**Báo cáo**

**Sơ lược về đề tài và các thiết lập ban đầu**

Mục lục

[Mục lục 1](#_Toc141456525)

[Danh mục bảng, hình ảnh 2](#_Toc141456526)

[I/ SƠ LƯỢC VỀ ĐỀ TÀI 4](#_Toc141456527)

[1/ Tên đề tài và các chức năng 4](#_Toc141456528)

[2/ Các công nghệ sử dụng trong dự án 4](#_Toc141456529)

[II/ THIẾT LẬP DỰ ÁN 4](#_Toc141456530)

[1/ Thiết lập node js và mongodb 4](#_Toc141456531)

[2/ Thiết lập react native 11](#_Toc141456532)

[III/ SƠ LƯỢC VỀ CÁC CÔNG NGHỆ SỬ DỤNG TRONG DỰ ÁN 18](#_Toc141456533)

[1/ Sơ lược về React-native 18](#_Toc141456534)

[1.1/ React-native là gì 18](#_Toc141456535)

[1.2/Props và state 19](#_Toc141456536)

[1.3/ Các components thông dụng: 21](#_Toc141456537)

[2/ Sơ lược về nodeJS 23](#_Toc141456538)

[2.1/ NodeJs là gì? 23](#_Toc141456539)

[2.2/ NPM(Node Package Manager) 24](#_Toc141456540)

[3/ Sơ lược về mongodb 24](#_Toc141456541)

[1.1/ Mongodb là gì? 24](#_Toc141456542)

[2.2/ Một số kiểu dữ liệu trong mongodb 25](#_Toc141456543)

[IV/ ỨNG DỤNG THƯƠNG MẠI ĐIỆN TỬ CỬA HÀNG SÁCH VỚI REACT-NATIVE VÀ NODE JS 26](#_Toc141456544)

[1/ Chi tiết chức năng 26](#_Toc141456545)

[2/ Bảng cơ sở dữ liệu 26](#_Toc141456546)

Danh mục bảng, hình ảnh

**Danh mục hình ảnh**

[Hình 1 Kết quả sau khi chạy lệnh “npm init” 5](#_Toc141456487)

[Hình 2 Tạo file index để thiết lập server 5](#_Toc141456488)

[Hình 3 sửa script để có thể chạy dự án 5](#_Toc141456489)

[Hình 4 Kết quả khi cài những công cụ cần thiết để thực hiện dự án 6](#_Toc141456490)

[Hình 5 Khai báo các công cụ và thiết lập server 6](#_Toc141456491)

[Hình 6 Kết quả chạy dự án 6](#_Toc141456492)

[Hình 7 Trang chủ mongodb 7](#_Toc141456493)

[Hình 8 Tạo dự án mới trên mongodb 7](#_Toc141456494)

[Hình 9 Xác nhận tạo dự án 8](#_Toc141456495)

[Hình 10 Màn hình sau khi đã tạo hoàn thành dự án 8](#_Toc141456496)

[Hình 11 Chọn gói CSDL 9](#_Toc141456497)

[Hình 12 Csdl sau khi được tạo 9](#_Toc141456498)

[Hình 13 Chọn phương thức kết nối csdl 10](#_Toc141456499)

[Hình 14 Link để kết nối với csdl 10](#_Toc141456500)

[Hình 15 Kết nối csdl và server 10](#_Toc141456501)

[Hình 16 trang chủ nodejs 11](#_Toc141456502)

[Hình 17 Các phiên bản của JDK 12](#_Toc141456503)

[Hình 18 Trang web cài đặt android studio 12](#_Toc141456504)

[Hình 19 chọn thiết bị để tạo máy ảo 13](#_Toc141456505)

[Hình 20 Chọn phiên bản android 13](#_Toc141456506)

[Hình 21 Thiết lập các cài đặt cho thiết bị ảo 14](#_Toc141456507)

[Hình 22 Máy ảo khi đã hoàn tất 14](#_Toc141456508)

[Hình 23 Khai báo biến môi trường JAVA và ANDROID\_HOME 15](#_Toc141456509)

[Hình 24 Kết quả chạy xong lệnh “npx create-expo-app AwesomeProject” 15](#_Toc141456510)

[Hình 25 Cd vào thư mục AwesomeProject 15](#_Toc141456511)

[Hình 26 Kết quả chạy lệnh “npx expo start”. 16](#_Toc141456512)

[Hình 27 Thiết bị ảo khi chạy project 16](#_Toc141456513)

[Hình 28 Đoạn code đã được chỉnh sửa 17](#_Toc141456514)

[Hình 29 Kết quả khi chỉnh sửa dòng code 17](#_Toc141456515)

[Hình 30 Giá trị number={index} là props 20](#_Toc141456516)

[Hình 31 Component nhận giá trị props để xử lý 20](#_Toc141456517)

[Hình 32 Khai báo state 21](#_Toc141456518)

[Hình 33 Hàm chuyển trạng thái 21](#_Toc141456519)

[Hình 34 component text 21](#_Toc141456520)

[Hình 35 component image 22](#_Toc141456521)

[Hình 36 component button 22](#_Toc141456522)

[Hình 37 Component TouchableOpacity 22](#_Toc141456523)

[Hình 38 Nhập module vào file package.json 24](#_Toc141456524)

**Danh mục bảng**

[Bảng 1 Ưu và nhược đểm của react-native 19](#_Toc141456479)

[Bảng 2 Ưu và nhược điểm của nodejs 24](#_Toc141456480)

[Bảng 3 Ưu và nhược điểm của mongo db 25](#_Toc141456481)

[Bảng 4 csdl Book 27](#_Toc141456482)

[Bảng 5 csdl author 27](#_Toc141456483)

[Bảng 6 Csdl user 28](#_Toc141456484)

I/ SƠ LƯỢC VỀ ĐỀ TÀI

1/ Tên đề tài và các chức năng

\*Tên đề tài: Ứng dụng thương mại điện tử cửa hàng sách với React-native và node js

\*Các chức năng chủ yếu:

* Đăng nhập/đăng ký
* Giỏ hàng
* Danh mục sách
* Phân loại sách
* Tìm kiếm sản phẩm

2/ Các công nghệ sử dụng trong dự án

* Front end: react-native
* Back end: nodejs + mongodb

3/ Lý do chọn đề tài

* Hiện nay nhu cầu mua sắm trên các nền tảng thương mại điện tử ngày càng phổ biến và trở thành thói quen đối với mọi người, những cửa hàng bán sách truyền thống cũng dần dần nắm bắt xu hướng để đáp ứng nhu cầu cho khách hàng. Vì vậy, để đáp ứng xu hướng đó ứng dụng thương mại điện tử cửa hàng sách sẽ giúp những khách hàng có thể dễ dàng tiếp cận với những sản phẩm mà họ muốn mua dễ dàng hơn bao giờ hết, chỉ với những thao tác đơn giản trên ứng dụng và người dùng có thể biết được chính xác giá cả, thông tin, về sách mà có thể đưa ra quyết định mua.

