

Document Model of Database for User Registration

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema

const UserSchema = new Schema({
  username: {
    type:String
  },
  useremail:{
    type:String
  },

  userid:{
    type:String
  },

  userpassword:{
    type:String,
    required:true
  },
  usertype:{
    type:String,
    required:true
  },
  date:{
    type:Date,
    default:Date.now
  }
})

module.exports = User = mongoose.model('users',UserSchema)
```

Document Model of Database for Q&A part of Quiz Application

Collection name: Questions

Each document is added to this collection when the examiner/ moderator creates a question.

Each document in this collection has..

- Question_id
- Question_description (type: String)
- Answer_type (type: integer)
- Difficulty_level (type: integer)
- created_time

Collection: Answers

Use-case: This collection has the choices stored for each question along with the correct answer.

Each document in the Collection has:

Question_id : Id referring to the question.

Choices: List of choices for the given question

Correct_choice_indices: List of indices of the correct choice.

Time Stamp

Answers

- _id
- Question_id
- AnswerList(type: List)
- Correct_choice_indices(type:List)
- created_time

Collection: Answer_type

Use-case: When the Question setter wants the moderator to give questions from specific type,this collection can be used to quickly filter questions

Each document in the Collection has:

Answer_type_id : [0- checkbox,1- Radio, 2- Textarea,3-File Upload]

Question_id: Question Id referring to the question

Answer_type: [String to indicate question_type. Eg: checkbox]

Answer_type

- Answer_type_id
- Question_id
- Answer_type(type:String)
- created_time

Collection: Question_category

Use-case: When the Question setter wants the moderator to give questions from specific category,this collection can be used to quickly filter questions

Each document in the Collection has:

Question_category_id : [0-OOPS,1-Networking, 2-React,3-Python]

Question_id: Id referring to the question

Question_category: [String to indicate question_category. Eg: OOPS]

Time Stamp

Parent_categories

- Parent_category_id
- Parent_category_name(type:String)

questions_in_a_category

- Question_id
- Parent-category_name
- created_time

Child_category

- Child_category_id
- Question_id
- Child_category_name(type:String)
- created_time

Collection: Tests

This collection stores details of each test as its documents.

Each document in the Collection has:

- Test_id
- Test_setter_id: Id of the person who wants to set the test
- Question_ids: Array of Question ids of questions in the test referring to Questions collection.
- Test_validity: Time Stamp indicating the time at which the test closes
- Test_duration: Time in hours of test duration

Tests

- Test_id
- Examiner_id
- Number_of Sections
- Test_Vaildity(type: Timestamp)
- created_time
- Questions_per_page(type: Integer, default: 1)
-

Test_sections

- _id
- Test_id
- Section_name
- Duration
- Question_ids
- Pos_score_per_question
- -ve score/question

Examinees

- Test_id
- id
- User_id
- status

Collection: Examinee_summary

This collection contains a summary about the results of each student for a test.

- Student_id
- Test_id
- Score(type: int/float)
- Pass(type: boolean)
-
- created_time

Collection: Marked_Answers

This collection contains information about the answers marked for a question by a student.

- Student_id
- Question_id
- test_id
- given_answer
- marks_allotted
- created_time

Collection: Certification_tests

- Certification_test_id
- certification_test_category(type:String)
- certification_test_sub_category(type:String)
- Difficulty_level
- total_score
- Question_id(type:Array)
- created_time