Faculty E-Library System - Backend API Specification

# Authentication Service

Handles user registration, login, token verification, and account expiration.

## POST /api/auth/register

Description: Register a new user with their school email.

Parameters: email, password, role, graduationYear

Response: 201 Created - User registered successfully

Roles Allowed: All

——————————————————————————————————————————————————

## POST /api/auth/login

Description: Authenticate user and return JWT token.

Parameters: email, password

Response: 200 OK - JWT token issued

Roles Allowed: All

——————————————————————————————————————————————————

## GET /api/auth/verify

Description: Verify if a user’s token is valid and not expired.

Parameters: token

Response: 200 OK - Valid token

Roles Allowed: All

——————————————————————————————————————————————————

## DELETE /api/auth/cleanup-expired

Description: Delete expired student accounts after graduation year.

Parameters: None

Response: 204 No Content - Expired accounts deleted

Roles Allowed: System/Automated

——————————————————————————————————————————————————

# User Management Service

Manages user data, roles, and account info.

## GET /api/users

Description: List all registered users.

Parameters: None

Response: 200 OK - List of users

Roles Allowed: Admin staff

——————————————————————————————————————————————————

## PUT /api/users/{id}/role

Description: Update a user’s role (e.g., promote student to academic staff).

Parameters: id, newRole

Response: 200 OK - Role updated

Roles Allowed: Admin staff

——————————————————————————————————————————————————

## DELETE /api/users/{id}

Description: Delete a specific user account.

Parameters: id

Response: 204 No Content - User deleted

Roles Allowed: Admin staff

——————————————————————————————————————————————————

# Document Management Service

Handles document upload, retrieval, and categorization.

## POST /api/docs/upload

Description: Upload a new document for approval.

Parameters: title, description, file, category

Response: 201 Created - Document uploaded and pending approval

Roles Allowed: Academic staff, Student

——————————————————————————————————————————————————

## GET /api/docs

Description: Fetch all approved documents.

Parameters: category, keyword

Response: 200 OK - List of approved documents

Roles Allowed: All

——————————————————————————————————————————————————

## GET /api/docs/{id}

Description: Retrieve a specific document by ID.

Parameters: id

Response: 200 OK - Document retrieved

Roles Allowed: All

——————————————————————————————————————————————————

# Admin Approval Service

Manages document review and approval workflow.

## GET /api/approvals/pending

Description: View all pending document uploads awaiting admin approval.

Parameters: None

Response: 200 OK - List of pending uploads

Roles Allowed: Admin staff

——————————————————————————————————————————————————

## PUT /api/approvals/{docId}/approve

Description: Approve a specific document for publishing.

Parameters: docId

Response: 200 OK - Document approved

Roles Allowed: Admin staff

——————————————————————————————————————————————————

## PUT /api/approvals/{docId}/reject

Description: Reject a specific document upload.

Parameters: docId, reason

Response: 200 OK - Document rejected

Roles Allowed: Admin staff

——————————————————————————————————————————————————

# Gateway Service

Routes all incoming requests to the appropriate microservice via internal LAN.

## GET /api/status

Description: Check the operational status of all backend services.

Parameters: None

Response: 200 OK - Service health report

Roles Allowed: Admin staff

——————————————————————————————————————————————————