II/ THIẾT LẬP DỰ ÁN

1/ Thiết lập node js và mongodb

\* Để có thể viết một api cho dự án bằng nodejs thì yêu cầu thiết bị phải được cài đặt nodejs. Nếu muốn cài đặt hãy truy cập trang chủ của nodejs “nodejs.org” để có thể cài đặt.

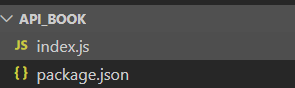
* Đầu tiên, hãy thiết lập một file dự án và làm theo các bước sau:
* Lệnh: “npm init” khi chạy lệnh này sẽ được một file package.json.

A screen shot of a computer

Description automatically generated

Hình 1 Kết quả sau khi chạy lệnh “npm init”

* Tạo thêm file index.js để thiết lập server.



Hình 2 Tạo file index để thiết lập server

* Sửa phần script trong file package.json thành: "start": "node index.js"

A computer screen shot of a computer code

Description automatically generated

Hình 3 sửa script để có thể chạy dự án

* Tiếp theo sẽ chạy các lệnh: ‘npm install express mongoose cors body-parser morgan dotenv’ để cài đặt những công cụ cần thiết cho dự án.

A screen shot of a computer

Description automatically generated

Hình 4 Kết quả khi cài những công cụ cần thiết để thực hiện dự án

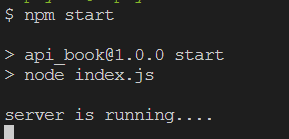
* Thiết lập server: để dùng các công cụ như express, cors, mongoose, body-paser, morgan, dotenv.

A screen shot of a computer program

Description automatically generated

Hình 5 Khai báo các công cụ và thiết lập server

* Và chạy thử server bằng lệnh npm start

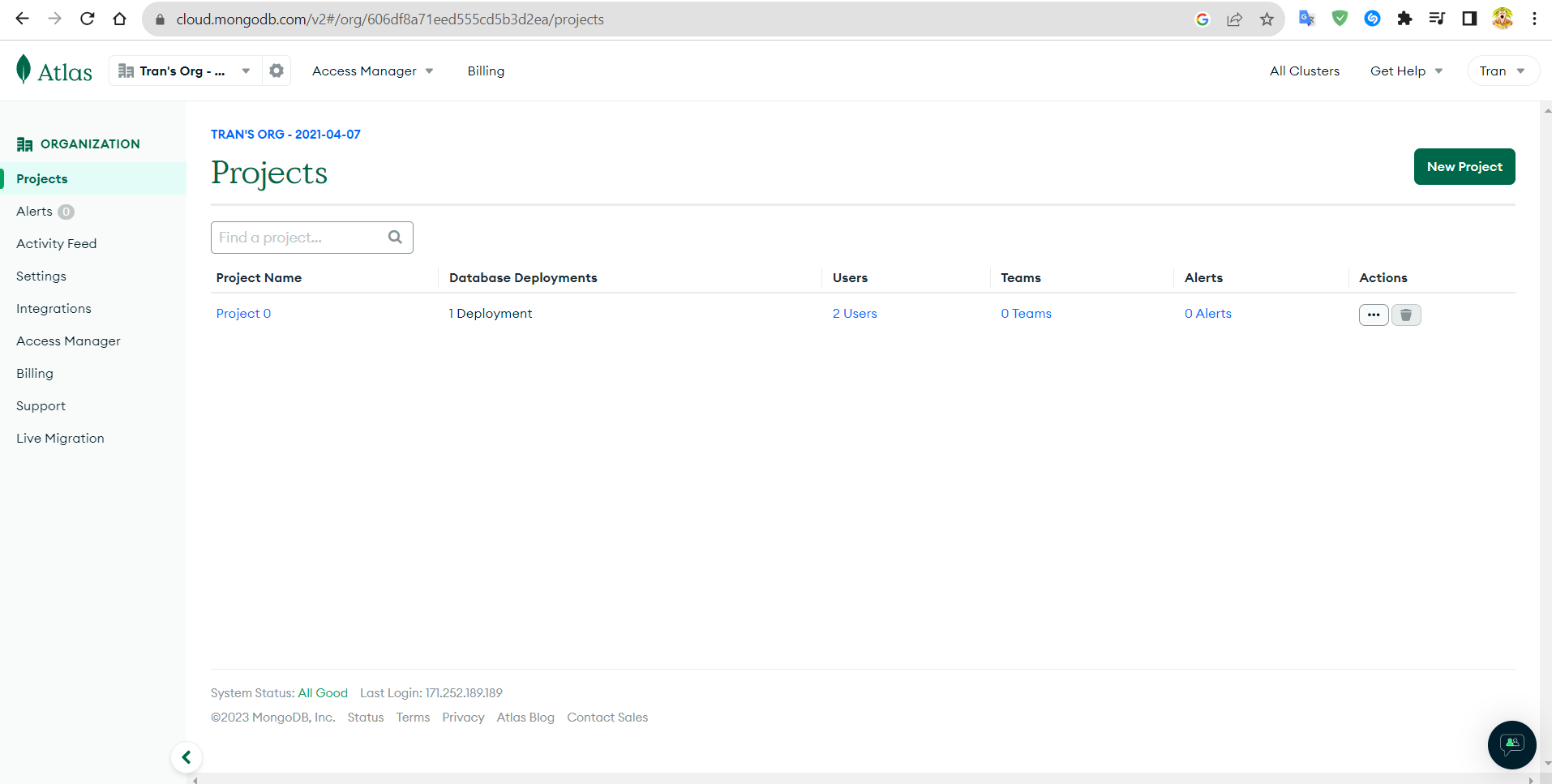


Hình 6 Kết quả chạy dự án

* Và kết quả như vậy có nghĩa server đã hoạt động

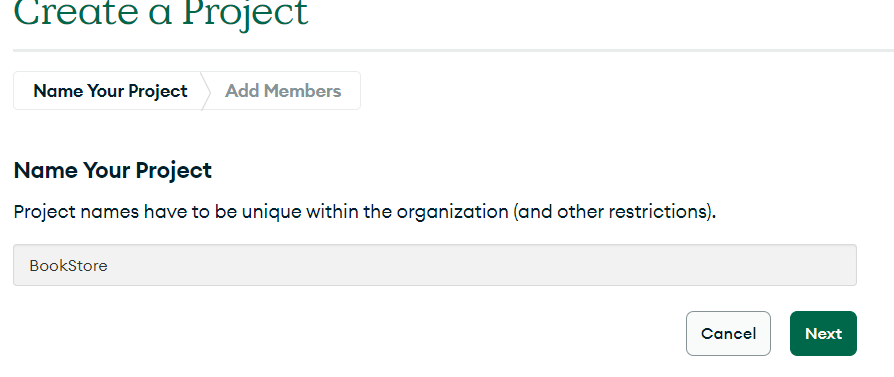
\* Sau khi đã thiết lập server nodejs thì sẽ thiết lập và kết nối cơ sở dữ liệu mongodb:

