

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**CÔNG NGHỆ PHẦN MỀM CHUYÊN SÂU**

**TRANG SƯU TẬP NHỮNG LỜI PHÊ – CHẤM ĐIỂM  
“HAY-ĐỘC-LẠ” CỦA HỌC SINH, SINH VIÊN**

**Giảng viên hướng dẫn:** ThS. Nguyễn Công Hoan

**Lớp:** SE214.I21

**Nhóm:** 1

**Sinh viên thực hiện:** Nguyễn Phúc Thiên Ân – 15520008

Nguyễn Sĩ Đạt – 15520105

Phạm Lê Huy – 15520312

*Thành phố Hồ Chí Minh – Tháng 6, 2018*

## NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# LỜI CẢM ƠN

Công nghệ phần mềm chuyên sâu là một môn học thú vị đối với sinh viên Đại học Công nghệ Thông tin nói chung và sinh viên khoa Công nghệ Phần mềm nói riêng. Trước hết, chúng em xin gửi lời cảm ơn đến thầy Nguyễn Công Hoan. Nhờ sự giảng dạy nhiệt tình của thầy, nhờ những kiến thức và cả những kinh nghiệm cá nhân quý giá do thầy truyền đạt, chúng em đã nắm được qui trình cũng như các yêu cầu trong quá trình phát triển một phần mềm và áp dụng những điều đó vào việc thực hiện đồ án cuối kỳ.

Tuy nhiên trong quá trình tìm hiểu và thực hiện đồ án, chúng em không tránh khỏi những sai sót. Vì vậy, chúng em rất mong nhận được những lời đóng góp chân thành từ thầy để có thể hoàn thành đồ án cuối kỳ một cách tốt nhất.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn.

# MỤC LỤC

<b>PHẦN 1: GIỚI THIỆU</b> .....	6
1. Tóm tắt đồ án: .....	6
2. Mục đích và phạm vi sử dụng:.....	6
3. Công nghệ sử dụng: .....	6
4. Khó khăn gặp phải: .....	7
<b>PHẦN 2: TỔ CHỨC DỰ ÁN</b> .....	8
1. Lịch trình:.....	8
2. Vai trò và trách nhiệm: .....	8
3. Cấu trúc tổ chức: .....	9
4. Các yêu cầu:.....	9
<b>PHẦN 3: QUẢN LÝ TIẾN ĐỘ</b> .....	10
1. Bảng chi tiết các Sprint: .....	10
2. Ước lượng thời gian và chi phí: .....	11
3. Kế hoạch làm việc: .....	12
3. Kế hoạch giám sát:.....	14
<b>PHẦN 4: SẢN PHẨM</b> .....	15
1. Các link cần biết: .....	15
2. Các màn hình: .....	15
a. Màn hình Dashboard:.....	15
b. Màn hình Post: .....	16
c. Màn hình tạo Post mới: .....	17
d. Màn hình My Post: .....	18
e. Màn hình Admin: .....	19
<b>PHẦN 5: DEPLOYMENT</b> .....	20
1. Tổng quan:.....	20
2. Cài đặt chi tiết: .....	20
a. Git:.....	20
b. Node.js & npm: .....	20
c. MongoDB: .....	21
d. Google OAuth: .....	28
e. Google Maps API: .....	37

<b>f. Cloudinary API:</b> .....	40
<b>g. VAPID keys:</b> .....	41
<b>h. Heroku:</b> .....	42
<b>PHẦN 6: PHỤ LỤC</b> .....	51
<b>1. Một số cách tối ưu React Application trên các công cụ tìm kiếm (SEO)</b> .....	51
<b>a. Thêm viewport meta tag:</b> .....	51
<b>b. Luôn có &lt;title&gt; tag ở mỗi trang:</b> .....	51
<b>c. Luôn trả về đúng HTTP status code:</b> .....	51
<b>d. Sử dụng anchor tag với miêu tả đầy đủ:</b> .....	51
<b>e. Sử dụng React với React Router v4:</b> .....	52
<b>2. Tài liệu tham khảo:</b> .....	52

# PHẦN 1: GIỚI THIỆU

## 1. Tóm tắt đề án:

Mục tiêu ban đầu của dự án là một ứng dụng web cho phép người dùng đăng tải hình ảnh về những lời phê hay-độc-lạ mà mọi người nhận được từ những thầy cô giáo “bá đạo” đi kèm theo đó là những câu chuyện thú vị xung quanh lời phê đó.

