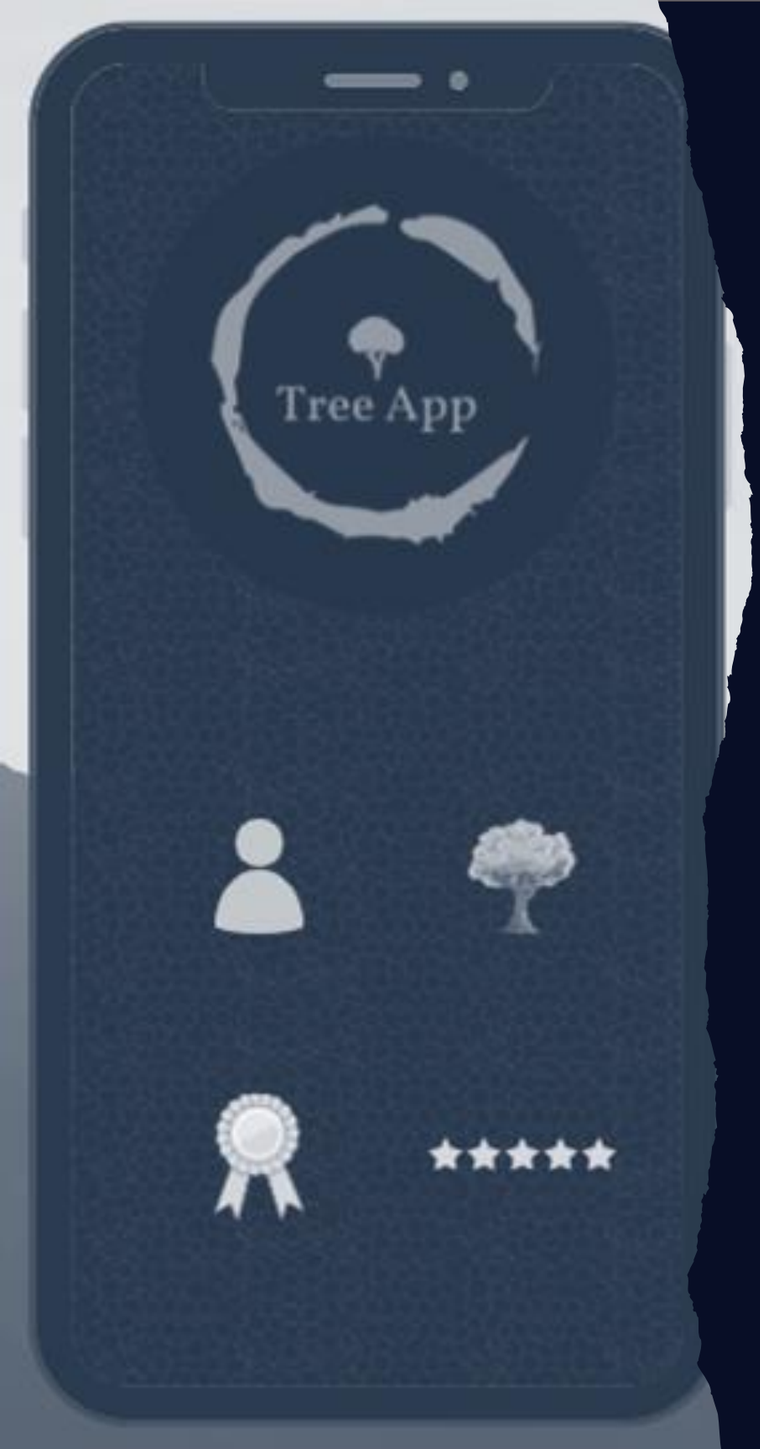