* Truy cập “cloud.mongodb.com”

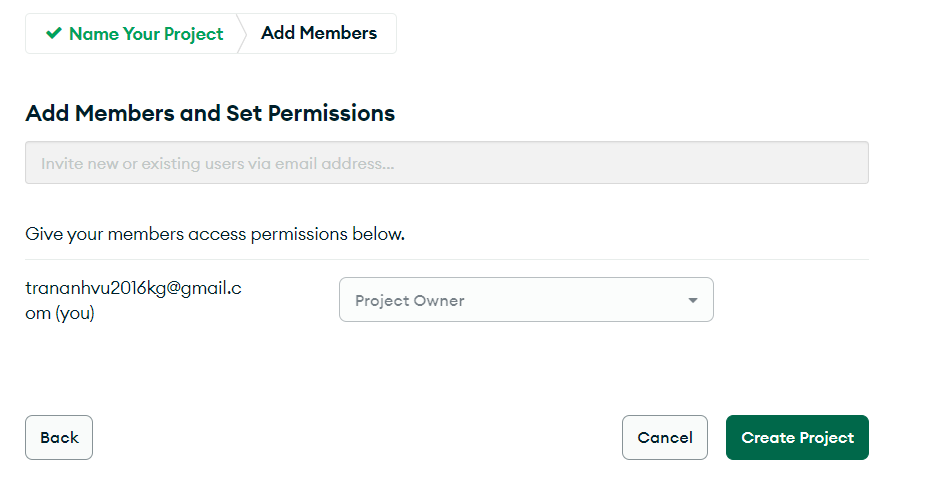


Hình 7 Trang chủ mongodb

* Chọn New Project => Đặt tên project

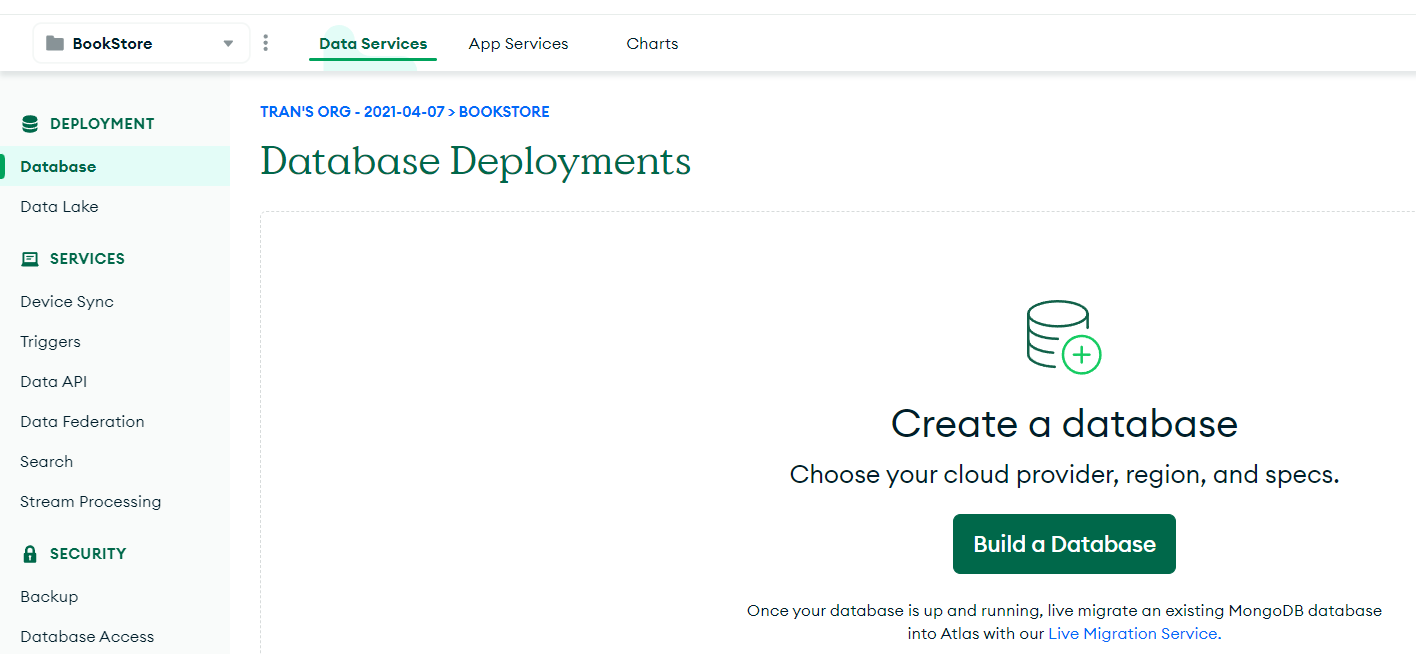


Hình 8 Tạo dự án mới trên mongodb



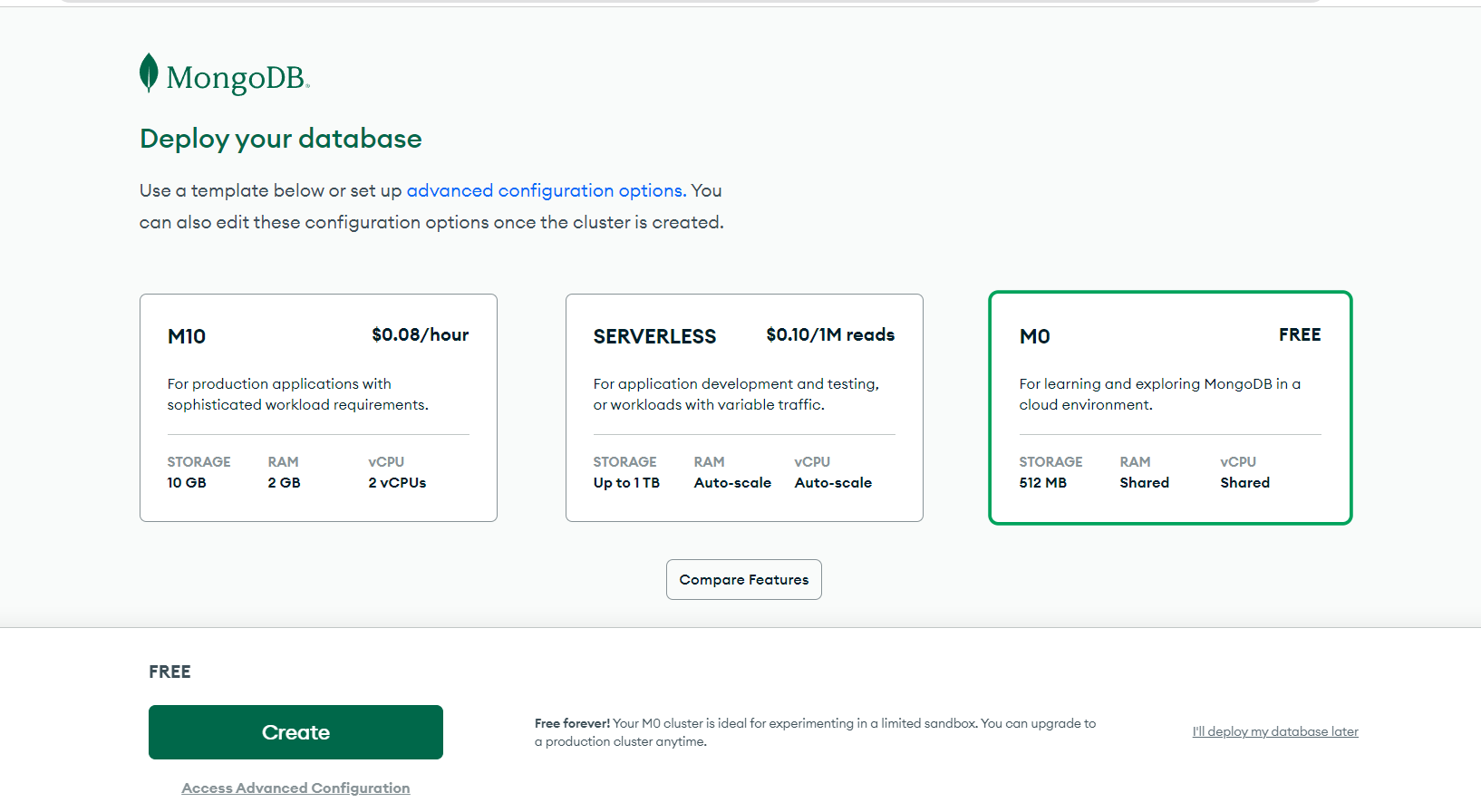
Hình 9 Xác nhận tạo dự án

* Sau khi đã tạo được dự án:



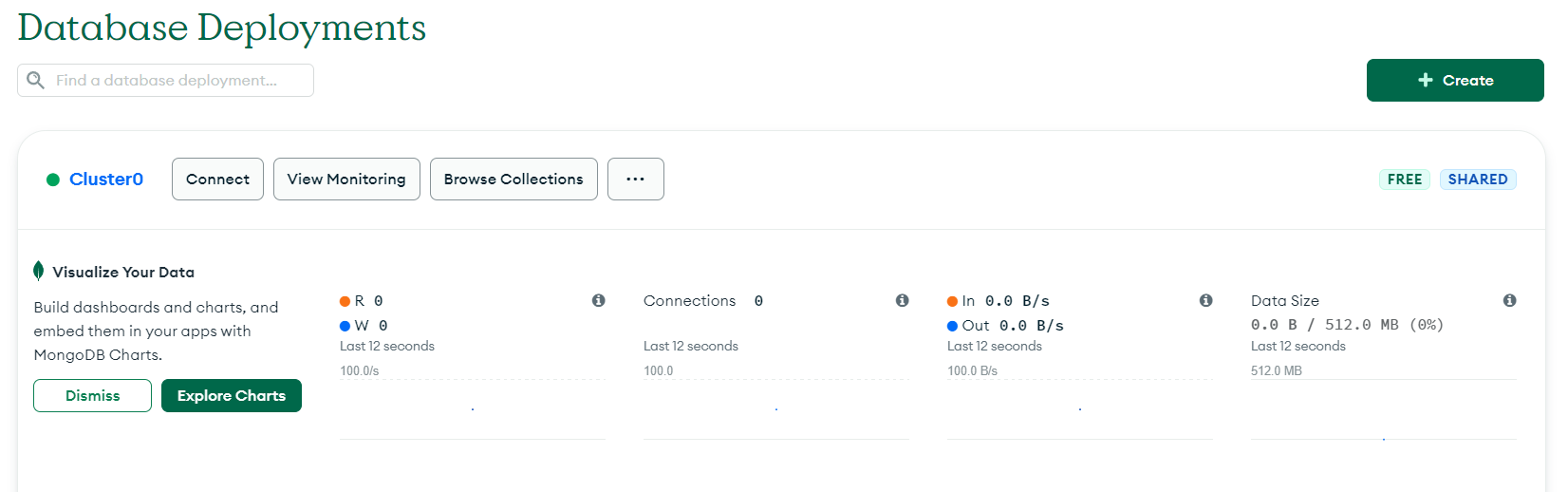
Hình 10 Màn hình sau khi đã tạo hoàn thành dự án

* Nhấn build a database => chọn gói free => Create:



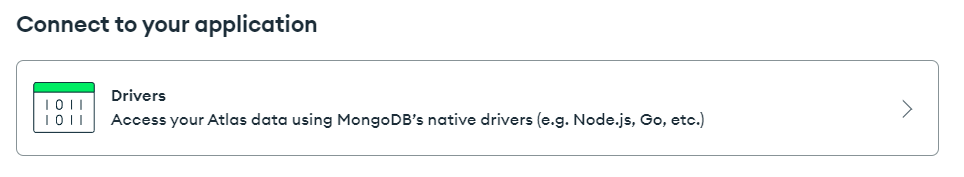
