MERN stack - Shopping Online
Lab 01

Contents

1 Preparation

1.1 MongoDB Atlas

- Sign up an account from MongoDB Atlas: `https://www.mongodb.com/atlas/database`

  - Menu Projects ⇒ New Project
    - Name your project: <project_name>
  - Menu Database ⇒ Create a free database
    - Cloud provider & Region: AWS + Singapore
    - Cluster name: <cluster>
  - Menu Database Access ⇒ Create a Database User
    - Authentication method: Username and Password
    - Password authentication: <db_user> and <db_pass>
  - Menu Network Access ⇒ Add an IP address
    - Access list entry: 0.0.0.0/0 (allow access from anywhere)

- Download and install the MongoDB Compass (GUI) from: `https://www.mongodb.com/try/download/compass`

  - New connection ⇒ copy the connection string:
    mongodb+srv://<db_user>:<db_pass>@<cluster>.mongodb.net/test
  - Create database: shoppingonline
    - Create collection: admins
      ⇒ Import JSON file from: `https://github.com/tsonkk/shoppingonline-resources/blob/main/mongodb/admins.json`
    - Create collection: categories
      ⇒ Import JSON file from: `https://github.com/tsonkk/shoppingonline-resources/blob/main/mongodb/categories.json`
    - Create collection: products
      ⇒ Import JSON file from: `https://github.com/tsonkk/shoppingonline-resources/blob/main/mongodb/products.json`

1.2 Hotmail

- Sign up an account from Microsoft: `https://signup.live.com`

  - New email: <email_user>@hotmail.com
  - Create password: <email_pass>

- Security tab: turn off the two-step verification

- Outlook application: review the welcome email

- SMTP settings from: `https://support.microsoft.com/en-us/office/pop-imap-and-smtp-settings-8361e398`

  - Server: smtp.office365.com
  - Port: 587
  - Encryption: STARTTLS

- Test an SMTP server:

```
$ telnet smtp.office365.com 587
```

2 MERN stack

2.1 Setting up

2.1.1 Installation

2.1.1.1 NodeJS

- Download and install the NodeJS from: `https://nodejs.org/en/download`

```
$ node --version
$ npm --version
```

2.1.1.2 Nodemon tool

- Install the Nodemon tool (use the sudo command for MacOS)

```
$ npm install nodemon --global
$ nodemon --version
```

2.1.1.3 ReactJS

- Installing create-react-app tool (use the sudo command for MacOS)

```
$ npm install create-react-app --global
$ create-react-app --version
```

2.1.1.4 Visual Studio Code IDE

- Download and install the Visual Studio Code from: `https://code.visualstudio.com/download`

2.1.2 Projects

```
|-- projectname
    |-- server
    |-- client-admin
    |-- client-customer
```

2.1.2.1 Server project

- Create server project

```
<server>$ npm init -y
```

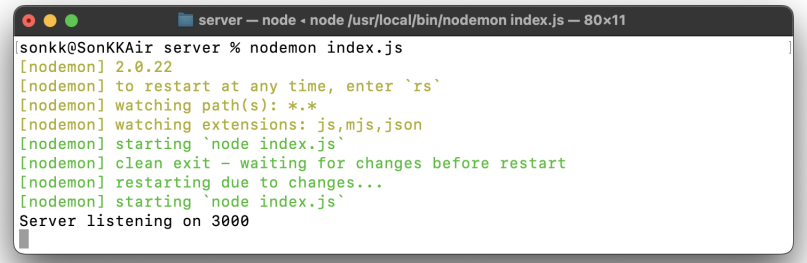
- Create 'server/index.js' file

```
//CLI: npm install express body-parser --save
const express = require('express');
const app = express();
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server listening on ${PORT}`);
});
// middlewares
const bodyParser = require('body-parser');
app.use(bodyParser.json({ limit: '10mb' }));
app.use(bodyParser.urlencoded({ extended: true, limit: '10mb' }));
// apis
app.get('/hello', (req, res) => {
  res.json({ message: 'Hello from server!' });
});
```

- Start server

```
<server>$ nodemon index.js
```

- Postman: (GET) http://localhost:3000/hello



2.1.2.2 Client-admin project

- Create client-admin project

```
<projectname>$ npx create-react-app client-admin
```

