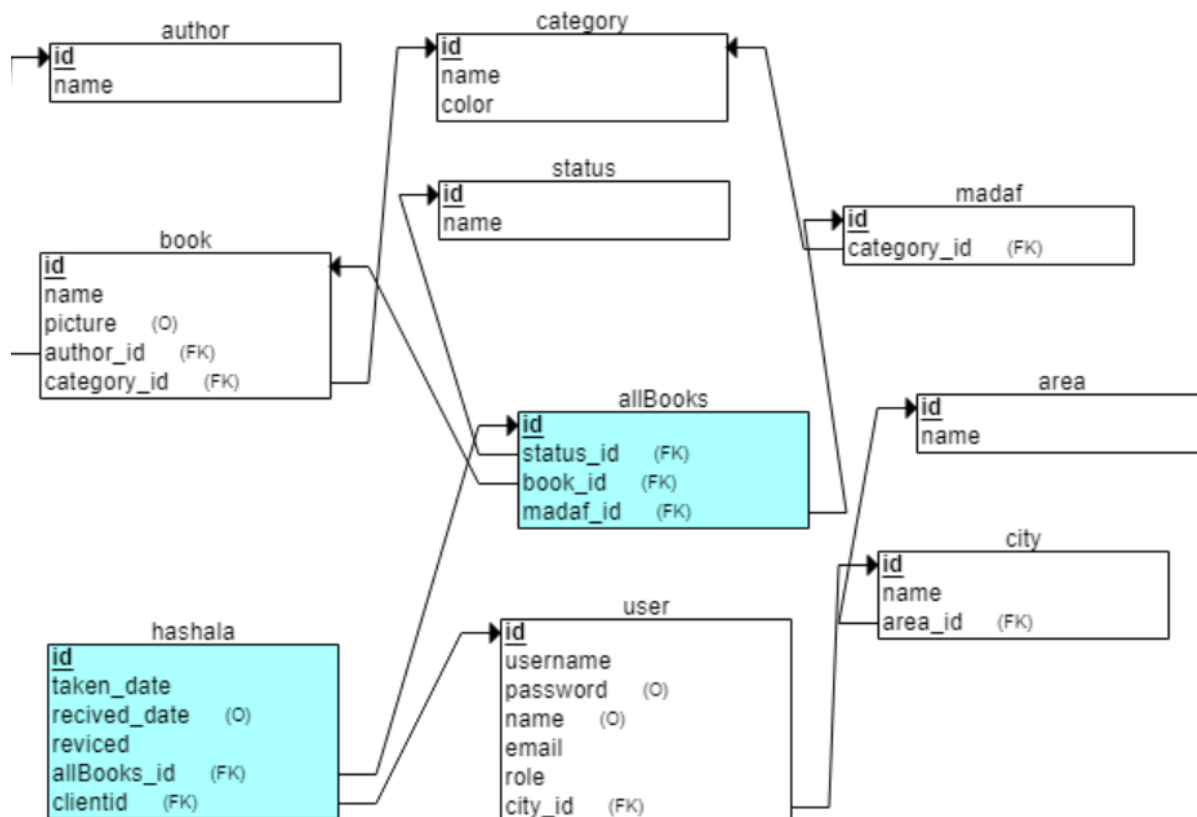


קבלת השאלות למשתמש - Ex6

נוסיף את מודלים hashala ו allBooks



1. יצירת המודלים

allBooks.js

```
const { sequelize, DataTypes } = require("../sequelize");
const AllBooks = sequelize.define("allBooks", {
  status_id: {
    type: DataTypes.INTEGER,
  },
  book_id: {
    type: DataTypes.INTEGER,
    allowNull: false
  }
});
```

```
module.exports = AllBooks;
const { sequelize, DataTypes } = require("./sequelize");
```

hashala.js

```
const Hashala = sequelize.define("hashala", {
  allBooks_id: {
    type: DataTypes.INTEGER,
    allowNull:false
  },
  user_id: {
    type: DataTypes.INTEGER,
    allowNull:false
  },
  recived: {
    type: DataTypes.BOOLEAN,
    defaultValue:false,
    allowNull:false
  },
  recivedDate: {
    type: DataTypes.DATE,
  }
});
module.exports = Hashala;
```