Hình 11 Chọn gói CSDL

* Sau đó thiết lập user và thêm ip address và nhấn finish. Sau đó đợi csdl được tạo thì hoàn thành.



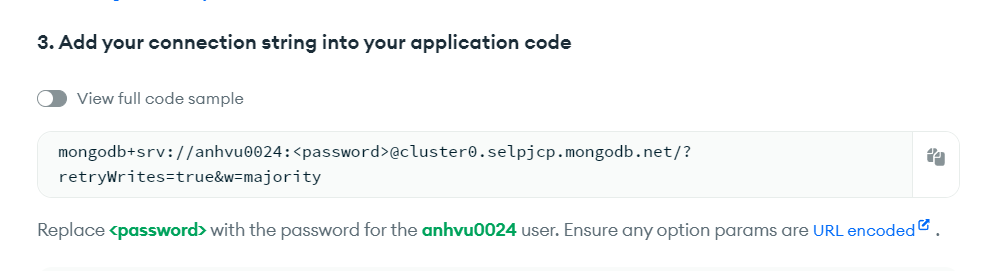
Hình 12 Csdl sau khi được tạo

* Để có thể kết nối với csdl thì chúng ta chọn “connect to your application”.



Hình 13 Chọn phương thức kết nối csdl

* Và tiếp theo chúng ta chỉ cần copy link:” mongodb+srv://anhvu0024:<password>@cluster0.selpjcp.mongodb.net/?retryWrites=true&w=majority” là có thể sử dụng.



