Sample data json

Prisma ORM: Open source Node.js and TypeScript ORM with a readable data model, automated migrations, type-safety, and auto-completion - https://www.prisma.io/

Users

```
model User {
  id              String      @id @default(auto()) @map("_id") @db.ObjectId
  firstname       String      @default("aaa")
  lastname        String      @default("bbb")
  password        String
  username        String
  avatar          String?
  public_id       String?
  birth           String?
  sex             String?
  phone           String?
  email           String      @unique
  role            Role        @default(STUDENT)
  updatedAt       String      @default("")
  createdAt       String      @default("")
  courseIDs   String[]    @db.ObjectId
  courses     Course[]    @relation(fields: [courseIDs], references: [id])
  quizResults QuizResult  @relation(fields: [quizResultsId], references: [id])
  quizResultsId String    @unique @db.ObjectId
  quizResults QuizResult[]
  courseResults CourseResult[]
}
```

Relation:

- One to many: User - QuizResult
- One to many: User - CourseResult

```
[
  {
    "id": "1",
    "firstname": "John",
    "lastname": "Doe",
    "password": "hashed_password",
    "username": "j.doe",
    "email": "john@example.com",
    "role": "STUDENT"
  },
  {
    "id": "2",
    "firstname": "Jane",
    "lastname": "Smith",
    "password": "hashed_password",
    "username": "j.smith",
    "email": "jane@example.com",
    "role": "TEACHER"
  },
  {
    "id": "3",
    "firstname": "Adam",
    "lastname": "Swift",
    "password": "hashed_password",
    "username": "a.swift",
    "email": "adam@example.com",
    "role": "ADMIN"
  }
]
```

Required fields : firstname, lastname, password (hashed based on bcryptjs), email, role (default STUDENT)

Category

```
model Category {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  name        String          @default("")
  courses     Course[]
}
```

Relation

- One to many: Category - Course

```
[
  {
    "id": "1",
    "name": "Mathematics"
  },
  {
    "id": "2",
    "name": "Science"
  }
]
```

Semester

```
model Semester {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  description String
  start_date  DateTime
  end_date    DateTime
  courses     Course[]
}
```

Relation

- One to many: Semester - Course

```
[
  {
    "id": "1",
    "description": "Spring Semester",
    "start_date": "2024-01-01T00:00:00Z",
    "end_date": "2024-06-30T23:59:59Z"
  },
  {
    "id": "2",
    "description": "Fall Semester",
    "start_date": "2024-08-01T00:00:00Z",
    "end_date": "2024-12-31T23:59:59Z"
  }
]
```

Course

```
model Course {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  title       String
  description String
  image       String?
  public_id   String?
  course_id   String

  VideoSections  VideoSection[]
  DocumentSections DocumentSection[]
  category    Category        @relation(fields: [categoryID], references: [id])
  categoryID  String          @db.ObjectId

  semester       Semester          @relation(fields: [semesterId], references:
[id])
  semesterId     String          @db.ObjectId

  users       User[]          @relation(fields: [usersId], references: [id])
  usersId     String[]        @db.ObjectId
  courseResults CourseResult[]
}
```

Relation

- Many to many: Course - User
- One to many: Course - CourseResult

```
[
  {
    "id": "1",
    "title": "Introduction to Algebra",
    "description": "A beginner's guide to algebraic concepts.",
    "categoryID": "1",
    "semesterId": "1"
  },
  {
    "id": "2",
    "title": "Biology 101",
    "description": "An introductory course to basic biology principles.",
    "categoryID": "2",
    "semesterId": "2"
  }
]
```

VideoSection

```
model VideoSection {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  title       String
  description String?
  videos      Video[]
  courseId    String          @map("course_id") @db.ObjectId
  course      Course          @relation(fields: [courseId], references: [id])
}
```

Relation:

- One to many: VideoSection - Video

```
[
  {
    "id": "1",
    "title": "Basic Algebraic Equations",
    "courseId": "1"
  },
  {
    "id": "2",
    "title": "Cell Structure",
    "courseId": "2"
  }
]
```

DocumentSection

```
model DocumentSection {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  title       String
  content     String?
  quiz        Quiz[]
  documentLink DocumentLink[]
  courseId    String          @map("course_id") @db.ObjectId
  course      Course          @relation(fields: [courseId], references: [id])
}
```

Relation:

- One to many: DocumentSection - Document
- One to many: DocumentSection - DocumentLink

```
[
  {
    "id": "1",
    "title": "Solving Linear Equations",
    "courseId": "1"
  },
  {
    "id": "2",
    "title": "DNA Structure",
    "courseId": "2"
  }
]
```