Dự án sử dụng công nghệ Progressive Web App (PWA) nên có thể tương thích trên nhiều loại thiết bị, quá trình cài đặt cũng trở nên dễ dàng hơn và trải nghiệm người dùng được chú trọng.

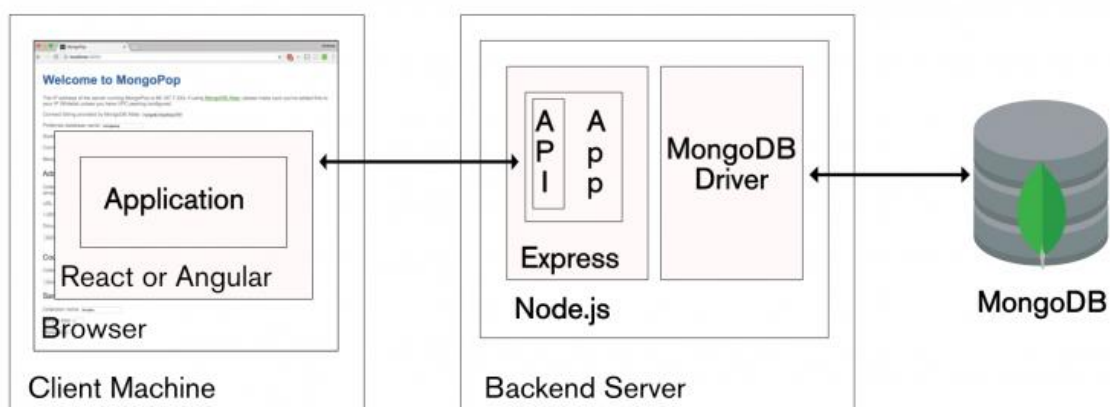
## 2. Mục đích và phạm vi sử dụng:

Sau khi đưa vào sử dụng, ứng dụng sẽ trở thành nơi giao lưu, chia sẻ của các bạn học sinh, sinh viên sau những giờ lên lớp, mang lại sự vui vẻ và thư giãn một cách nhẹ nhàng.

Phần mềm hướng đến đối tượng là các bạn học sinh, sinh viên. Với mục đích là nơi giao lưu, chia sẻ thông tin sau những giờ lên lớp, qua đó mang lại niềm vui, sự giải trí nhẹ nhàng.

## 3. Công nghệ sử dụng:

Phần mềm được phát triển dựa trên MERN stack bao gồm: MongoDB, Express, React, Node.js cùng với một vài tính năng của Progressive Web App.



#### **4. Khó khăn gặp phải:**

- Lần đầu làm việc theo quy trình Agile/Scrum nên nhóm còn nhiều ngỡ ngàng, chưa phát huy được hết các ưu điểm trong việc giám sát, quản lý quy trình của Scrum.
- Progressive Web App là một công nghệ còn khá mới, nên việc nghiên cứu và nắm bắt của nhóm gặp nhiều khó khăn và mất nhiều thời gian.
- Lượng kiến thức mà nhóm tiếp thu được còn hạn chế nên vẫn chưa làm nổi bật hết những tính năng hay của Progressive Web App.

## PHẦN 2: TỔ CHỨC DỰ ÁN

### 1. Lịch trình:

Dự án được chia thành 8 giai đoạn:

Giai đoạn	Deadline
Tìm hiểu các kiến thức cần thiết cho đồ án.	26/03/2018
Hoàn thành Sprint 1	02/04/2018
Hoàn thành Sprint 2	09/04/2018
Hoàn thành Sprint 3	16/04/2018
Hoàn thành Sprint 4	23/04/2018
Hoàn thành Sprint 5	07/05/2018
Hoàn thành Sprint 6	14/05/2018
Hoàn thành Sprint 7	21/05/2018