Hình 14 Link để kết nối với csdl

* Thiết đặt kết nối với mongodb trên server

A screen shot of a computer program

Description automatically generated

Hình 15 Kết nối csdl và server

* Vậy chúng ta đã hoàn thành bước thiết lập server ban đầu để có thể phát triển api cho ứng dụng.

2/ Thiết lập react native

Nếu muốn phát triển ứng dụng react-native thì bắt buộc trong máy phải được cài NodeJs. Vì vậy chúng ta sẽ phải tải nodejs đầu tiên và hiện nay các phiên bản mới của nodejs đề hỗ trợ react-native vì thế chúng ta sẽ chọn phiên bản ổn định nhất là 16 hoặc 18 để cài đặt.

A screenshot of a computer

Description automatically generated

Hình 16 trang chủ nodejs

Tiếp theo, chúng ta sẽ cài môi trường phát triển ứng dụng, vì em phát triển ứng dụng cho android nên em sẽ cài môi trường hỗ trợ cho việc phát triển ứng dụng android:

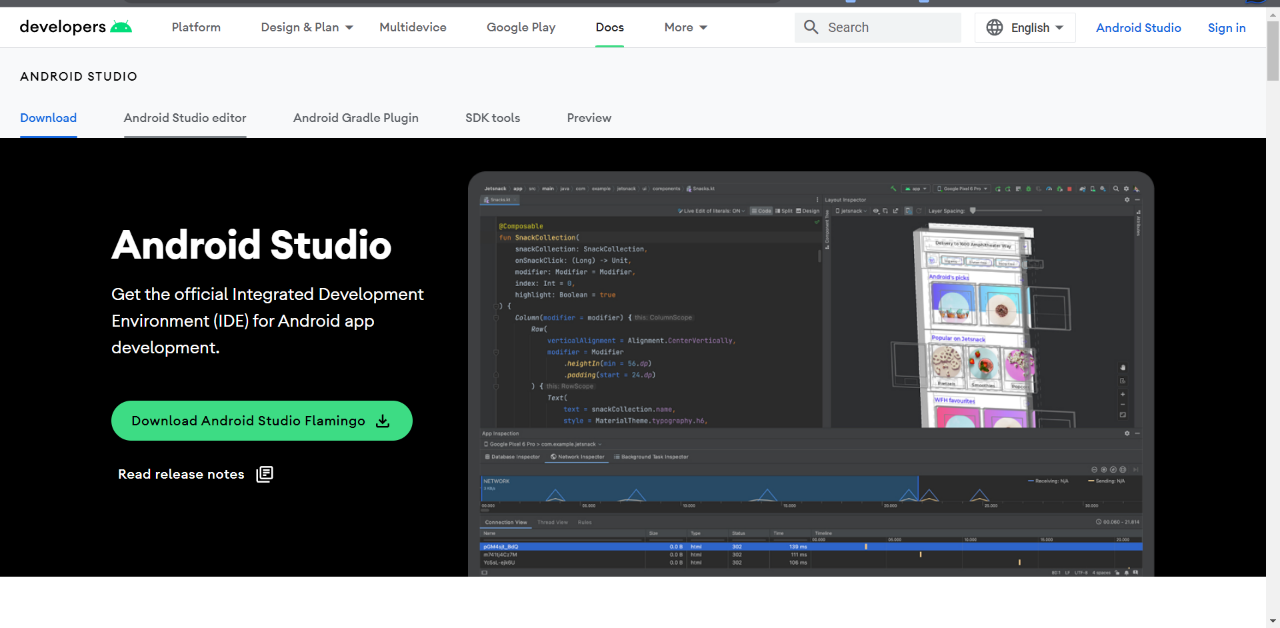
* Đầu tiên em sẽ cài đặt JDK:

A screenshot of a computer

Description automatically generated

Hình 17 Các phiên bản của JDK

* Cài đặt android studio:



Hình 18 Trang web cài đặt android studio

Sau khi cài android studio thì chúng ta sẽ cần tạo một máy ảo để tiện cho việc chạy ứng dụng(nếu có một chiếc điện thoại android thật thì hãy bỏ qua bước này):

* Ở bước này có thể chọn thiết bị với kích thước màn hình, độ phân giải và số dpi.

Graphical user interface

Description automatically generated

Hình 19 chọn thiết bị để tạo máy ảo

Bước này thì có thể chọn phiên bản android cho thiết bị android.

A screenshot of a computer

Description automatically generated

Hình 20 Chọn phiên bản android

Ở bước này có thể chọn chế độ hiển thị, và các thiết lập khác như camera(sử dụng từ webcam của laptop hoặc tùy chọn còn lại), chế độ khởi động của thiết bị(khởi động lại thiết bị trong mỗi lần tắt hoặc mở máy nhanh. Trong thiết đặt này em khuyến khích nên dùng khởi động lại thiết bị trong mỗi lần mở máy vì có những lúc máy ảo sẽ bị đơ hoặc lỗi UI not responsing thì chỉ còn cách khởi động lại thì mới dùng tiếp được, còn tùy chọn mở máy nhanh thì khi dính lỗi đó chỉ có cách là xóa cài lại máy ảo).

Graphical user interface, application

Description automatically generated

Hình 21 Thiết lập các cài đặt cho thiết bị ảo

* Sau khi tạo xong máy ảo, chọn run máy ảo và sau đó cần ít thời gian để chạy lần đầu.

A screen shot of a cell phone

Description automatically generated

Hình 22 Máy ảo khi đã hoàn tất

* Thêm biến môi trường JAVA và ANDROID\_HOME: là địa chỉ thư mục jdk và android sdk.

A screenshot of a computer

Description automatically generated with medium confidence

Hình 23 Khai báo biến môi trường JAVA và ANDROID\_HOME

\*Khởi tạo dự án bằng expo go:

Đầu tiên khi lên web reactnative.dev thì chúng ta sẽ thấy có hai cách tạo một dự án đó là dùng expo go và cli

Vì vậy, đầu tiên chúng ta sẽ khởi tạo một dự án bằng expo go(ở trên web sẽ có hai tùy chọn khởi tạo bằng lên yarn và npm thì ở đây em sẽ dùng lệnh npm do hai lệnh cũng giống nhau):

* Đầu tiên sẽ dùng lệnh: “npx create-expo-app AwesomeProject” ở đây “AwesomeProject” sẽ là tên của dự án khi được tạo hoàn thành, cho nên chúng ta có thể tùy biến nó theo ý bản thân.

A screenshot of a computer

Description automatically generated with low confidence

Hình 24 Kết quả chạy xong lệnh “npx create-expo-app AwesomeProject”

* Di chuyển vào thư mục AwesomeProeject bằng lệnh “cd AwesomeProject”.