Video

```
model Video {
  id                  String          @id @default(auto()) @map("_id")
@db.ObjectId
  title               String
  description         String?
  url                 String
  videoSection        VideoSection    @relation(fields: [videoSectionId],
references: [id])
  videoSectionId      String          @db.ObjectId
}
```

```
[
  {
    "id": "1",
    "title": "Introduction to Linear Equations",
    "url": "https://example.com/video1",
    "videoSectionId": "1"
  },
  {
    "id": "2",
    "title": "Prokaryotic Cells",
    "url": "https://example.com/video2",
    "videoSectionId": "2"
  }
]
```

DocumentLink

```
model DocumentLink {
  id                 String           @id @default(auto()) @map("_id")
@db.ObjectId
  title              String
  description        String?
  url                String
  documentSection    DocumentSection  @relation(fields: [documentSectionId],
references: [id])
  documentSectionId  String           @db.ObjectId
}
```

```
[
  {
    "id": "1",
    "title": "Solving Quadratic Equations Worksheet",
    "url": "https://example.com/worksheet1",
    "documentSectionId": "1"
  },
  {
    "id": "2",
    "title": "Cell Biology Lecture Notes",
    "url": "https://example.com/lecture_notes1",
    "documentSectionId": "2"
  }
]
```

Quiz

```
model Quiz {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  title       String
  description String?
  start_date  DateTime?
  end_date    DateTime?
  time_limit  Int?
  factor      Float           @default(0)
  questions   Question[]
  documentSection     DocumentSection        @relation(fields: [documentSectionId],
references: [id])
  documentSectionId  String          @db.ObjectId
  quizResults  QuizResult[]
}
```

Relation:

- One to many: Quiz - Question

```
[
  {
    "id": "1",
    "title": "Algebra Quiz",
    "description": "Test your knowledge on algebraic equations.",
    "start_date": "2024-04-10T00:00:00Z",
    "end_date": "2024-04-20T23:59:59Z",
    "time_limit": 60,
    "documentSectionId": "1"
  },
  {
    "id": "2",
    "title": "Biology Quiz",
    "description": "Test your understanding of basic biology concepts.",
    "start_date": "2024-05-10T00:00:00Z",
    "end_date": "2024-05-20T23:59:59Z",
    "time_limit": 45,
    "documentSectionId": "2"
  }
]
```

Question

```
model Question {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  content     String
  options     String[]
  answer      String
  quiz        Quiz            @relation(fields: [quizId], references: [id])
  quizId      String          @db.ObjectId
}
```

```
[
  {
    "id": "1",
    "content": "What is the solution to 2x + 3 = 7?",
    "options": ["x = 2", "x = 3", "x = 4", "x = 5"],
    "answer": "x = 2",
    "quizId": "1"
  },
  {
    "id": "2",
    "content": "What is the powerhouse of the cell?",
    "options": ["Nucleus", "Mitochondria", "Golgi apparatus", "Endoplasmic
reticulum"],
    "answer": "Mitochondria",
    "quizId": "2"
  }
]
```

QuizResult

```
model QuizResult {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  score       Float           @default(0)
  userId      String          @db.ObjectId
  user        User            @relation(fields: [userId], references: [id])
  quiz        Quiz            @relation(fields: [quizId], references: [id])
  quizId      String          @db.ObjectId
  createdAt   DateTime        @default(now())
  updatedAt   DateTime        @default(now())
}
```

```
[
  {
    "id": "1",
    "score": 85.5,
    "userId": "1",
    "quizId": "1"
  },
  {
    "id": "2",
    "score": 92.0,
    "userId": "2",
    "quizId": "2"
  }
]
```

CourseResult

```
model CourseResult {
  id          String          @id @default(auto()) @map("_id") @db.ObjectId
  score_array Float[]         @default([0, 0, 0, 0])
  average_score Float         @default(0)
  userId      String          @db.ObjectId
  user        User            @relation(fields: [userId], references: [id])
  course      Course          @relation(fields: [courseId], references: [id])
  courseId    String          @db.ObjectId
  createdAt   DateTime        @default(now())
  updatedAt   DateTime        @default(now())
}
```

```
[
  {
    "id": "1",
    "score_array": [75, 80, 85, 90],
    "average_score": 82.5,
    "userId": "1",
    "courseId": "1"
  },
  {
    "id": "2",
    "score_array": [80, 85, 90, 95],
    "average_score": 87.5,
    "userId": "2",
    "courseId": "2"
  }
]
```