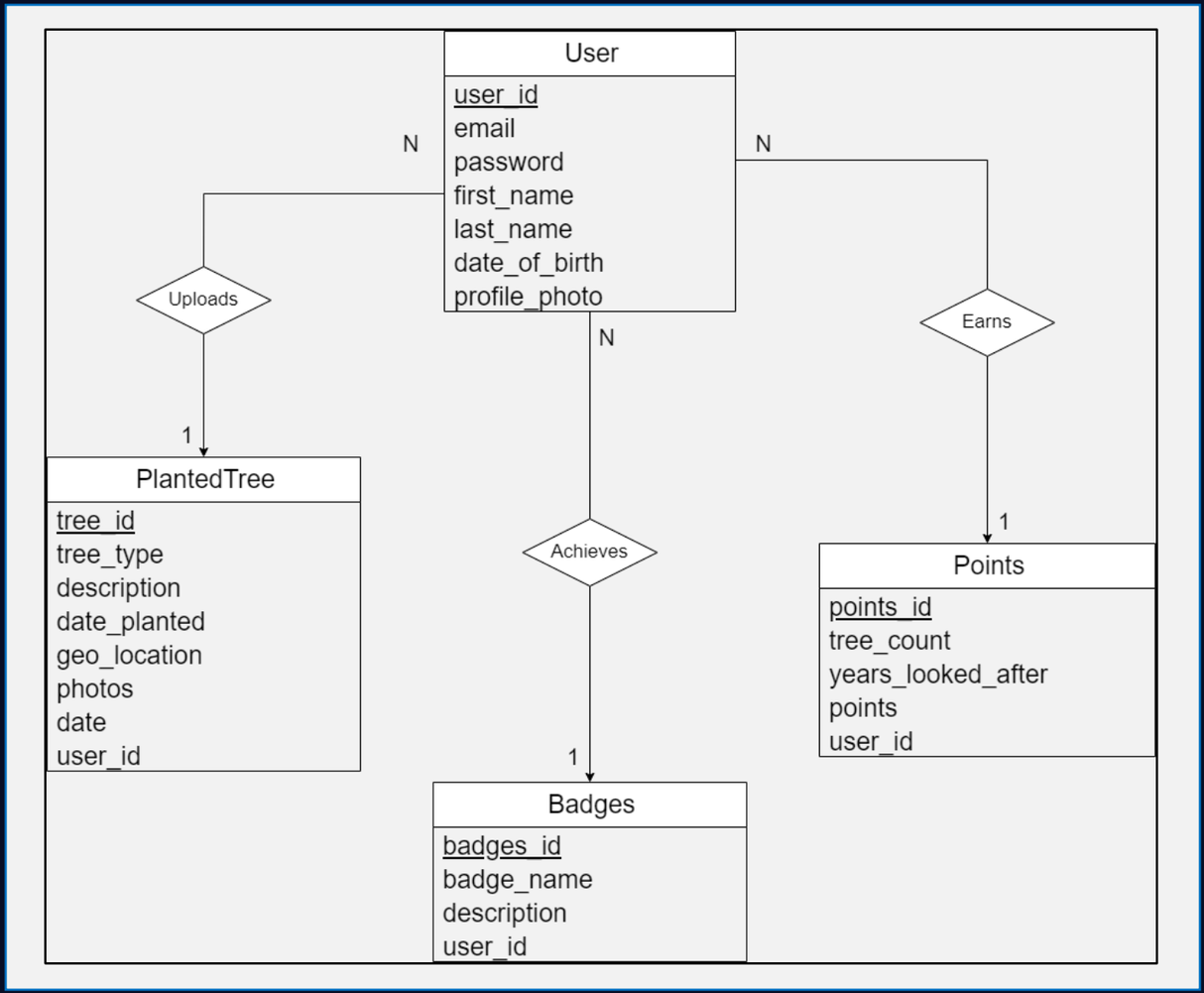