### 2. Vai trò và trách nhiệm:

Vai trò	Trách nhiệm
Product Owner	Chịu trách nhiệm mọi mặt về sản phẩm và yêu cầu đầu ra của sản phẩm: <ul style="list-style-type: none"><li>- Xây dựng, hiệu chỉnh và duy trì Product Backlog.</li><li>- Đánh giá, xác định thứ tự ưu tiên của các backlog từ đó đưa ra Sprint Backlog.</li><li>- Giám sát thường xuyên quá trình phát triển và xem xét các thay đổi của dự án.</li></ul>
Scrum Master	Chịu trách nhiệm chính để nhóm thực hiện đúng các yêu cầu của Scrum: <ul style="list-style-type: none"><li>- Đảm bảo cho qui trình Scrum được tuân thủ.</li><li>- Tổ chức họp nhóm để trao đổi thông tin, kiến thức cần thiết.</li><li>- Nhắc nhở, đôn đốc để đạt mục tiêu của Sprint hiện tại.</li></ul>



Developer	Chịu trách nhiệm chính về việc phát triển phần mềm: - Hiểu rõ yêu cầu, phân tích, thiết kế, coding sản phẩm. - Đảm bảo về chất lượng và thời gian hoàn thành của công việc được giao. - Sửa lỗi và đóng góp cải tiến sản phẩm. - Hiểu rõ và tuân thủ quy trình Scrum
-----------	--

### 3. Cấu trúc tổ chức:

MSSV	Họ và tên	Vai trò trong dự án
15520008	Nguyễn Phúc Thiên Ân	Developer (chính)
15520105	Nguyễn Sĩ Đạt	Product Owner + Developer (phụ)
15520312	Phạm Lê Huy	Scrum Master + Developer (phụ)

### 4. Các yêu cầu:

- Đảm bảo các thành viên có đầy đủ môi trường phát triển sản phẩm: Laptop, IDE, phương tiện trao đổi thông tin (facebook, email).
- Các thành viên cần đảm bảo có đủ lượng kiến thức để hoàn thành dự án, biết cách áp dụng Scrum vào quản lý quá trình phát triển sản phẩm.

## PHẦN 3: QUẢN LÝ TIẾN ĐỘ

### 1. Bảng chi tiết các Sprint:

Sprint	User story	Point(s)	Thời gian hoàn thành
Sprint 1	Là một người dùng, tôi muốn trang web có giao diện dễ nhìn, dễ tương tác và tương thích với nhiều màn hình thiết bị khác nhau.	3	1 tuần
	Là một người dùng tôi muốn đăng nhập trang web thông qua tài khoản Google	5	
Sprint 2	Là một người dùng tôi muốn ứng dụng cho phép tôi đăng bài viết, xóa bài viết của tôi, đồng thời có thể thả tim (love reaction) và comment vào bài viết khác	8	1 tuần
Sprint 3	Là một người dùng, tôi muốn ứng dụng khởi động nhanh, sử dụng ít lưu lượng internet	2	1 tuần
	Là một người dùng tôi muốn ứng dụng có chức năng Install to Homescreen để có thể sử dụng như một ứng dụng mobile bình thường	2	
Sprint 4	Là một người dùng tôi muốn ứng dụng cho phép tôi truy cập camera của thiết bị đang sử dụng để chụp ảnh, nếu thiết bị không có camera thì tôi có thể chọn ảnh từ thư viện để đăng bài viết mới	3	2 tuần
	Là một người dùng tôi muốn ứng dụng cho phép tôi truy cập địa điểm (location) của thiết bị đang sử dụng, nếu thiết bị không cho phép truy cập địa điểm thì tôi có thể nhập từ bàn phím	5	
Sprint 5	Là một người dùng tôi muốn nhận được thông báo khi có ai đó đăng bài viết mới	5	1 tuần
Sprint 6	Là một người dùng, tôi muốn sử dụng trực tiếp trang web trên internet	5	1 tuần
	Là một admin, tôi muốn quản lý danh sách người dùng và các thông báo	2	
Sprint 7	Là một người dùng, tôi muốn hiển thị giao diện chú trọng hơn vào các bài viết	5	1 tuần

## 2. Ước lượng thời gian và chi phí:

\* Phương pháp: Agile Estimation

\* Công thức tính: **Chi phí = REP/PM/FF**

**Thời gian phát hành = REP/EV (số sprint)**

\* Trong đó:

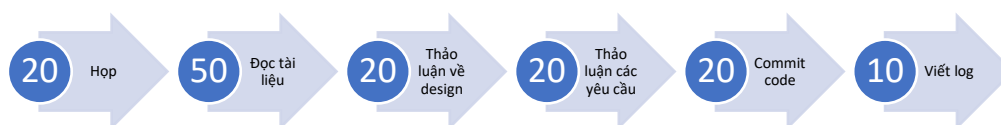
- REP: Release Estimated Points = Số point ước tính của release
- PM: Point – Man = quy đổi 1 point tương ứng man-day
- EV: Estimated Velocity = Tốc độ ước tính
- FF: Focus Factor = Hệ số tập trung

\* Một quy trình ước tính chi phí cơ bản sẽ trải qua các bước sau đây:

*Bước 1: Xác định Focus Factor*

**Focus factor** là tỉ lệ thời gian sản xuất thực tế của nhóm dành cho các story (sau khi trừ đi các thời gian họp hành, học tập, giải lao, ốm đau v.v.). Một nhóm mới thành lập hoặc va phải công nghệ lạ lẫm thì hệ số tập trung càng thấp

Một ngày làm 4 tiếng theo trình tự dưới đây (tính theo phút):



$$FF = 1.0 - (20+50+20+20+20+10)/60*4 = 41.67\%$$

*Bước 2: Xác định Estimated*

Ước lượng khoảng **EV** = 6 (1 Sprint làm 6 point)

*Bước 3: Xác định Point – Man*

- Tính man-day: Nhóm có tổng cộng 3 developer, làm việc 5 ngày/1 tuần, 1 sprint khoảng 2 tuần:  $3*5*2 = 30$  man-day.

- Do 30 man-day/sprint và trung bình 6 point 1 sprint nên

$$PM = 30/6 = 5 \text{ man-day (1 point tương ứng 5 man-day)}$$

*Bước 4: Áp dụng công thức để tính*

$$REP = 45 \text{ (Tổng số points)}$$

$$\text{Thời gian} = REP/EV \text{ (số sprint)} = 45/6 = \sim 7 \text{ sprint} * 2 = 14 \text{ tuần (1 sprint} \sim 2 \text{ tuần)}$$

**Chi phí** = REP/PM/FF = 45/5/0.417 = 22 \* 20 = 440\$ (1 man-day là 20\$/ngày)

### 3. Kế hoạch làm việc:

STT	Task	Người thực hiện	Ngày bắt đầu	Ngày kết thúc	Ghi chú
<b><i>Sprint 1: Giao diện và đăng nhập</i></b>					
1	Dựng giao diện chính cho ứng dụng	Nguyễn Phúc Thiên Ân			Đã hoàn thành
2	Responsive	Nguyễn Phúc Thiên Ân			Đã hoàn thành
3	Đăng nhập thông qua tài khoản Google	Nguyễn Phúc Thiên Ân			Đã hoàn thành
<b><i>Sprint 2: Tính năng liên quan đến bài viết</i></b>					
1	Tạo bài viết mới, xóa bài viết	Nguyễn Phúc Thiên Ân			Đã hoàn thành
2	Thêm love reaction và comment bài viết	Nguyễn Phúc Thiên Ân			Đã hoàn thành
<b><i>Sprint 3: Tối ưu việc khởi động và cài đặt</i></b>					
1	Làm ứng dụng khởi động nhanh và tốn ít lưu lượng	Nguyễn Sĩ Đạt			Đã hoàn thành
2	Tính năng Install To Homescreen	Nguyễn Phúc Thiên Ân			Đã hoàn thành
<b><i>Sprint 4: Camera và Location</i></b>					
1	Truy cập camera của thiết bị hoặc	Nguyễn Phúc Thiên Ân			Đã hoàn thành

	chọn ảnh từ thư viện				
2	Truy cập được location của thiết bị hoặc nhập từ bàn phím	Phạm Lê Huy			Đã hoàn thành
<b><i>Sprint 5: Push Notifications</i></b>					
1	Thông báo người dùng khi có bài viết mới	Nguyễn Sĩ Đạt			Đã hoàn thành
<b><i>Sprint 6: Deploy và quản lý ứng dụng</i></b>					
1	Deploy ứng dụng	Nguyễn Sĩ Đạt			Đã hoàn thành
2	Tạo trang admin quản lí người dùng và thông báo	Nguyễn Phúc Thiên Ân			Đã hoàn thành
<b><i>Sprint 7: Chỉnh sửa giao diện</i></b>					
1	Chỉnh sửa để giao diện chú trọng hơn vào bài viết	Phạm Lê Huy			Đã hoàn thành