- Update 'client-admin/package.json' file

```
{
  ...,
  "homepage": "/admin",
  "proxy": "http://localhost:3000"
}
```

- Update 'client-admin/src/App.js' file

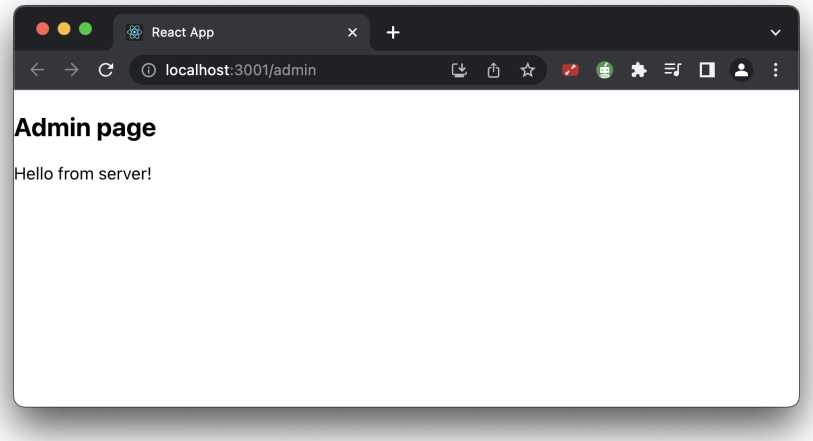
```
//CLI: npm install axios --save
import axios from 'axios';
import React, { Component } from 'react';

class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      message: 'Loading...'
    };
  }
  render() {
    return (
      <div>
        <h2>Admin page</h2>
        <p>{this.state.message}</p>
      </div>
    );
  }
  componentDidMount() {
    axios.get('/hello').then((res) => {
      const result = res.data;
      this.setState({ message: result.message });
    });
  }
}
export default App;
```

- Start client-admin

```
<client-admin>$ npm start
```

- Browser: http://localhost:3001/admin



2.1.2.3 Client-customer project

- Create client-customer project

```
<projectname>$ npx create-react-app client-customer
```

- Update 'client-customer/package.json' file

```
{
  ...,
  "homepage": "/",
  "proxy": "http://localhost:3000"
}
```

- Update 'client-customer/src/App.js' file

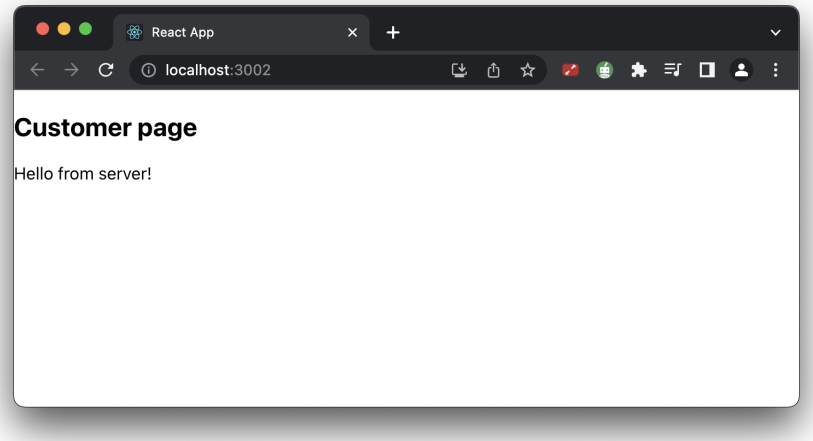
```
//CLI: npm install axios --save
import axios from 'axios';
import React, { Component } from 'react';

class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      message: 'Loading...'
    };
  }
  render() {
    return (
      <div>
        <h2>Customer page</h2>
        <p>{this.state.message}</p>
      </div>
    );
  }
  componentDidMount() {
    axios.get('/hello').then((res) => {
      const result = res.data;
      this.setState({ message: result.message });
    });
  }
}
export default App;
```

- Start client-customer

```
<client-customer>$ npm start
```

- Browser: http://localhost:3002


React App
localhost:3002
Customer page
Hello from server!