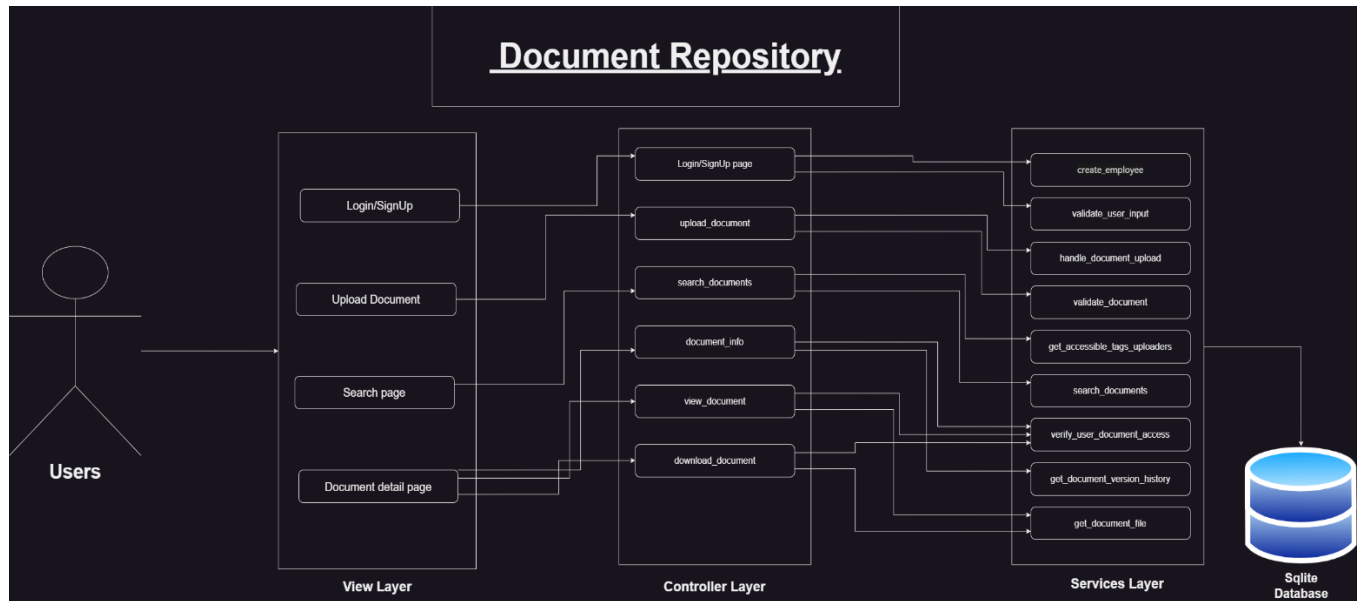


# Youssef Khaled

## Deliverable 1 - System Architecture Diagram



## Design choices

### Layered Architecture

- **View:** Jinja templates for forms and results; no business logic
- **Controller:** Flask Blueprints handle requests, sessions, and input validation
- **Services:** Core business logic in helpers/services.py; data validation in helpers/validators.py
- **Data:** SQLAlchemy models over SQLite; files stored locally in backend/files

### Document Management

- Document + DocumentVersion model for immutable versioning
- Tag-based search via many-to-many relationships
- Department-based access control (NULL = public)

### Security & Scalability

- Session-based auth with hashed passwords; admin-only signup

- Access checks at controller and service layers; safe file handling with `secure_filename`
- Rate limiting scaffolded (Flask-Limiter) for abuse protection
- Service layer abstraction enables easy S3 migration and horizontal scaling

