quiz3

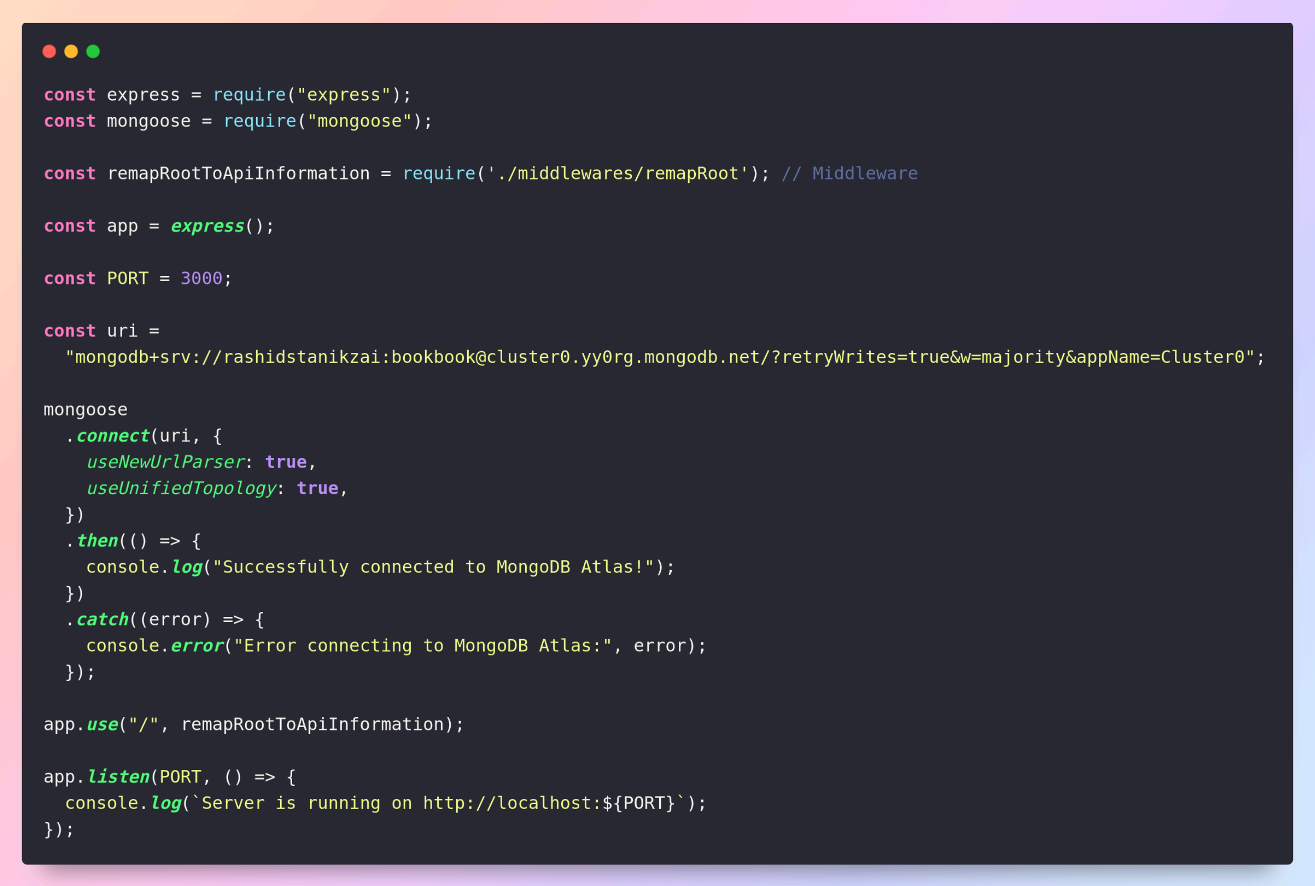
This is sample solution of the last Quiz and this is how your written answer should look like.