Tree_App Database Project

- TreeApp is a not-for-profit social startup that wants to accelerate reforestation by using technology.
- They are developing an app, where anyone can register and upload pictures of the trees they have planted, with other information such as description, date, geo location etc.
- Additionally, the users will get points based on the number of trees they plant and a number of years they look after them, and Badges (displayed on their profile) based on their achievements.

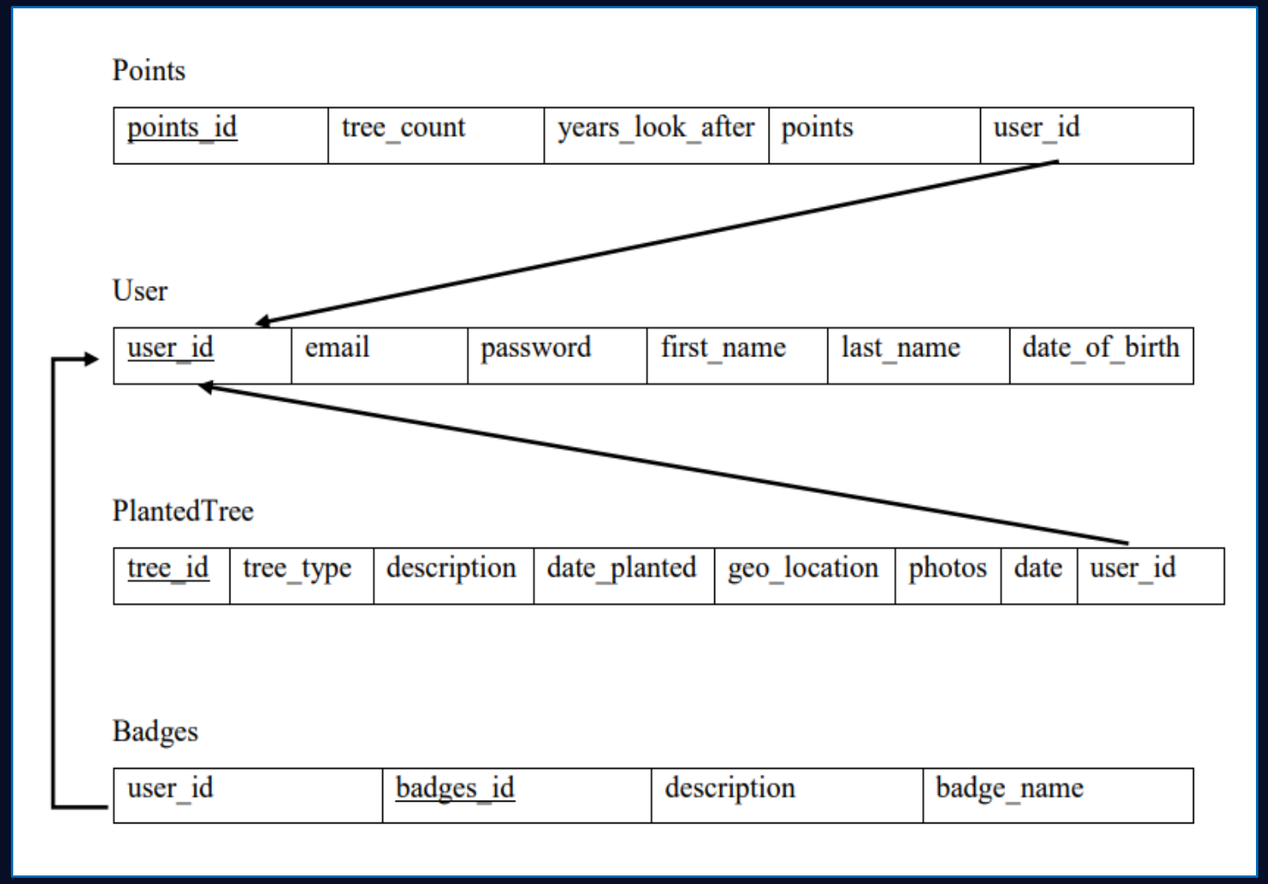
ER Diagram of tree_app

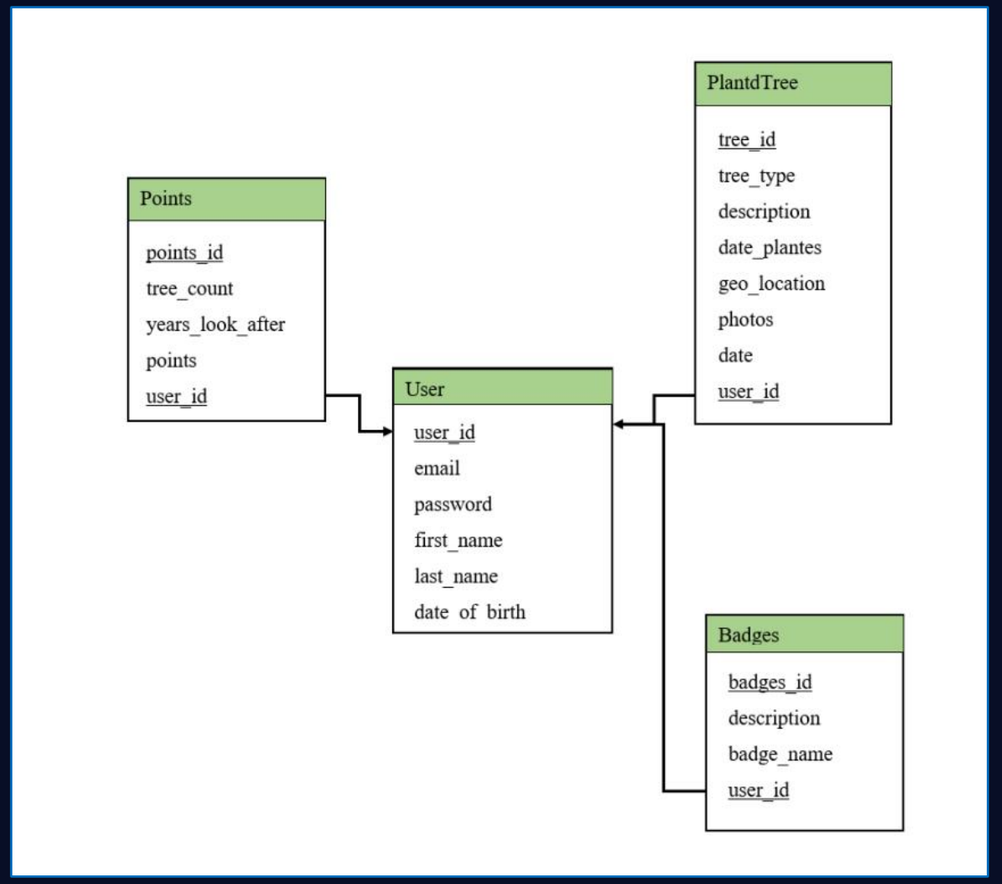


- 4 entities
  1. User
  2. Planted Tree
  3. Points
  4. Badges

- Assumptions:
  1. A user can plant multiple trees, but each tree has only one user.
  2. A user can earn multiple points, but each point is related to only one user.
  3. A user can earn multiple badges, but each badge is associated with only one user

Schema Diagram of tree_app





SQL database structure of tree_app

- 4 tables for 4 entities as:
    1. users
    2. planted_trees
    3. points
    4. badges

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	badges	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	planted_trees	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	points	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	users	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	4 tables	Sum	18	InnoDB	utf8mb4_general_ci	112.0 KiB	0 B

users table

- **SQL query to create users table:**

```
CREATE TABLE users (
user_id VARCHAR(255) PRIMARY KEY,
email VARCHAR(255),
password VARCHAR(255),