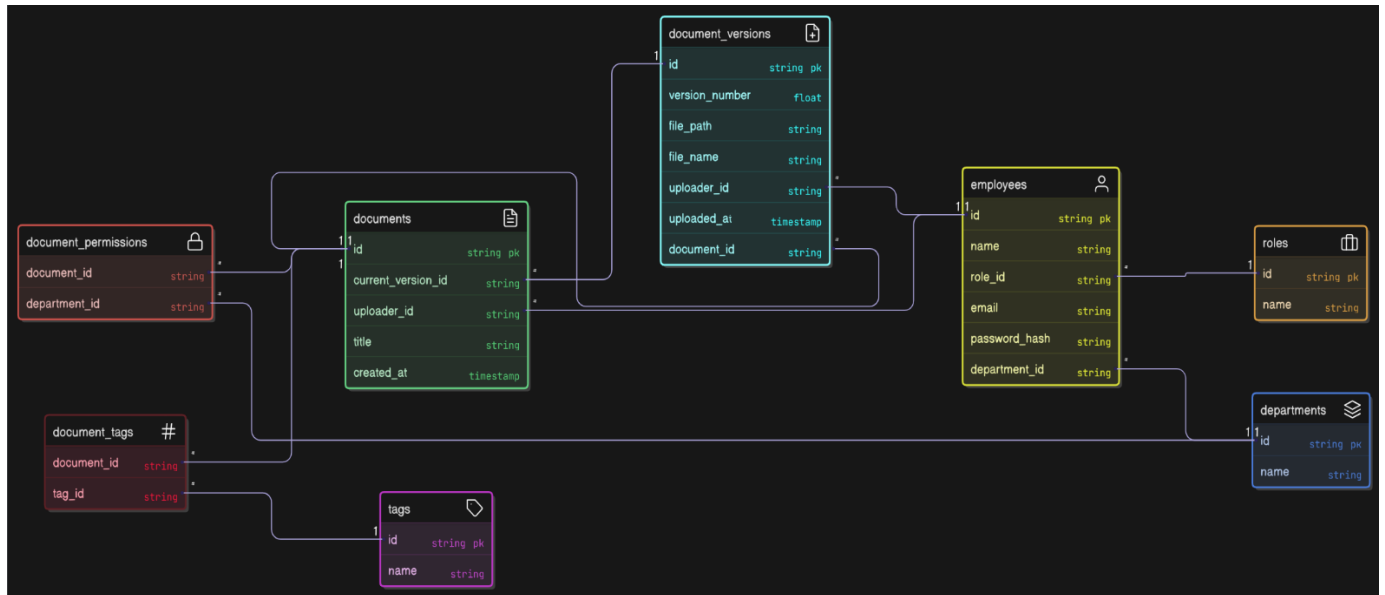
### **User-Friendly Features**

- Upload: validates file type/size, prevents duplicate versions, auto-creates tags/departments
- Search: filters by title/uploader/tags with permission-aware queries and preloaded dropdowns
- Access: verify permissions before showing metadata or serving files
- Clear error messages and flash notifications for user feedback

### **Trade-offs**

- SQLite for simplicity; filesystem storage for speed
- Department-based RBAC keeps queries fast vs per-user ACLs
- Ready for cloud storage and database migration via service layer abstraction

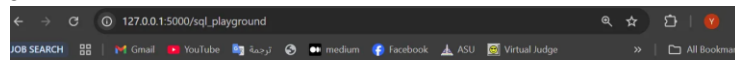
## Deliverable 2 - ER Diagram



## SQL Queries

-- Get all documents accessible to a user in their department

```
select d.title from employees e join document_permissions dp on
(dp.department_id = e.department_id or dp.department_id is null)
join documents d on d.id = dp.document_id where e.id = 1
```



### SQLite Playground

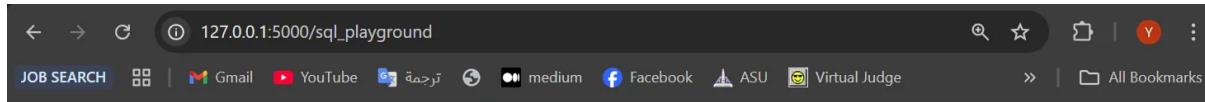
```
select d.title from employees e
join document_permissions dp on (dp.department_id = e.department_id or dp.department_id is null)
join documents d on d.id = dp.document_id
where e.id = 1
```

Run

title
IT Security Policy
Onboarding Checklist
Infrastructure Overview
Infrastructure Overview
Company Policies
Benefits Overview
Network Topology

-- Get the 10 most recently uploaded documents tagged as Finance.

```
select d.* from documents d join document_tags dt on d.id =  
dt.document_id join tags t on t.id = dt.tag_id where t.name =  
"Finance" order by d.created_at desc limit 10
```



## SQLite Playground

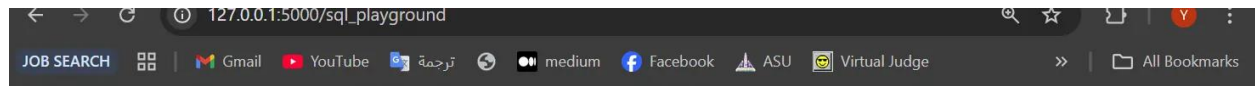
```
-- Get the 10 most recently uploaded documents tagged as Finance.  
select d.* from documents d  
join document_tags dt on d.id = dt.document_id  
join tags t on t.id = dt.tag_id  
where t.name = "Finance"  
order by d.created_at desc  
limit 10
```

Run

id	current_version_id	title	uploader_id	created_at
3	6	Quarterly Finance Report	4	2025-09-09 10:28:39.067880

-- Find all versions of a given document (by document ID).

```
select version_number from document_versions where document_id = 1
```



## SQLite Playground

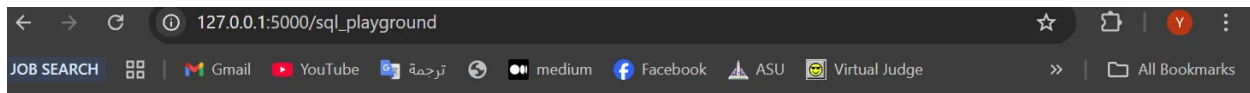
```
-- Find all versions of a given document (by document ID).  
select version_number from document_versions  
where document_id = 1
```

Run

version_number
1.0
1.1

-- Get the number of documents uploaded by each department in the last 30 days.

```
select d.name , count(*) as "number of documents" from  
document_versions dv join employees e on e.id = dv.uploader_id join  
departments d on d.id = e.department_id where dv.uploaded_at >=  
DATE('now', '-30 days') group by d.name
```



## SQLite Playground

```
-- Get the number of documents uploaded by each department in the last 30 days.  
select d.name , count(*) as "number of documents" from document_versions dv  
join employees e on e.id = dv.uploader_id  
join departments d on d.id = e.department_id  
where dv.uploaded_at >= DATE('now', '-30 days')  
group by d.name
```

Run

name	number of documents
Finance	2
HR	4
IT	7
Legal	1
Sales	1

## Deliverable 3 - GitHub Repo & POC video

[github.com Document-Repository](#)

[POC Working Video](#)