**Indexes Report**

There are three core tables: user, pet, and adoption—in the Flask-based pet adoption application.

**1. Table: user**

1. **Index: PRIMARY KEY (user\_id)**
   * **Definition:** Automatically created by the PRIMARY KEY constraint.
   * **Queries Benefiting:**
     + Any lookup by user\_id, especially in joins (e.g., retrieving user details for an adoption record).
   * **Used In:**
     + **Adoption History** feature (joining adoption.user\_id = user.user\_id).
   * **Justification:**
     + Provides fast, unique access to user records.
2. **Index: idx\_user\_email**
   * **Definition:** CREATE UNIQUE INDEX idx\_user\_email ON user (email);
   * **Queries Benefiting:**
     + SELECT \* FROM user WHERE email = :email;
   * **Used In:**
     + **Authentication & Account Management** (login, registration, uniqueness check).
   * **Justification:**
     + Enforces email uniqueness and accelerates email-based lookups, a common operation in user-centric workflows.

**2. Table: pet**

1. **Index: PRIMARY KEY (pet\_id)**
   * **Definition:** Automatically created by the PRIMARY KEY constraint.
   * **Queries Benefiting:**
     + Lookups by pet\_id (e.g., editing or deleting a pet record).
   * **Used In:**
     + **Edit Pet**, **Delete Pet**, and **Pet Detail** pages (Pet.query.get(pet\_id)).
   * **Justification:**
     + Ensures efficient CRUD operations on individual pet records.
2. **Index: idx\_pet\_type\_status**
   * **Definition:** CREATE INDEX idx\_pet\_type\_status ON pet (type, status);
   * **Queries Benefiting:**
     + SELECT \* FROM pet WHERE type = :type AND status = 'available';
     + Partitioning the pet listing by type and availability.
   * **Used In:**
     + **Pet Listing** page with filters (e.g., browsing available dogs or cats).
   * **Justification:**
     + Composite index optimizes the common pattern of filtering by both type and status.
3. **Index: idx\_pet\_age**
   * **Definition:** CREATE INDEX idx\_pet\_age ON pet (age);
   * **Queries Benefiting:**
     + SELECT type, COUNT(\*) AS count, AVG(age) AS avg\_age FROM pet WHERE type = :type AND age BETWEEN :min\_age AND :max\_age GROUP BY type;
   * **Used In:**
     + **Pet Age Statistics Report** (/report route).
   * **Justification:**
     + Accelerates range-based queries on age, crucial for analytical reporting features.

**3. Table: adoption**

1. **Index: PRIMARY KEY (adoption\_id)**
   * **Definition:** Automatically created by the PRIMARY KEY constraint.
   * **Queries Benefiting:**
     + Lookups by adoption\_id.
   * **Used In:**
     + CRUD operations on adoption records.
   * **Justification:**
     + Supports fast, unique retrieval of each adoption entry.
2. **Index: idx\_adoption\_user\_id**
   * **Definition:** CREATE INDEX idx\_adoption\_user\_id ON adoption (user\_id);
   * **Queries Benefiting:**
     + SELECT \* FROM adoption WHERE user\_id = :user\_id;
     + Joins to fetch adopted pets for a user:
     + SELECT p.\*
     + FROM adoption a
     + JOIN pet p ON a.pet\_id = p.pet\_id
     + WHERE a.user\_id = :user\_id;
   * **Used In:**
     + **User Profile**: Adoption History section.
   * **Justification:**
     + Speeds up user-centric queries, enhancing profile and dashboard features.
3. **Index: idx\_adoption\_pet\_id**
   * **Definition:** CREATE INDEX idx\_adoption\_pet\_id ON adoption (pet\_id);
   * **Queries Benefiting:**
     + SELECT \* FROM adoption WHERE pet\_id = :pet\_id;
     + Joins to fetch adopters for a pet:
     + SELECT u.\*
     + FROM adoption a
     + JOIN user u ON a.user\_id = u.user\_id
     + WHERE a.pet\_id = :pet\_id;
   * **Used In:**
     + **Pet Detail** page: shows adoption status and adopter info.
   * **Justification:**
     + Optimizes pet-specific adoption history lookups.

**Notes & Future Considerations**

* These indexes could be declared in SQLAlchemy models via \_\_table\_args\_\_:
* class User(db.Model):
* \_\_table\_args\_\_ = (
* db.Index('idx\_user\_email', 'email', unique=True),
* )
* Consider adding an index on adoption\_date for a **Monthly Adoption Trends** report:
* CREATE INDEX idx\_adoption\_date ON adoption (adoption\_date);
* Regularly monitor index usage and maintenance overhead; drop unused indexes as the application evolves.

**Summary Table**

| **Table** | **Index Name** | **Columns** | **Feature / Report** |
| --- | --- | --- | --- |
| user | PRIMARY KEY | user\_id | Lookup by ID (joins, detail pages) |
| user | idx\_user\_email | email | Login / Registration |
| pet | PRIMARY KEY | pet\_id | CRUD operations |
| pet | idx\_pet\_type\_status | type, status | Pet Listing (available by type) |
| pet | idx\_pet\_age | age | Pet Age Statistics Report |
| adoption | PRIMARY KEY | adoption\_id | CRUD operations |
| adoption | idx\_adoption\_user\_id | user\_id | User Adoption History |
| adoption | idx\_adoption\_pet\_id | pet\_id | Pet Adoption History |