first_name VARCHAR(255),
last_name VARCHAR(255),
date_of_birth DATE,
);
```

- **Sample SQL query to insert values to users table:**

```
INSERT INTO users (user_id, email, password, first_name,
last_name, date_of_birth, profile_photo) VALUES

(U001, ‘Sanduni@gmail.com', ‘sanduni123', ‘Sanduni', ‘Silva',
‘2000-06-18’),

(U002, ‘buddhileka@gmail.com', ‘buddhileka123',
‘Buddhileka', ‘Gunarathna', ‘2000-07-22);
```

←T→			user_id	email	password	first_name	last_name	date_of_birth
☐	Edit	Copy	Delete	U001	sanduni@gmail.com	sanduni123	Sanduni	Silva	2000-06-18
☐	Edit	Copy	Delete	U002	buddhileka@gmail.com	buddhileka123	Buddhileka	Gunarathna	2000-07-22
☐	Edit	Copy	Delete	U003	lakmee@gmail.com	lakmee123	Lakmee	Chamodya	1999-01-28
☐	Edit	Copy	Delete	U004	prabhasha@gmail.com	prabhasha123	Bodhini	Prabhasha	2000-03-02

planted_trees table

- **SQL query to create planted_trees table:**

```
CREATE TABLE planted_trees (
tree_id VARCHAR(255) PRIMARY KEY,
tree_type VARCHAR(255),
description TEXT,
date_planted DATE,
geo_location VARCHAR(255),
photos VARCHAR(255), date DATE,
user_id VARCHAR(255, FOREIGN KEY
(user_id) REFERENCES users(user_id) );
```

- **Sample SQL query to insert values to planted_trees table:**

```
INSERT INTO planted_trees (tree_id, tree_type, description,
date_planted, geo_location, photos, date, user_id) VALUES

(Tree001, 'Oak', 'A beautiful oak tree planted in the
backyard.', '2021-03-15', '42.1234,-71.5678', 'photo1.jpg',
'2021-03-15', U001),

(Tree002, 'Maple', 'A maple tree planted in the park.', '2022-
06-10', '39.9876,-75.4321', 'photo2.jpg', '2022-06-10', U002);
```