Text

Description automatically generated

Hình 25 Cd vào thư mục AwesomeProject

* Tiếp theo chạy lệnh “npx expo start”.

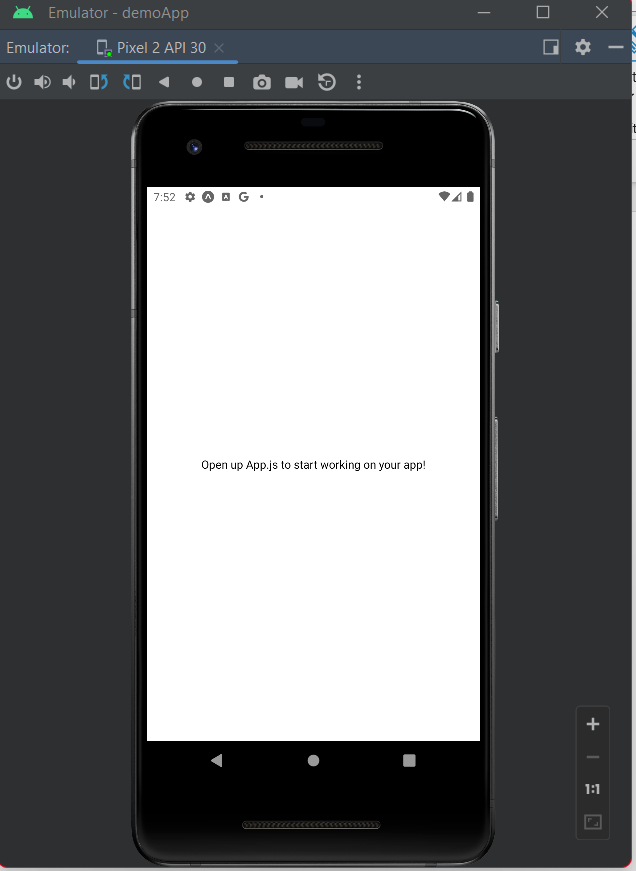
Text

Description automatically generated

Hình 26 Kết quả chạy lệnh “npx expo start”.

- Các tùy chọn để có thể chạy trên các nền tảng khác nhau như: Nhấn a(mở thiết bị android), j(mở debugger), r(chạy lại app),….

- Ở đây em chọn nhấn a để có thể chạy trên thiết bị android ảo(mở sẵn thiết bị android ảo).



Hình 27 Thiết bị ảo khi chạy project

* Sau khi chạy được trên máy ảo ta có thể dùng ứng dụng visual studio code mở thư mục dự án, khi đó ta có thể chỉnh sửa và viết thêm trên giao diện app.

A screen shot of a computer program

Description automatically generated

Hình 28 Đoạn code đã được chỉnh sửa

A picture containing text, screenshot, monitor, electronics

Description automatically generated

Hình 29 Kết quả khi chỉnh sửa dòng code

* Vậy qua quá trình trên chúng ta đã hoàn thành việc cài đặt môi trường của react native trên hệ điều hành window và khởi tạo ứng dụng đầu tiên bằng expo go.

\* Khởi tạo dự án theo CLI

Và tiếp theo đó là khởi tạo dự án bằng CLI:

* Hiện tại khi khởi tạo dự án bằng lệnh: “npx react-native@latest init AwesomeProject” tức là chúng ta sẽ khởi tạo với phiên bản mới nhất của react-native thì khi tạo ra thư mục và các file sẽ mặc định là dạng typescript(.tsx). Vì vậy, nếu muốn khởi tạo ra file javascript(.js) thì có thể dùng lệnh: “npx react-native@X.XX.X init AwesomeProject --version X.XX.X” để có thể chọn phiên bản cũ hơn để tạo ra file .js.
* Và nếu khởi tạo theo cách này để phát triển dự án cho android thì chỉ có thể chạy ứng dụng qua máy thật(cắm usb) hoặc máy ảo cài lên máy, chứ không thể chạy qua expo app được.
* Và lệnh để khởi chạy ứng dụng là :”npx react-native start” hoặc “npm start”. Tiếp theo đó nhấn “a” để chạy ứng dụng trên android hoặc lệnh” npx react-native run-android”.

III/ SƠ LƯỢC VỀ CÁC CÔNG NGHỆ SỬ DỤNG TRONG DỰ ÁN

1/ Sơ lược về React-native

1.1/ React-native là gì

* React-native là một framework được tạo ra bởi facebook vào năm 2015 và hiện tại đang duy trì phát triển đến hiện tại, framework này giúp chúng ta có thể phát triển giao diện ứng dụng trên hai nền tảng ios và android một cách nhanh chóng bằng ngôn ngữ lập trình rất phổ biến đó là javascript.
* Và như bao framework hay bất kỳ những ngôn ngữ lập trình khác thì react-native cũng có những ưu điểm và nhược điểm khác nhau:

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| - Cho phép chúng ta sử dụng kiến thức về ngôn ngữ javascript để xây dựng ứng dụng di động(ios và android).  - Có thể chia sẻ hầu hết mã trên nhiều nền tảng khác nhau, điều này giúp giảm thời gian cho chúng ta khi phát triển ứng dụng di động trên các nền tảng của di động khác nhau.  - Cộng đồng sử dụng lớn vây quanh Reactjs và React native. Vì thế, chúng ta có thể dễ dàng nhận được các câu trả lời và hướng dẫn khi gặp khó khăn. | - Nếu muốn tạo ra các tính năng gốc không có sẵn(Native Components) thì phải viết một số mã nền tảng cụ thể.  - Không thể xây dựng các ứng dụng của IOS trên hệ điều hành Window và Linux được. |

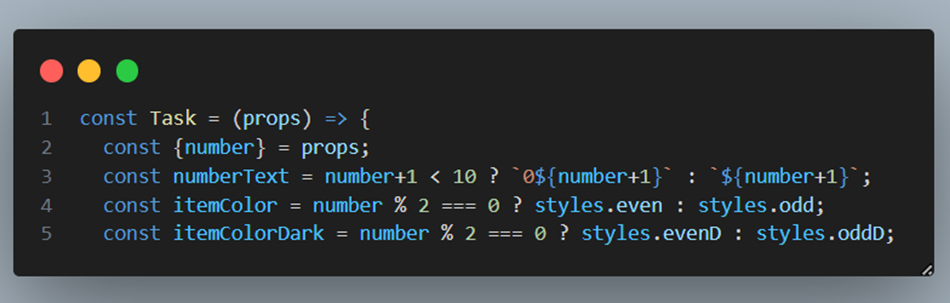
Bảng 1 Ưu và nhược đểm của react-native

1.2/Props và state

\***Props(Properties):** có thể hiểu là việc trừu tượng hóa các component để có thể sử dụng lại các component đó ở nhiều vị trí khác nhau ở trong ứng dụng. Các component con sẽ nhận props từ các component cha và đọc giá trị ra, lưu ý là không được thay đổi giá trị của props.