2. עדכון המודל במסד הנתונים

ונוסיף אותו בקובץ הראשי של מסד הנתונים index.js

```
db.allBooks = require('./allBooks')
db.hashala = require('./hashala')
```

3. עדכון קשרים

```
const { sequelize } = require("./sequelize");
```

```

const applyExtraSetup = () => {
  const { book, author, category, user, hashala, allBooks, status } =
sequelize.models;
  book.belongsTo(category, { foreignKey: "cateogry_id", as: "category" });
  book.belongsTo(author, { foreignKey: "author_id", as: "author" });
  author.hasMany(book, { foreignKey: "author_id", as: "books" });
  category.hasMany(book, { foreignKey: "cateogry_id", as: "books" });

  //ALL BOOKS TO STATUS
  allBooks.belongsTo(status, { foreignKey: "status_id", as: "status" });
  status.hasMany(allBooks, { foreignKey: "status_id", as: "statuses" });
  //ALL BOOKS TO BOOK
  allBooks.belongsTo(book, { foreignKey: "book_id", as: "book" });
  book.hasMany(allBooks, { foreignKey: "book_id", as: "books" });

  //HASHALA TO USER
  hashala.belongsTo(user, { foreignKey: "user_id", as: "user" });
  user.hasMany(hashala, { foreignKey: "user_id", as: "hashala" });

  //HASHALA TO USER
  hashala.belongsTo(allBooks, { foreignKey: "allBooks_id", as: "allBooks"
});
  allBooks.hasMany(hashala, { foreignKey: "allBooks_id", as: "hashala" });

};

module.exports = { applyExtraSetup };

```

4. יצירת הקונטרולר

hashalaController.js

לקבלת כל ההשאלות למשתמש נוכחי עם קישור בתוך קישור

```

const db = require('../models/index')
const Hashala = db.hashala
const AllBooks = db.allBooks

```

```

const Book = db.books
const getAll = async (req, res) => {
  const {id:user_id} = req.user
  const hashala = await Hashala.findAll(
    {where:{
      user_id:user_id
    },
    include : [{
      model: AllBooks,
      attributes:['id','book_id'],
      as: 'allBooks',
      include : [{
        model: Book,
        as: 'book',

      }]
    }]

  })
  if (!hashala?.length) {
    return res.status(400).json({ message: 'No hashala found' })
  }
  res.json(hashala)
}

module.exports = {getAll}

```

5. יצירת ROUTE מאובטח

hashalaRoutes.js

```

const express = require('express')
const router = express.Router()
const hashalaController = require("../controllers/hashalaController")
const verifyJWT = require("../middleware/verifyJWT")

router.use(verifyJWT)
router.route('/')
  .get(hashalaController.getAll)

```

```
module.exports = router
```

6. עדכון ROUTE בקובץ ראשי

```
app.use("/api/hashala", require("./routes/hashalaRoutes"));
```

והתוצאה

The screenshot shows a REST client interface with a GET request to `http://localhost:3600/api/hashala`. The response is displayed in JSON format, showing a nested object with user and book information.

```
1  [
2    {
3      "id": 2,
4      "allBooks_id": 2,
5      "user_id": 2,
6      "recived": true,
7      "recivedDate": "2022-01-01T00:00:00.000Z",
8      "createdAt": "2022-01-01T00:00:00.000Z",
9      "updatedAt": "2022-01-01T00:00:00.000Z",
10     "allBooks": {
11       "id": 2,
12       "book_id": 1,
13       "book": {
14         "id": 1,
15         "name": "זחות עלומה",
16         "picture": "",
17         "cateogry_id": 1,
18         "author_id": 1,
19         "createdAt": "2023-02-14T09:55:01.000Z",
20         "updatedAt": "2023-02-14T09:55:01.000Z"
21       }
22     }
23   }
24 ]
```