### 3. Kế hoạch giám sát:

\* Công cụ sử dụng để quản lý quy trình Scrum: Trello

STT	Hình thức	Nội dung
1	Sprint Planning (Họp Kế hoạch Sprint)	Developer gặp gỡ với Product Owner để lên kế hoạch làm việc cho một Sprint.
2	Daily Scrum (Họp Scrum hằng ngày)	Scrum Master tổ chức cho nhóm họp hàng ngày để chia sẻ tiến độ công việc cũng như các khó khăn gặp phải khi thực hiện sprint
3	Sprint Review (Họp Sơ kết Sprint)	Cuối sprint, developer cùng với Product Owner sẽ rà soát lại các công việc đã hoàn tất trong sprint vừa qua và đề xuất các chỉnh sửa hoặc thay đổi cần thiết cho sản phẩm
4	Sprint Retrospective (Họp Cải tiến Sprint)	Dưới sự trợ giúp của Scrum Master, nhóm sẽ rà soát lại toàn diện sprint vừa kết thúc và tìm cách cải tiến quy trình làm việc cũng như sản phẩm.
5	Họp tổng kết sau khi hoàn thành dự án	Báo cáo các chức năng đã làm được, mức độ hoàn thiện so với mục tiêu ban đầu, đánh giá việc áp dụng quy trình Scrum vào dự án.

## PHẦN 4: SẢN PHẨM

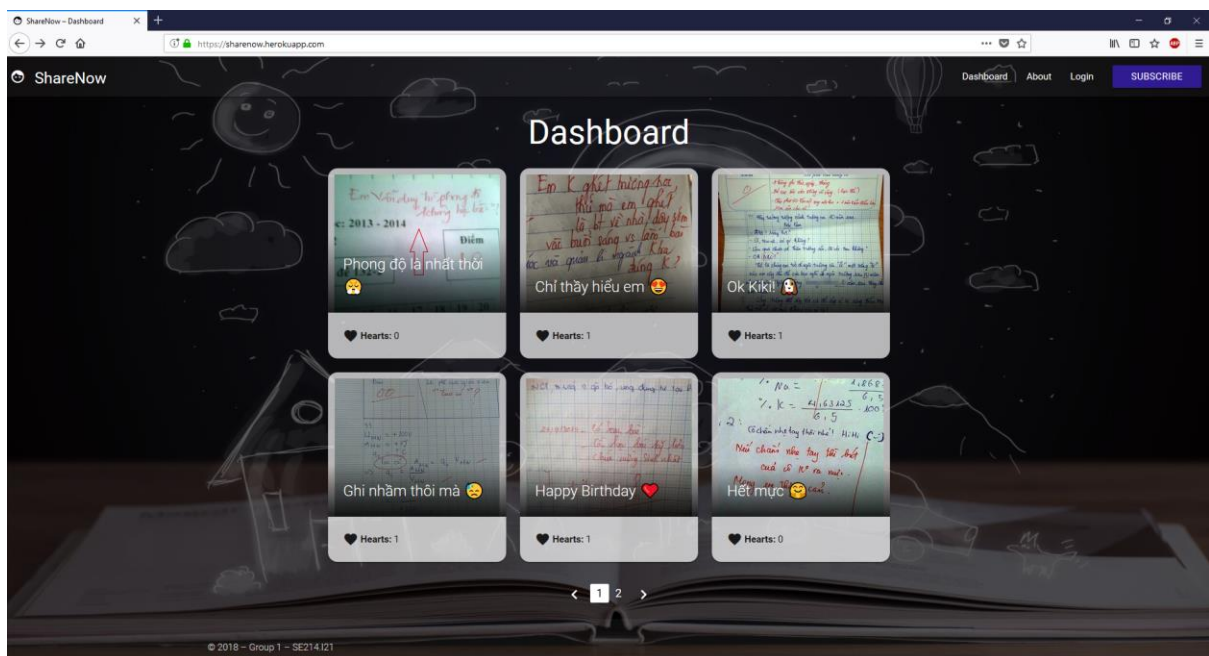
### 1. Các link cần biết:

