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

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

POST /api/auth/register # Register new user (with email validation)

POST /api/auth/login # Login with email/password

POST /api/auth/refresh # Refresh JWT token

POST /api/auth/logout # Logout user

POST /api/auth/forgot-password # Request password reset

POST /api/auth/reset-password # Reset password with token

GET /api/auth/validate-token # Validate JWT token

```

### \*\*User Service (Port: 8082)\*\*

```

# User Management

GET /api/users # Get all users (Admin only)

GET /api/users/{id} # Get user by ID

PUT /api/users/{id} # Update user details

DELETE /api/users/{id} # Delete user (Admin only)

GET /api/users/profile # Get current user profile

PUT /api/users/profile # Update current user profile

# Role Management (Admin only)

GET /api/users/roles # Get all roles

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

GET /api/users/by-role/{role} # Get users by role

# Student Account Management

GET /api/users/students/expiring # Get students with expiring accounts

POST /api/users/students/cleanup # Manual cleanup of expired accounts

GET /api/users/students/stats # Get student statistics

```

### \*\*Document Service (Port: 8083)\*\*

```

# Document CRUD

POST /api/documents/upload # Upload new document (authenticated users)

GET /api/documents # Get all approved documents (with filters)

GET /api/documents/{id} # Get document details

PUT /api/documents/{id} # Update document metadata (owner/admin)

DELETE /api/documents/{id} # Delete document (owner/admin)

GET /api/documents/download/{id} # Download document file

# Category-specific endpoints

GET /api/documents/category/educational-articles

GET /api/documents/category/projects-research

GET /api/documents/category/lecture-notes

GET /api/documents/category/past-questions

GET /api/documents/category/textbooks

# Search & Filter

GET /api/documents/search # Search documents (title, author, description)

GET /api/documents/filter # Filter by category, date, uploader

GET /api/documents/my-uploads # Get current user's uploads

# Categories

GET /api/documents/categories # Get all categories

POST /api/documents/categories # Create category (Admin)

# Statistics

GET /api/documents/stats # Get document statistics

GET /api/documents/popular # Get most downloaded documents

GET /api/documents/recent # Get recently uploaded documents

```

### \*\*Approval Service (Port: 8084)\*\*

```

# Approval Management (Admin Staff only)

GET /api/approvals/pending # Get all pending documents

GET /api/approvals/{id} # Get approval request details

POST /api/approvals/{id}/approve # Approve document

POST /api/approvals/{id}/reject # Reject document (with reason)

GET /api/approvals/history # Get approval history

GET /api/approvals/my-requests # Get current user's approval requests

# Statistics (Admin only)

GET /api/approvals/stats # Get approval statistics

GET /api/approvals/by-admin/{adminId} # Get approvals by admin

```

### \*\*Notification Service (Port: 8085)\*\*

```

GET /api/notifications # Get user notifications

GET /api/notifications/{id} # Get notification details

PUT /api/notifications/{id}/read # Mark notification as read

DELETE /api/notifications/{id} # Delete notification

GET /api/notifications/unread # Get unread notifications count