←T→	tree_id	tree_type	description	date_planted	geo_location	photos	date	user_id
Edit Copy Delete	Tree001	Oak	A beautiful oak tree planted in the backyard.	2021-03-15	42.1234,-71.5678	photo1.jpg	2022-03-15	U001
Edit Copy Delete	Tree002	Maple	A maple tree planted in the park.	2022-06-10	39.9876,-75.4321	photo2.jpg	2023-05-10	U002
Edit Copy Delete	Tree003	Pine	A pine tree planted near the lake.	2023-01-05	37.5555,-122.3333	photo3.jpg	2023-01-05	U003
Edit Copy Delete	Tree004	Mango	A mango tree planted in the garden.	2018-06-05	39.9876,-75.4321	photo4.jpg	2023-05-05	U002
Edit Copy Delete	Tree005	Apple	An apple tree planted in the backyard.	2015-06-05	38.5555,-162.3345	photo5.jpg	2023-02-10	U004
Edit Copy Delete	Tree006	Mandarin	A mandarin tree planted in the garden.	2020-06-05	42.1234,-71.5678	photo6.jpg	2023-01-05	U001

points table

- **SQL query to create points table:**

```
CREATE TABLE points (
points_id VARCHAR(255) PRIMARY KEY,
tree_count INT,
years_looked_after INT,
points INT,
user_id VARCHAR(255, FOREIGN KEY
(user_id) REFERENCES users(user_id) );
```

- **Sample SQL query to insert values to points table:**

```
INSERT INTO points (points_id, tree_count,
years_looked_after, points, user_id) VALUES

(P001, 2, 3, 10, U001),
(P002, 2, 5, 14, U002),
(P003, 1, 1, 4, U003)
(P004, 1, 8, 18, U004);
```

		points_id	tree_count	years_looked_after	points	user_id
	Edit Copy Delete	P001	2	3	10	U001
	Edit Copy Delete	P002	2	5	14	U002
	Edit Copy Delete	P003	1	1	4	U003
	Edit Copy Delete	P004	1	8	18	U004

Click the drop-down arrow to toggle column's visibility.

badges table

- **SQL query to create badges table:**

```
CREATE TABLE badges (
badges_id VARCHAR(255) PRIMARY
KEY, badge_name VARCHAR(255),
description TEXT,
user_id VARCHAR(255, FOREIGN KEY
(user_id) REFERENCES users(user_id) );
```

- **Sample SQL query to insert values to badges table:**

```
INSERT INTO badges (badges_id, badge_name, description,
user_id) VALUES

(B001, ‘Green Thumb', 'Awarded to user who have started to
plant this year.’, U003),

(B002, ‘Tree Champion’, 'Awarded to user who have looked
after trees for 8 or more years.’, U004);
```

←T→	badges_id	badge_name	description	user_id
Edit Copy Delete	B001	Green Thumb	Awarded to users who have started to plnat this ye...	U003
Edit Copy Delete	B002	Tree Champion	Awarded to users who have looked after trees for m...	U004

Python: Functionality to insert a tree

- Code:

```python
import mysql.connector

# Connect to the MySQL database
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="tree_app"
)

# Function to insert a new tree record
def insert_tree(tree_id, tree_type, description, date_planted, geo_location, photos, date, user_id):
    cursor = db.cursor()
    sql = "INSERT INTO planted_trees (tree_id, tree_type, description, date_planted, geo_location, photos, date, user_id) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
    values = (tree_id, tree_type, description, date_planted, geo_location, photos, date, user_id)
    cursor.execute(sql, values)
    db.commit()
    print("Tree record inserted successfully!")
```

```python
# Insert 2 new trees
insert_tree("Tree007", "Birch", "A beautiful birch tree planted in the garden.", "2023-06-20", "40.1234,-74.5678", "photo7.jpg", "2023-07-03", "U001")
insert_tree("Tree008", "Guava", "A small guava tree planted in the backyard.", "2023-05-01", "40.1484,-74.6278", "photo8.jpg", "2023-07-04", "U004")
```

- Updated database:

☐	Edit	Copy	Delete	Tree007	Birch	A beautiful birch tree planted in the garden.	2023-06-20	40.1234,-74.5678	photo7.jpg	2023-07-03	U001
☐	Edit	Copy	Delete	Tree008	Guava	A small guava tree planted in the backyard.	2023-05-01	40.1484,-74.6278	photo8.jpg	2023-07-04	U004

