**POST APIs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Purpose** | **Method** | **Route** | **Request Body (Json)** | **Response (Json)** |
| Get all questions with optional filter | GET | /api/questions  ?&courseId=  &tag=  &keyword= | N/A | [  {  questionId: number,  title: string,  content: string,  userId:number,  courseId: number,  viewCount: number,  upVotes: number,  downVotes: number,  isAnonymous: bool,  createdAt: string,  updatedAt: string  }  ] |
| Get a single question with all the answers and comments | GET | /api/questions/:questionId | N/A | {  questionId: number,  title: string,  content: string,  userId: number,  courseId: number,  viewCount: number,  upVotes: number,  downVotes: number,  isAnonymous: bool,  createdAt: string,  updatedAt: string,  answers: [],  comments: []  } |
| Post a question | POST | /api/questions/ | {  title: string,  content: string,  userId: number,  courseId: number,  isAnonymous:bool  } | {  questionId: number,  title: string,  content: string,  userId: number,  courseId: number,  viewCount: number,  upVotes: number,  downVotes: number,  isAnonymous: bool,  createdAt: string,  updatedAt: string,  answers: []  } |
| Update a question | PUT | /api/questions/:questionId | {  title: string,  content: string,  courseId: number,  isAnonymous:bool  } | {  questionId: number,  title: string,  content: string,  courseId: number,  isAnonymous: bool,  updatedAt: string  } |
| Delete a question | DELETE | /api/questions/:questionId | N/A | {  success: bool,  message: string  } |

**ANSWER APIs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Purpose** | **Method** | **Route** | **Request Body (Json)** | **Response (Json)** |
| Get an answer | GET | /api/answers/:answerId | N/A | {  answerId: number,  questionId: number,  content: string,  is\_accepted: 0 (F) or 1(T),  isAnonymous: 0(F) or 1(T)  score: number,  firstname: string,  lastname: string,  created\_ai: string,  updated\_at: string,  upvotes: number,  downVotes: number  } |
| Post an answer | POST | /api/answers/ | {  content: string,  questionId: string,  userId: string,  isAnonymous:bool  } | {  answerId: number,  questionId: number,  content: string,  is\_accepted: 0 (F) or 1(T),  isAnonymous: 0(F) or 1(T)  score: number,  firstname: string,  lastname: string,  created\_ai: string,  updated\_at: string,  upvotes: number,  downVotes: number  } |
| Update an answer | PUT | /api/answers/:answerId | {  content: string,  isAnonymous:bool  } | {  answerId: string,  content: string,  isAnonymous: 0(F) or 1(T),  updated\_at: string  } |
| Delete an answer | DELETE | /api/answers/:answerId | N/A | {  success: bool,  message: string  } |

**COMMENT APIs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Generate an AI comment to an answer | POST | /api/comments/generate-ai-comment | {  body: string,  answer\_id: number  } | {  comment\_id: number,  answer\_id: number,  body: string,  created\_at: string,  user\_id: number,  last\_name: string,  first\_name: string  } |

**RATING APIs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Purpose** | **Method** | **Route** | **Request Body (Json)** | **Response (Json)** |
| Rate a question | POST | /api/questions/:questionId/rate | {  type: number (1 for upvote and -1 for downvote),  userId: string  } | {  questionId: string,  upVotes: number,  downVotes: number  } |
| Rate an answer | POST | /api/answers/:answerId/rate | {  type: number (1 for upvote and -1 for downvote),  userId: string  } | {  answerId: string,  upVotes: number,  downVotes: number  } |

**USER APIs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Purpose** | **Method** | **Route** | **Request Body (Json)** | **Response (Json)** |
| Login a user | POST | /api/users/signin | {  username: string,  password: string  } | {  userId: string,  firstName: string,  lastName: string,  email: string  } |
| Register a user | POST | /api/users/signup | {  firstName: string,  lastName: string,  username: string,  password: string,  email: string  } | {  userId: string,  firstName: string,  lastName: string,  email: string  } |