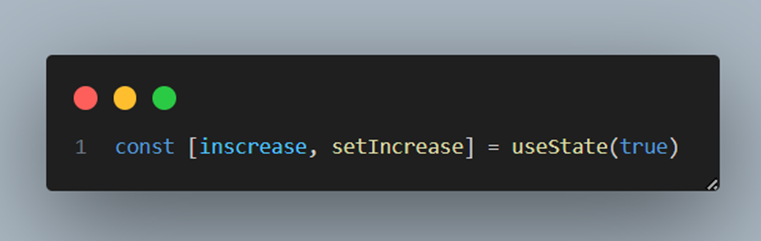
Hình 30 Giá trị number={index} là props



Hình 31 Component nhận giá trị props để xử lý

Như hai ảnh trên ta thấy giá trị number ở Hình 3 được gán bằng giá trị của index và ở Hình 4 trong component Task thì giá trị props được lấy ra để xử lý và khi đó giá trị của props là bằng với giá trị index được lấy trong Hình 3.

\* **State:** là biến điều khiển trạng thái của một component, mỗi lần state thay đổi thì trạng thái của component sẽ thay đổi và được render lại. Để dễ hiểu hơn em sẽ lấy một ví dụ đơn giản như sau:



Hình 32 Khai báo state

A screen shot of a computer

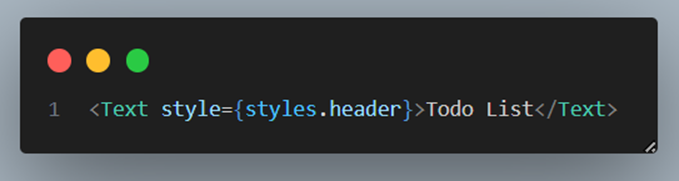
Description automatically generated with medium confidence

Hình 33 Hàm chuyển trạng thái

Như ví dụ trên ban đầu chúng ta sẽ khai báo usestate với hai tham số inscrease, setIncrease và giá trị state ban đầu là ‘true’. Tiếp theo đó, hàm handleSortById sẽ được viết nhằm mục đích khi ta gán hàm handleSortById vào một nút bấm Sort thì khi nhấm vào nút bấm sẽ kích hoạt hàm và khi đó trạng thái sẽ được đặt là ‘false’ để chuyển trạng thái cho một chức năng cụ thể nào đó.

1.3/ Các components thông dụng:

**Text**: dùng để hiển thị 1 message lên màn hình. Có thể sử dụng text cố định hoặc in nội dung của một biến lên màn hình.



Hình 34 component text

**Image**: dùng để hiển thị ảnh trên màn hình. Có 3 cách hiển thị ảnh: hiển thị từ local, hiển thị ảnh bằng URL, và hiển thị ảnh bằng Base 64.

A picture containing text, businesscard, screenshot, font

Description automatically generated

Hình 35 component image

**Button**: component thường dùng để tạo cái nút bấm dùng để kích hoạt một sự kiện nào đó.

A screen shot of a computer

Description automatically generated with medium confidence

Hình 36 component button

**TouchableOpacity**: component này có thể sử dụng giống như component button(onPress) nhưng có thể tùy biến các thành phần hoặc hình dạng đa dạng hơn.

A screen shot of a computer code

Description automatically generated with low confidence

Hình 37 Component TouchableOpacity

**ScrollView**: component này cho phép tạo một khoảng hiển thị giới hạn bởi chiều cao và rộng của chính nó, nhưng các thành phần con được chưa bên trong có thể được cuộn vô hạn. Có thể tùy chỉnh cuộn dọc và cuộn ngang.

2/ Sơ lược về nodeJS

2.1/ NodeJs là gì?

Node.js được phát triển bởi Ryan Dahl, một kỹ sư phần mềm tại Google, và được phát hành lần đầu vào năm 2009. Node.js nhanh chóng trở nên phổ biến và hiện được sử dụng bởi một số công ty lớn nhất thế giới, bao gồm Netflix, Uber và eBay.

Node.js là một nền tảng phần mềm mã nguồn mở, đa nền tảng, được thiết kế để tạo các ứng dụng web có thể mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, sử dụng kỹ thuật điều khiển theo sự kiện, nhập/xuất không đồng bộ để tối thiểu tổng chi phí và tối đa khả năng mở rộng.

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| - Node.js sử dụng mô hình lập trình không đồng bộ, cho phép nó xử lý nhiều yêu cầu đồng thời mà không bị chặn. Điều này khiến nó trở thành một lựa chọn tốt cho việc xây dựng các ứng dụng web có tính năng cao.  - Linh hoạt: Node.js có thể được sử dụng để tạo nhiều loại ứng dụng web, bao gồm ứng dụng web động, ứng dụng web thời gian thực và ứng dụng web di động.  - Cộng đồng lớn: Node.js có một cộng đồng lớn và tích cực, với nhiều tài nguyên có sẵn để giúp bạn học cách sử dụng Node.js.  - Được sử dụng rộng rãi: Node.js được sử dụng bởi một số công ty lớn nhất thế giới, bao gồm Netflix, Uber và eBay.  - Có nhiều thư viện và công cụ: Có rất nhiều thư viện và công cụ có sẵn cho Node.js, giúp bạn tiết kiệm thời gian và công sức khi phát triển các ứng dụng web. | - Có thể tốn nhiều tài nguyên: Node.js sử dụng nhiều tài nguyên hơn một số nền tảng máy chủ web khác  - Node.js có thể khó học nếu bạn chưa quen với JavaScript hoặc lập trình hướng sự kiện.  - Node.js không phù hợp với tất cả các loại ứng dụng. Ví dụ, Node.js không phải là lựa chọn tốt cho việc phát triển các ứng dụng web yêu cầu xử lý lượng lớn dữ liệu. |

Bảng 2 Ưu và nhược điểm của nodejs

2.2/ NPM(Node Package Manager)

- Là một kho lưu trữ mã nguồn mở cho các mô-đun Node.js. Các mô-đun là các tập hợp các hàm và mã có thể được sử dụng trong các ứng dụng Node.js. NPM cho phép bạn cài đặt và quản lý các mô-đun từ kho lưu trữ NPM.

* Để cài đặt một mô-đun, có thể sử dụng lệnh npm install. Ví dụ: để cài đặt mô-đun express, sẽ sử dụng lệnh sau: “npm install express”
* Sau khi mô-đun được cài đặt, có thể sử dụng nó trong ứng dụng của mình bằng cách nhập nó. Ví dụ: để nhập mô-đun express trong ứng dụng của, chúng ta sẽ thêm dòng sau vào tệp package.json:

**A black background with white text

Description automatically generated**

Hình 38 Nhập module vào file package.json

* Sau đó, có thể nhập mô-đun bằng cách sử dụng cú pháp sau:

const express = require("express");

* Vì thế, NPM là một công cụ mạnh mẽ có thể giúp chúng ta tiết kiệm thời gian và công sức khi phát triển các ứng dụng Node.js. Bằng cách sử dụng NPM, chúng ta có thể dễ dàng cài đặt và quản lý các mô-đun, giúp tập trung vào việc phát triển ứng dụng của mình.

