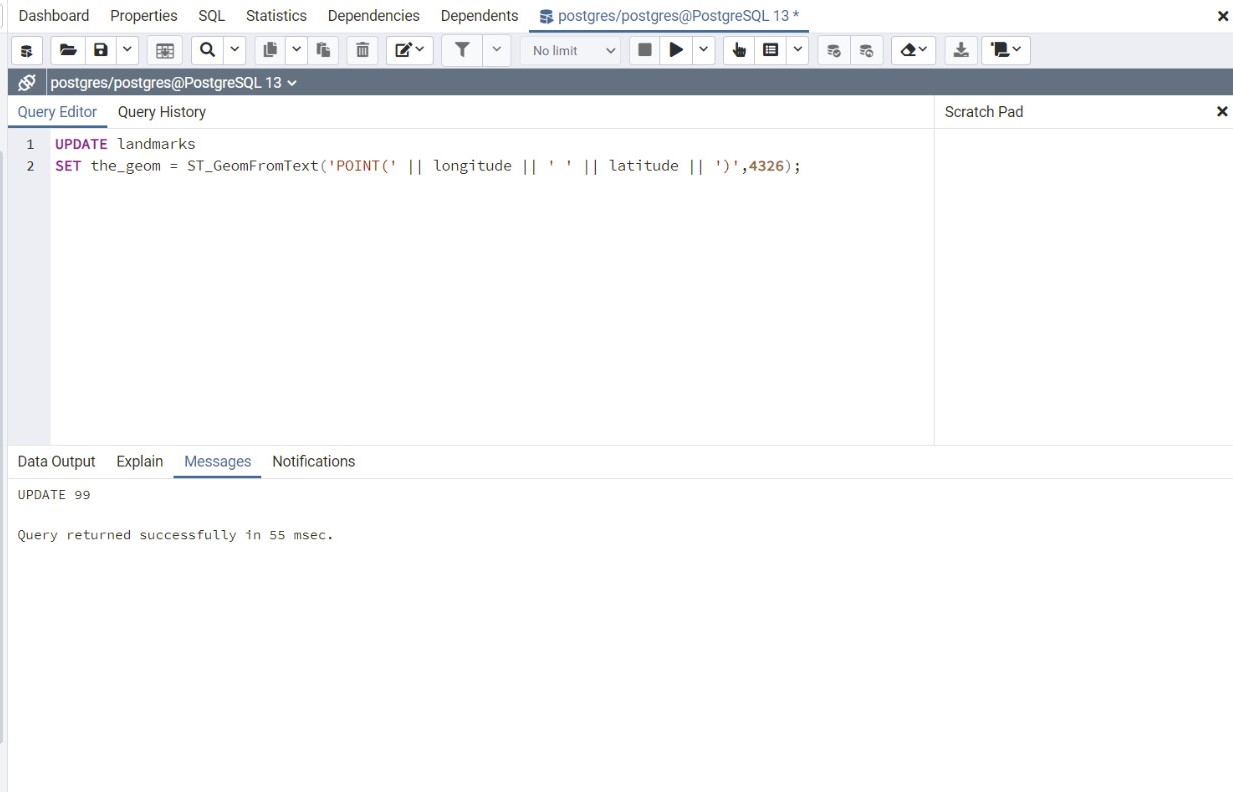
**Step 3: Copy the CSV data into the database**

Copy landmarks(name,address,date\_built,architect,landmark,latitude,longitude) FROM '/local/path/to/Individual\_Landmarks.csv' DELIMITERS ',' CSV HEADER;

**CSV data Imported**



**Step 4: Translate latitude and longitude into POINT geometry**

****

**Step 5: Running Queries**

**Query returns the 5 closest landmarks to a given latitude and longitude**

**Query:**

SELECT

ST\_Distance(ST\_GeomFromText('POINT(-87.6348345 41.8786207)', 4326), landmarks.the\_geom) AS planar\_degrees,