Link GitHub: <https://github.com/nguyenphucthienan/ShareNow>

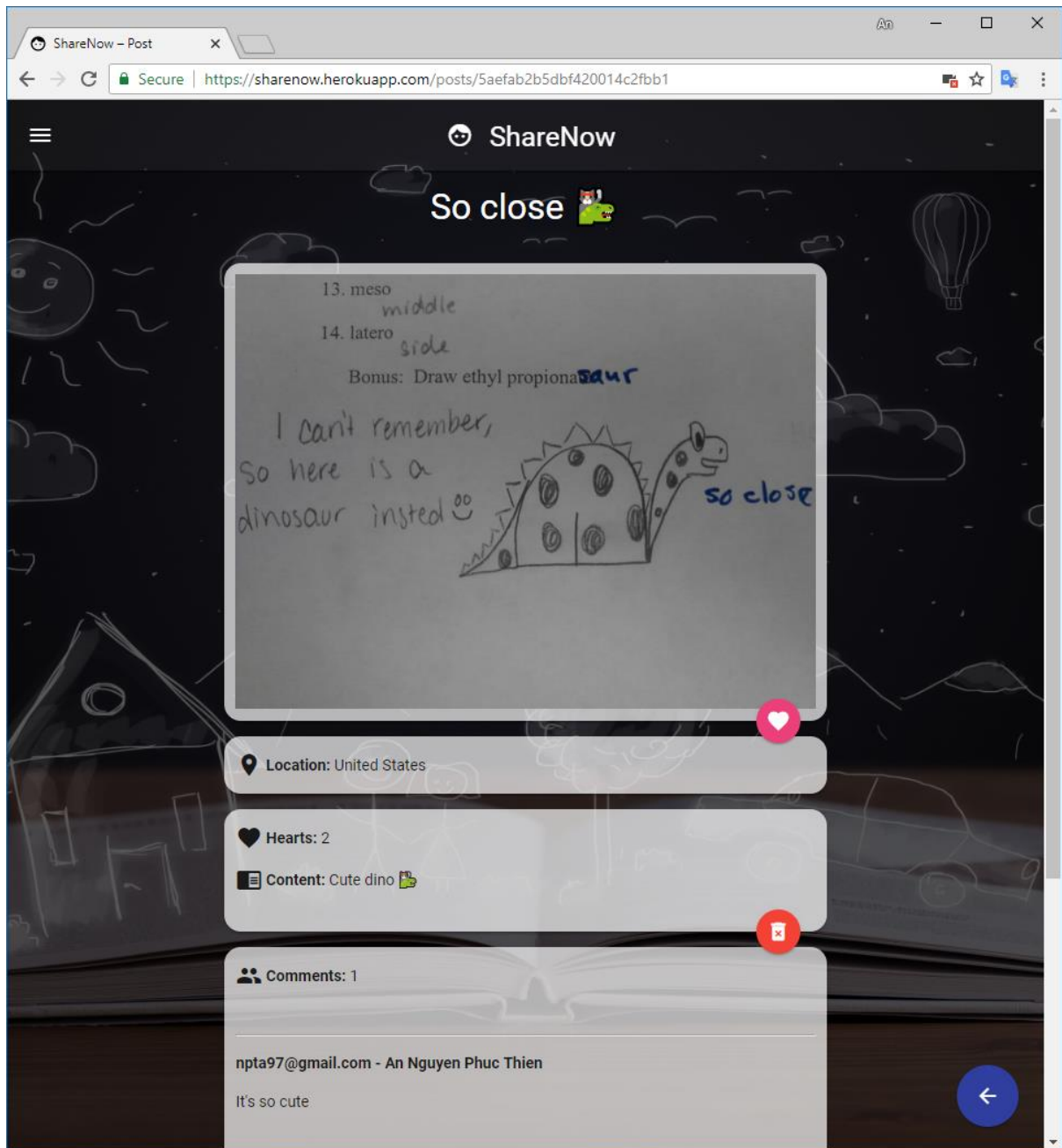
Link sử dụng trực tiếp: <https://sharenow.herokuapp.com/>