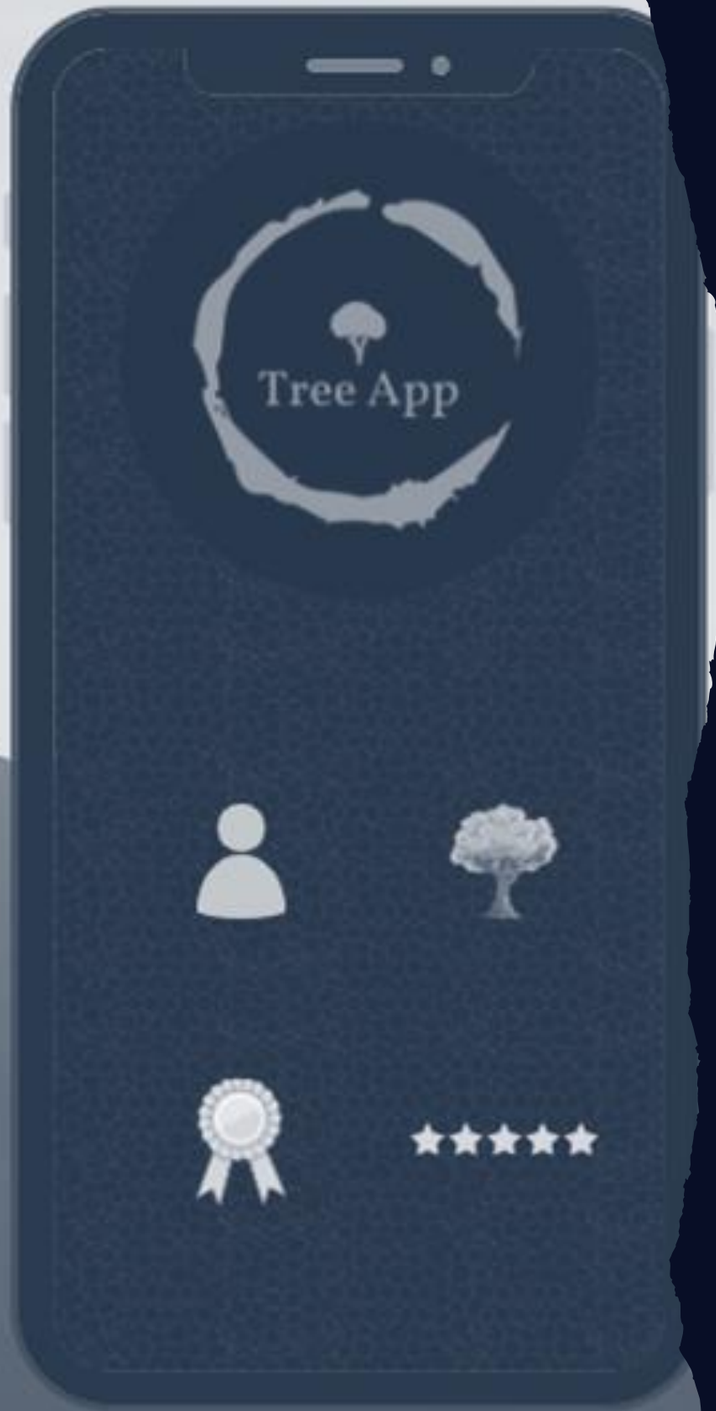
Python: Functionality to generate 2 reports

- Report 01: Details of Planted Trees

```python
# Report 01: Details of Planted trees
def get_planted_trees():
    cursor = db.cursor()
    sql = "SELECT * FROM planted_trees"
    cursor.execute(sql)
    result = cursor.fetchall()
    print("Details of Planted Trees:")
    for row in result:
        print("Tree ID:", row[0])
        print("Tree Type:", row[1])
        print("Description:", row[2])
        print("Date Planted:", row[3])
        print("Geo Location:", row[4])
        print("Photos:", row[5])
        print("Date:", row[6])
        print("User ID:", row[7])
        print("--------------------------")
```

- Report 02: Users & Points

```python
# Report 02: Users & Points
def report_users_and_points():
    cursor = db.cursor()
    sql = "SELECT users.first_name, users.last_name, users.email, SUM(points) as total_points FROM users JOIN points ON users.user_id = points.user_id GROUP BY users.user
    cursor.execute(sql)
    result = cursor.fetchall()
    print("Users and Points Report:")
    for row in result:
        print("Name:", row[0], row[1])
        print("Email:", row[2])
        print("Total Points:", row[3])
        print("--------------------------")
```



END