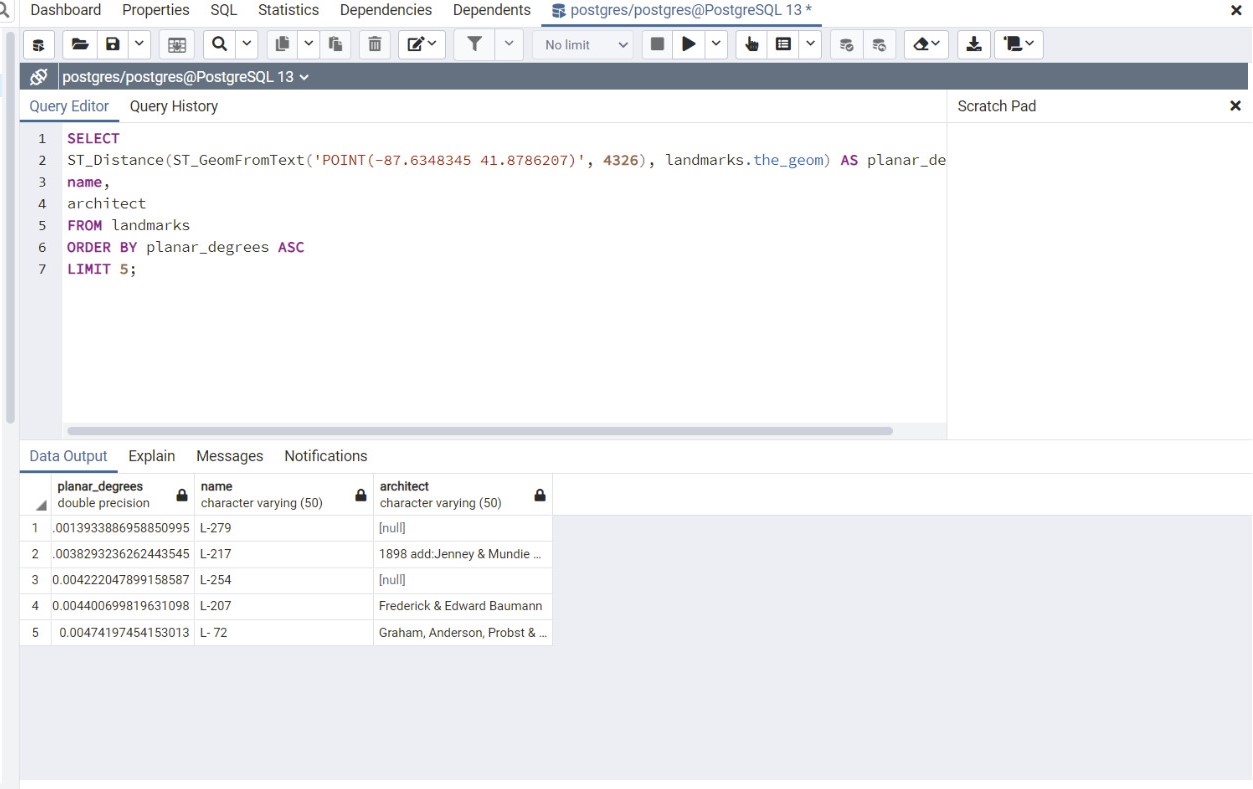
name,

architect

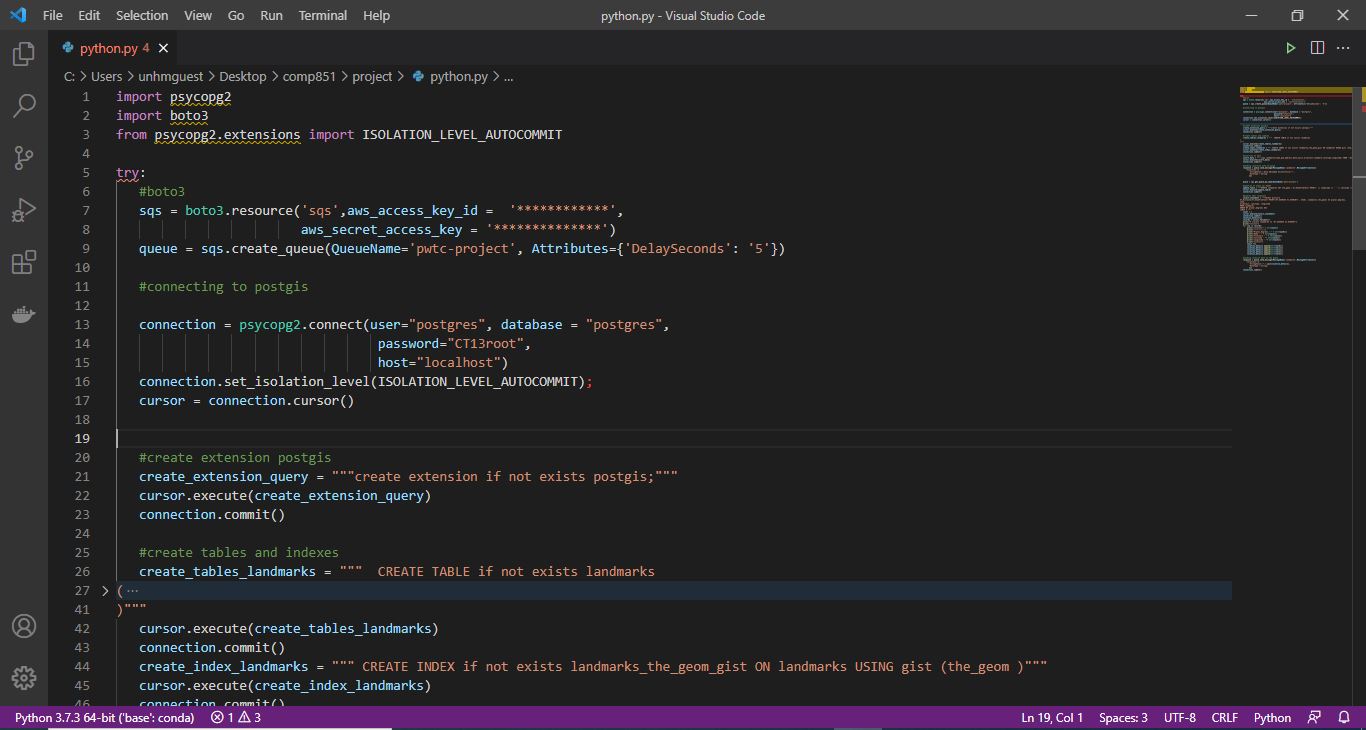
FROM landmarks

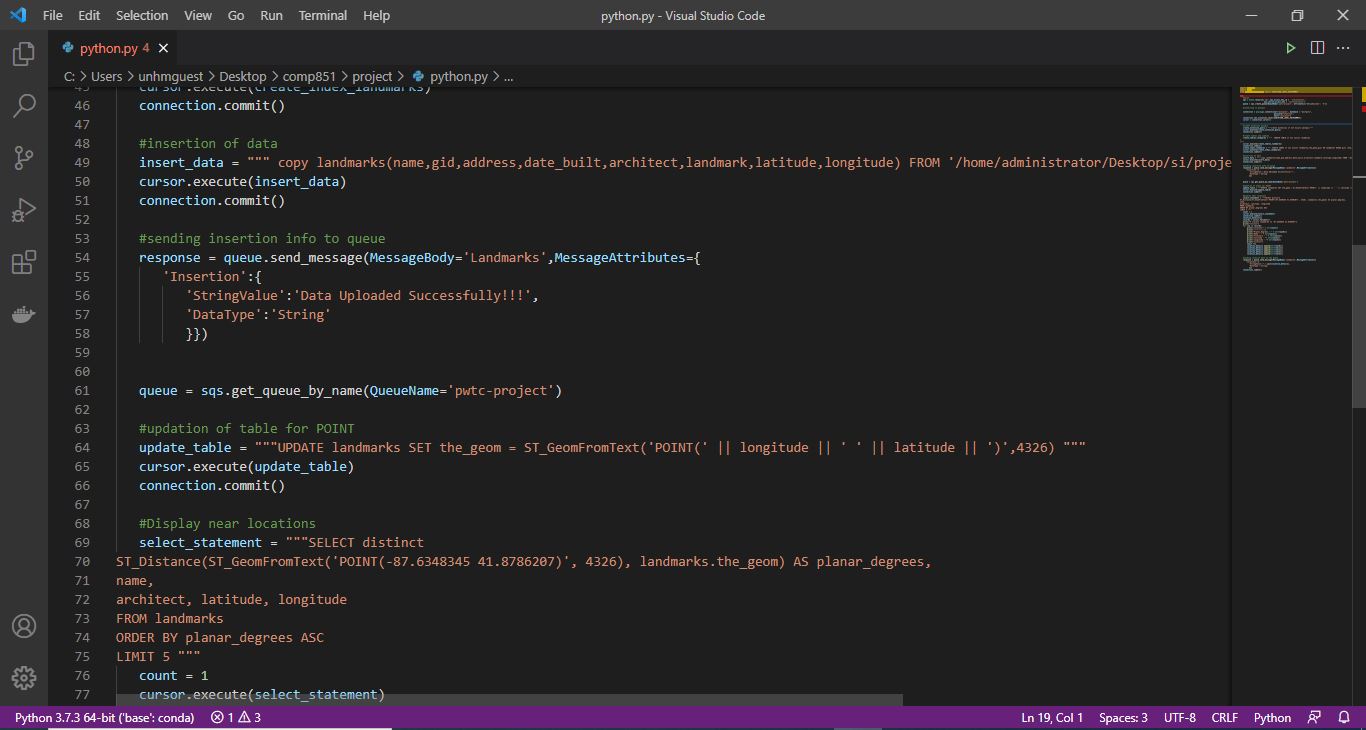
ORDER BY planar\_degrees ASC

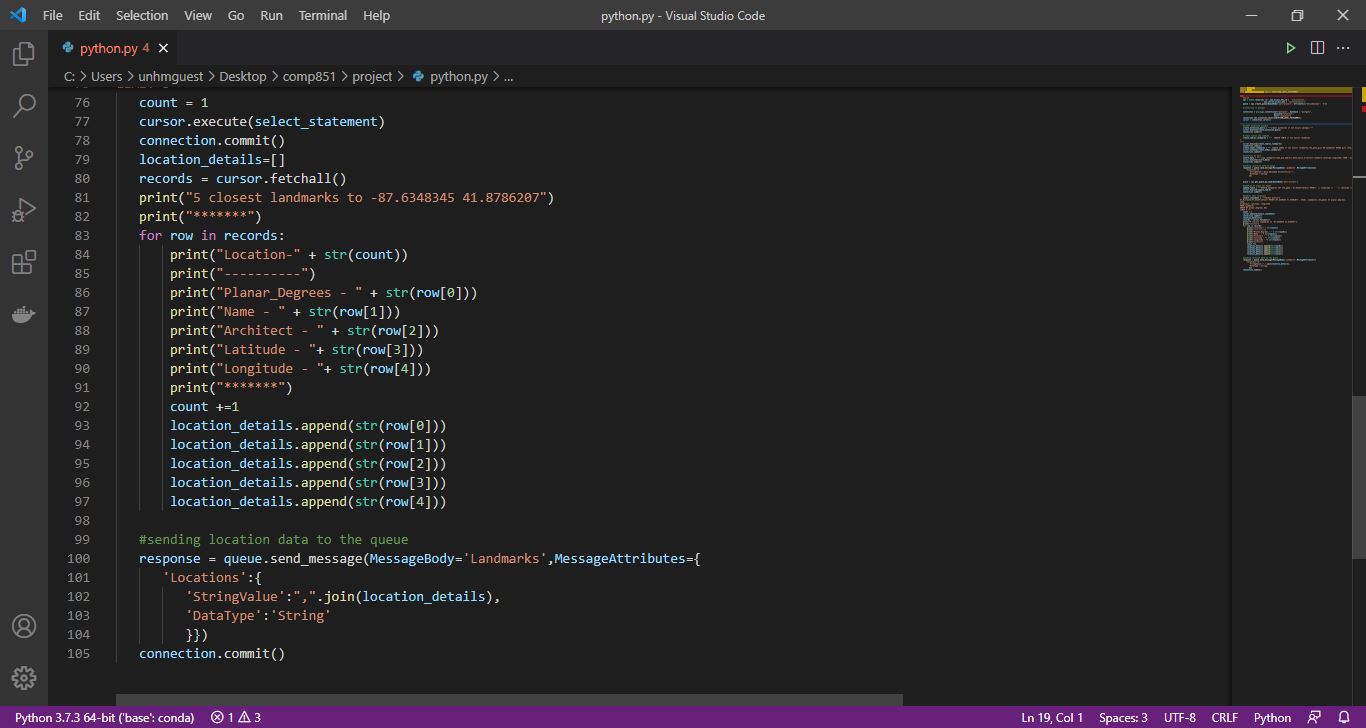
LIMIT 5;



**Implementing the project using Python3**

****

****



**Output:**

(base) unhmguest@UNHM-128-11:~/comp851/project$ python3 project.py

5 closest landmarks to -87.6348345 41.8786207

\*\*\*\*\*\*\*

Location-1

----------

Planar\_Degrees - 0.0007463848779255456

Name - Brooks Building

Architect - Holabird & Roche

Latitude - 41.87787644

Longitude - -87.63477822

\*\*\*\*\*\*\*

Location-2

----------

Planar\_Degrees - 0.0013933886958850995

Name - 300 West Adams Street Office Building

Architect - None

Latitude - 41.87972743

Longitude - -87.63568107

\*\*\*\*\*\*\*

Location-3

----------

Planar\_Degrees - 0.0018793811696624153

Name - Continental And Commercial National Bank Building

Architect - None

Latitude - 41.87907898

Longitude - -87.63301185

\*\*\*\*\*\*\*

Location-4

----------

Planar\_Degrees - 0.002712009853856664

Name - Chicago Board of Trade Building

Architect - Holabird & Root

Latitude - 41.87773513

Longitude - -87.63227115

\*\*\*\*\*\*\*

Location-5

----------

Planar\_Degrees - 0.00307102762438656

Name - Rookery Building

Architect - Burnham & Root

Latitude - 41.87907613

Longitude - -87.63179743

\*\*\*\*\*\*\*