Important Points

- Order of the files was important
- The database connection should be made in `index.js/server.js/app.js`
- The server should run in `index.js/server.js/app.js`
- Remapping the `root` / to `api/information` should be made in `middlewares/<middle_ware_name>.js`
- The data should have returned in `contorlers/<your_contorler_name>.js`
- At least one database query should be performed in the controller.
- Proper error handling should be done

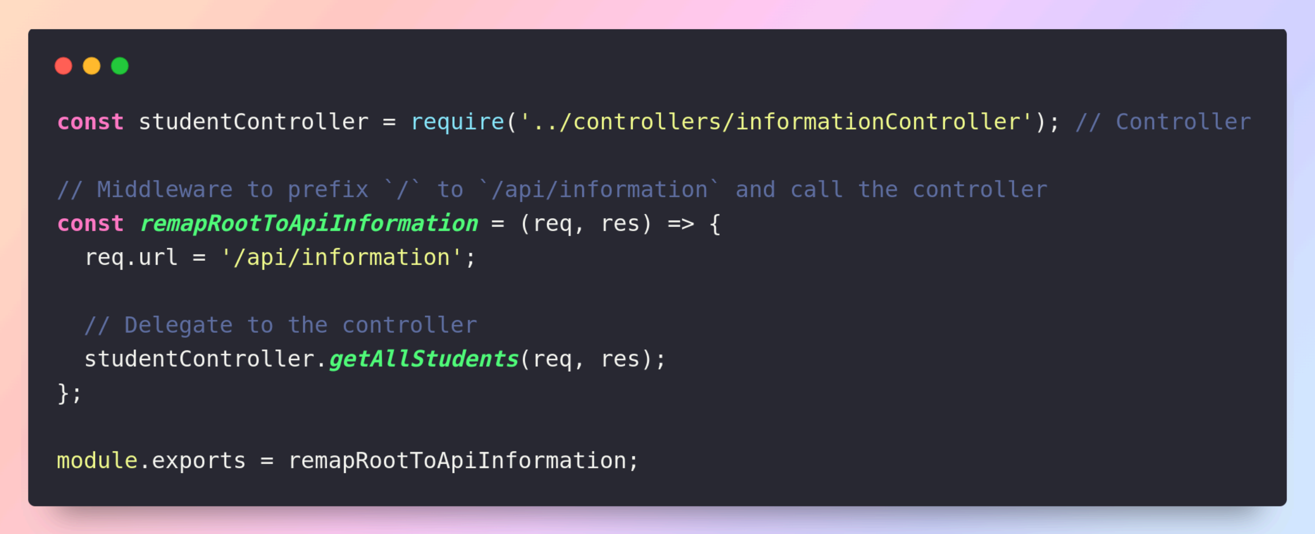
Sample Answer

`index.js/server.js/app.js`



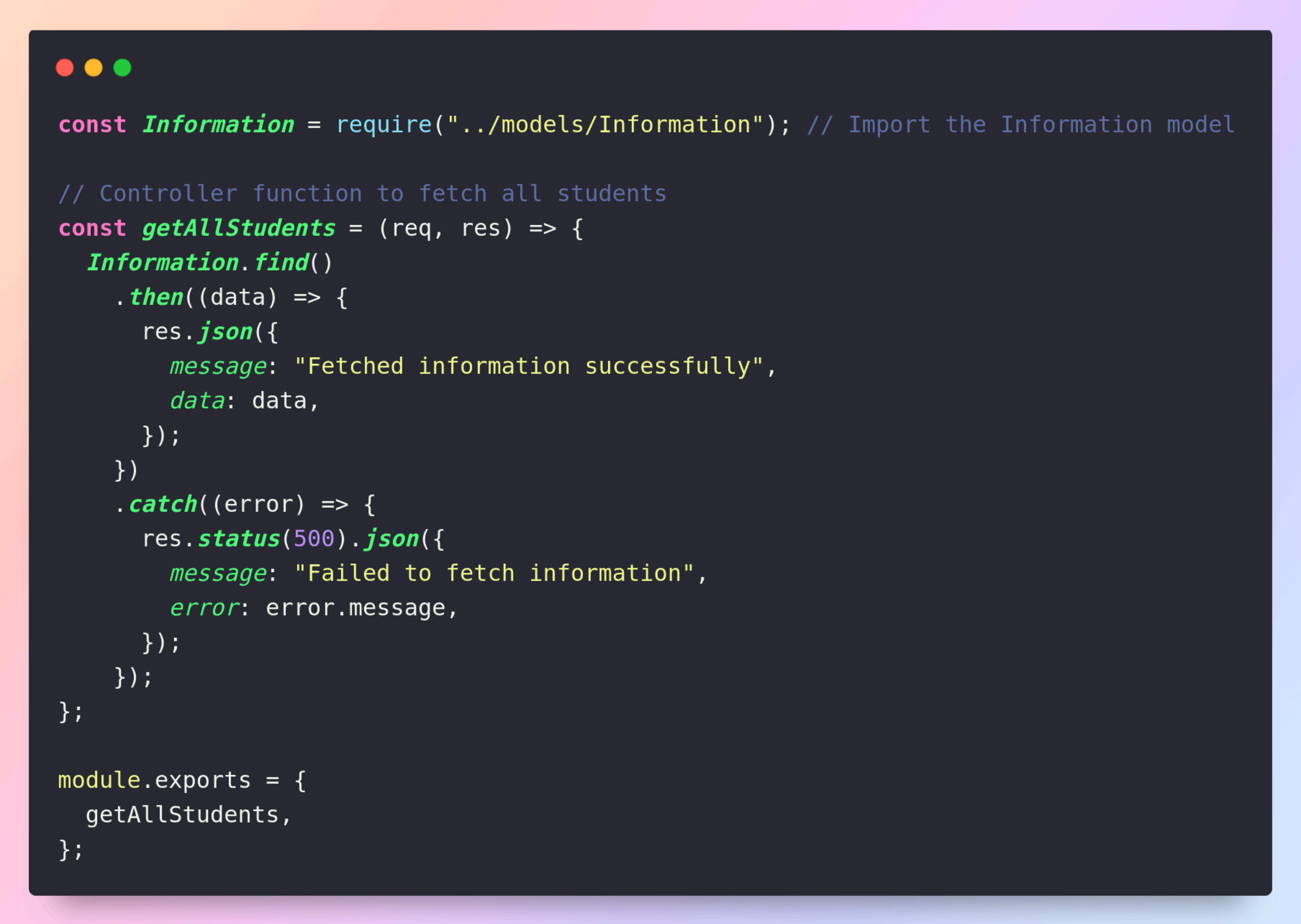
```javascript
const express = require("express");
const mongoose = require("mongoose");

const remapRootToApiInformation = require('./middlewares/remapRoot'); // Middleware

const app = express();

const PORT = 3000;

const uri =
  "mongodb+srv://rashidstanikzai:bookbook@cluster0.yy0rg.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0";

mongoose
  .connect(uri, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then(() => {
    console.log("Successfully connected to MongoDB Atlas!");
  })
  .catch((error) => {
    console.error("Error connecting to MongoDB Atlas:", error);
  });

app.use("/", remapRootToApiInformation);

app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

`middlewares/<middle_ware_name>.js`



```
const studentController = require('../controllers/informationController'); // Controller

// Middleware to prefix `/` to `/api/information` and call the controller
const remapRootToApiInformation = (req, res) => {
  req.url = '/api/information';

  // Delegate to the controller
  studentController.getAllStudents(req, res);
};

module.exports = remapRootToApiInformation;
```

`contorlers/<your_contorler_name>.js`



```
const Information = require("../models/Information"); // Import the Information model

// Controller function to fetch all students
const getAllStudents = (req, res) => {
  Information.find()
    .then((data) => {
      res.json({
        message: "Fetched information successfully",
        data: data,
      });
    })
    .catch((error) => {
      res.status(500).json({
        message: "Failed to fetch information",
        error: error.message,
      });
    });
};

module.exports = {
  getAllStudents,
};
```

Some Other Points

- It didn't have to look exactly like this but should follow this pattern
- The project file for the complete solution is provided in a zip file, you can run and see it for yourself.

- In the project I used Atlas to make it easy for you to run it in your own machines but you can connect it to your local machine database the same way.