**API Contracts - OpenAPI Specification for CAPS**

This OpenAPI specification defines the **endpoints, request parameters, and response details** for the **Comprehensive Assessment Preparation System (CAPS)**.

**General Information**

openapi: 3.0.0

info:

title: CAPS API

description: Comprehensive Assessment Preparation System API Documentation

version: 1.0.0

servers:

- url: http://localhost/api

description: Local development server

- url: https://caps.example.com/api

description: Production server

**Authentication Endpoints**

**1. Register a New User**

paths:

/register:

post:

summary: Register a new user

tags:

- Authentication

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

userCode:

type: string

username:

type: string

email:

type: string

format: email

password:

type: string

responses:

201:

description: User registered successfully

400:

description: Validation error

**2️. User Login**

/login:

post:

summary: User login

tags:

- Authentication

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

email:

type: string

password:

type: string

responses:

200:

openapi: 3.0.0

info:

title: CAPS API

description: Comprehensive Assessment Preparation System API Documentation

version: 1.0.0

servers:

- url: http://localhost/api

description: Local development server

- url: https://caps.example.com/api

description: Production server

paths:

/register:

post:

summary: Register a new user

tags:

- Authentication

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

userCode:

type: string

username:

type: string

email:

type: string

format: email

password:

type: string

responses:

201:

description: User registered successfully

400:

description: Validation error

/login:

post:

summary: User login

tags:

- Authentication

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

email:

type: string

password:

type: string

responses:

200:

description: Login successful

401:

description: Invalid credentials

**3️. User Logout**

/logout:

post:

summary: User logout

tags:

- Authentication

security:

- bearerAuth: []

responses:

200:

description: Logout successful

401:

description: Unauthorized

description: Login successful

401:

description: Invalid credentials

**3️. User Logout**

/logout:

post:

summary: User logout

tags:

- Authentication

security:

- bearerAuth: []

responses:

200:

description: Logout successful

401:

description: Unauthorized

**Subjects Endpoints**

**4️. Get All Subjects**

/subjects:

get:

summary: Get all subjects

tags:

- Subjects

responses:

200:

description: List of subjects retrieved

**5️. Create a New Subject (Admin)**

post:

summary: Create a new subject

tags:

- Subjects

security:

- bearerAuth: []

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

subjectCode:

type: string

subjectName:

type: string

responses:

201:

description: Subject created successfully

403:

description: Forbidden (only admins can create subjects)

**Questions Endpoints**

**6️. Get All Questions (Faculty/Admin)**

/questions:

get:

summary: Get all questions

tags:

- Questions

security:

- bearerAuth: []

responses:

200:

description: List of questions retrieved

**7️. Submit a Question (Faculty)**

post:

summary: Submit a new question (Pending Approval)

tags:

- Questions

security:

- bearerAuth: []

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

questionnaireID:

type: integer

questionText:

type: string

imagePath:

type: string

nullable: true

responses:

201:

description: Question submitted successfully (Pending Approval)

**8️. Approve a Question (Program Chair/Dean)**

/questions/{id}/approve:

put:

summary: Approve a question

tags:

- Questions

security:

- bearerAuth: []

parameters:

- name: id

in: path

required: true

schema:

type: integer

responses:

200:

description: Question approved

404:

description: Question not found

**9️. Modify a Question (Instructor)**

/questions/{id}/modify:

put:

summary: Request to modify a question (Pending Approval)

tags:

- Questions

security:

- bearerAuth: []

parameters:

- name: id

in: path

required: true

schema:

type: integer

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

questionText:

type: string

responses:

200:

description: Modification request submitted

**10. Modify Choices of a Question**

/questions/{id}/choices/modify:

put:

summary: Modify choices for a question (Pending Approval)

tags:

- Questions

security:

- bearerAuth: []

parameters:

- name: id

in: path

required: true

schema:

type: integer

requestBody:

required: true

content:

application/json:

schema:

type: array

items:

type: object

properties:

choiceText:

type: string

isCorrect:

type: boolean

responses:

200:

description: Choices modified successfully

**11. Get All Pending Modifications**

/questions/pending-modifications:

get:

summary: Get all pending question modifications

tags:

- Questions

security:

- bearerAuth: []

responses:

200:

description: List of pending modifications

**12. Approve or Reject Question Modifications**

/questions/{id}/approve-modification:

put:

summary: Approve or reject a question modification

tags:

- Questions

security:

- bearerAuth: []

parameters:

- name: id

in: path

required: true

schema:

type: integer

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

status:

type: string

enum: [Approved, Rejected]

responses:

200:

description: Modification status updated

**Exams Endpoints**

**13. Start an Exam (Student)**

/exams:

post:

summary: Start an exam

tags:

- Exams

security:

- bearerAuth: []

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

subjectCode:

type: string

type:

type: string

responses:

201:

description: Exam started successfully

**14. Submit an Answer**

/answers:

post:

summary: Submit an answer

tags:

- Exams

security:

- bearerAuth: []

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

examID:

type: integer

questionID:

type: integer

choiceID:

type: integer

responses:

200:

description: Answer submitted.