OCR Feature

Steps:

1. Import PDF
2. Loop through pdf pages to save interested area as img
3. Pass image through easyOCR library to generate questions for validation
4. First round of validation through deepseek api
5. Validate questions and answers
6. Label questions and answers

APIs required:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Steps | API | Param | Response | Notes | Status |
| 1 | /POST importPDF | {  file: file  filename: string  } | Success 200OK  {  fileId: uuid  filename: string  uploadedFilePath: string  } | File size and format validation |  |
| 2 | /GET importedPDF/{fileId} | {  fileId: uuid  } | Success 200OK  {  filePath: string  } |  |  |
|  | /POST defineArea/{fileId} | {  [  coordinates: list[int]  type: string (question/option)  questionNum: int  ],[…]  } | Success 200OK  {  imgId: uuid  imgFileFolder: string  type: string (question/ option)  } | Only can recognize text, how to handle if questions include img like graph >> focus on text first  Pre-processing techniques:  Save as image (like a screencapture)  Future:  Use edge detection (e.g., Canny edge detection) to detect areas with heavy text vs. graphical content. |  |
| 3 | /POST processImg | {  imgId: uuid  } | Success 200OK  {  questionId: uuid  question: string  options: string  } | May have to include math parsing libraries |  |
|  | GET /processImgStatus/{imgId} | {  imgId: uuid  } | Success 200OK  {  status: string  } | Implementing status and progress tracking for long-running tasks |  |
| 4 | /GET questions/{questionId} |  | Success 200OK  {  questionId: uuid  question: string  options: string  } |  |  |
|  | GET /labelOptions |  | Success 200OK  {  topics: List[string],  difficulty: List[int]  } |  |  |
|  | /PUT updateQuestions/{questionId} | {  questionId: uuid  question: string  options: string  } | Success 200OK  {  questionId: uuid  question: string  options: string  } |  |  |
| 5 | /GET questions/{questionId} |  | Success 200OK  {  questionId: uuid  question: string  options: string  } |  |  |
|  | /PUT  labelQuestions/{questionId} | {   questionId: uuid  topic: string  difficulty: int  correctAnswer: string  school: string  } | Success 200OK  {  questionId: uuid  question: string  options: string  topic: string  difficulty: int  correctAnswer: string  school: string  others: string  } | Future:  machine learning model to classify questions into topics |  |
|  | /GET exportQuestions |  | Success 200OK  {  file: file  } | Export as CSV |  |
| 6 | /delete  deleteUploadedPDF |  |  |  |  |
|  | /delete  image |  |  |  |  |
|  | /delete  deleteProcessedImg |  |  |  |  |

DB

|  |  |  |
| --- | --- | --- |
| PDF DB | IMG DB | Question bank DB |
| fileId: uuid  filename: string  uploadedFilePath: string | imgId: id  imgFileFolder: string  type: string (question/ option) | questionId: uuid  question: {  question: “”,  options:  {1:”string”, 2:”string”…}  correctOption: int  }  topic: string  difficulty: int  school: string  others: string  question: string  options: list[string]  correctAnswer: string |