3/ Sơ lược về mongodb

1.1/ Mongodb là gì?

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, phân tán, dựa trên tài liệu. Nó được phát triển bởi 10gen (nay là MongoDB Inc.) và được phát hành lần đầu tiên vào năm 2009. MongoDB là một trong những hệ quản trị cơ sở dữ liệu NoSQL phổ biến nhất hiện nay và được sử dụng bởi các công ty như Facebook, eBay và Uber.

MongoDB sử dụng mô hình tài liệu(Document) để lưu trữ dữ liệu. Tài liệu(Document) là một cấu trúc giống như JSON có thể chứa các cặp khóa-giá trị. MongoDB không sử dụng bảng và ràng buộc như các hệ quản trị cơ sở dữ liệu quan hệ. Thay vào đó, nó sử dụng các tập hợp(Collection) tài liệu để lưu trữ dữ liệu. Các tập hợp tài liệu là một tập hợp các tài liệu có cùng định dạng.

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Linh hoạt: MongoDB có thể lưu trữ bất kỳ loại dữ liệu nào, không bị giới hạn bởi các ràng buộc.  Tốc độ: MongoDB rất nhanh trong việc truy vấn và truy cập dữ liệu.  Phân tán: MongoDB có thể được phân tán trên nhiều máy chủ, giúp nó có thể mở rộng theo nhu cầu.  Dễ sử dụng: MongoDB rất dễ học và sử dụng, ngay cả đối với những người không quen thuộc với cơ sở dữ liệu. | Không có ràng buộc: MongoDB không có ràng buộc, có thể dẫn đến các vấn đề về tính toàn vẹn dữ liệu.  Không có tính khả dụng cao: MongoDB không có tính khả dụng cao theo mặc định, cần phải thiết lập thủ công.  Không có phân tích: MongoDB không được thiết kế cho phân tích, cần phải sử dụng một công cụ khác để phân tích dữ liệu. |

Bảng 3 Ưu và nhược điểm của mongo db

* Tóm lại, MongoDB là một cơ sở dữ liệu mạnh mẽ và linh hoạt có thể được sử dụng để lưu trữ và truy cập dữ liệu cho nhiều loại ứng dụng. Tuy nhiên, người dùng cần cân nhắc các ưu và nhược điểm của MongoDB trước khi quyết định sử dụng nó cho ứng dụng của mình.

2.2/ Một số kiểu dữ liệu trong mongodb

Chuỗi: Đây là kiểu dữ liệu được sử dụng phổ biến nhất để lưu giữ dữ liệu. Chuỗi trong MongoDB phải là UTF-8 hợp lệ.

Số nguyên: Kiểu dữ liệu này được sử dụng để lưu một giá trị số. Số nguyên có thể là 32 bit hoặc 64 bit phụ thuộc vào Server của bạn.

Boolean: Kiểu dữ liệu này được sử dụng để lưu giữ một giá trị Boolean (true/false).

Double: Kiểu dữ liệu này được sử dụng để lưu các giá trị số thực dấu chấm động.

Mảng: Kiểu dữ liệu này được sử dụng để lưu giữ các mảng hoặc danh sách hoặc nhiều giá trị vào trong một key.

Timestamp: Đánh dấu thời điểm một Document được sửa đổi hoặc được thêm vào.

Object: Kiểu dữ liệu này được sử dụng cho các Document được nhúng vào.

Null: Kiểu dữ liệu này được sử dụng để lưu một giá trị Null.

Symbol: Kiểu dữ liệu này được sử dụng giống như một chuỗi

Date : Kiểu dữ liệu này được sử dụng để lưu giữ date và time hiện tại trong định dạng UNIX time.

Object ID: Kiểu dữ liệu này được sử dụng để lưu giữ ID của Document.

Binary data: Kiểu dữ liệu này được sử dụng để lưu giữ dữ liệu nhị phân.

Regular expression: Kiểu dữ liệu này được sử dụng để lưu giữ Regular Expresion.

IV/ ỨNG DỤNG THƯƠNG MẠI ĐIỆN TỬ CỬA HÀNG SÁCH VỚI REACT-NATIVE VÀ NODE JS

1/ Chi tiết chức năng

- Đăng nhập/đăng ký: ở chức năng này ứng dụng phía người dùng sẽ post api lên server để thực hiện xác thực đăng nhập hoặc đăng ký cho ứng dụng.

- Giỏ hàng: ở chức năng này người dùng có thể thêm các sản phẩm cần mua vào giỏ hàng và tạo đơn hàng

- Danh mục sách: Chức năng này hiển thị các thể loại sách theo danh mục để người dùng dễ tìm hiểu.

- Phân loại sách: chức năng này giúp người dùng có thể lọc sản phẩm theo thể loại sách.

- Tìm kiếm sản phẩm: chức năng này sẽ giúp người dùng tìm kiếm những cuốn sách cần tìm.

2/ Bảng cơ sở dữ liệu

\*bảng csdl Book

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Khóa** | **Nullable** | **Unique** | **Mô tả** |
| 1 | idBook | ObjectId | P |  | x | Id của các sản phẩm |
| 2 | name | String |  |  |  | Tên của sách |
| 3 | price | Number |  |  |  | Giá của sách |
| 5 | publishedDate | String |  |  |  | Ngày/năm xuất bản |
| 5 | genres | [String]  Mảng string |  |  |  | Thể loại(một quyển sách có thể có nhiều thể loại) |
| 6 | imgLink | String |  |  |  | Link ảnh của sách |

Bảng 4 csdl Book

\*Bảng Author(tác giả)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Khóa** | **Nullable** | **Unique** | **Mô tả** |
| 1 | IdAuthor | ObjectId | p |  | x | Id của tác giả |
| 2 | name | String |  |  |  | Tên của tác giả |
| 3 | year | Number |  |  |  | Năm sinh của tác giả |
| 4 | imgLink | String |  |  |  | Link ảnh tác giả |

Bảng 5 csdl author

Và quan hệ giữa tác giả và sách là 1-n(một tác giả có thể sáng tác nhiều sách)

\*Bảng user:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Khóa** | **Nullable** | **Unique** | **Mô tả** |
| 1 | UserId | ObjectId | P |  | x | Id của user |
| 2 | username | String |  |  | x | Tên của user |
| 3 | password | String |  |  |  | Mật khẩu của user |
| 4 | email | String |  |  | x | Email của user |
| 5 | Admin(role) | boolean |  |  |  | Loại tài khoản: admin hoặc user |

Bảng 6 Csdl user