### 2. Các màn hình:

#### a. Màn hình Dashboard:



## b. Màn hình Post:





### c. Màn hình tạo Post mới:

ShareNow - New Post

Secure | <https://sharenow.herokuapp.com/posts/new>

ShareNow

# New Post

Choose File No file chosen

Location

Title

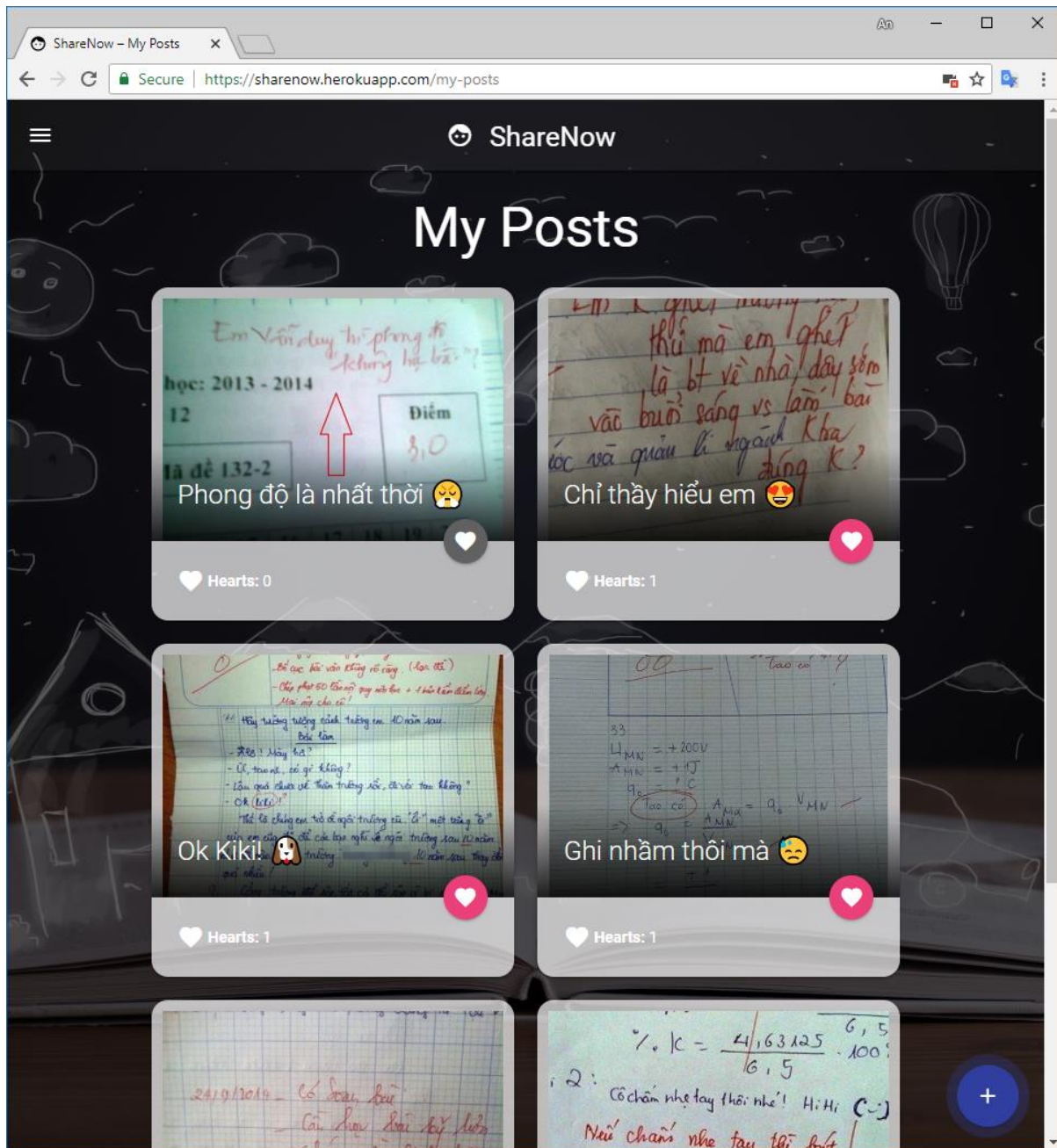
Content

✓ POST

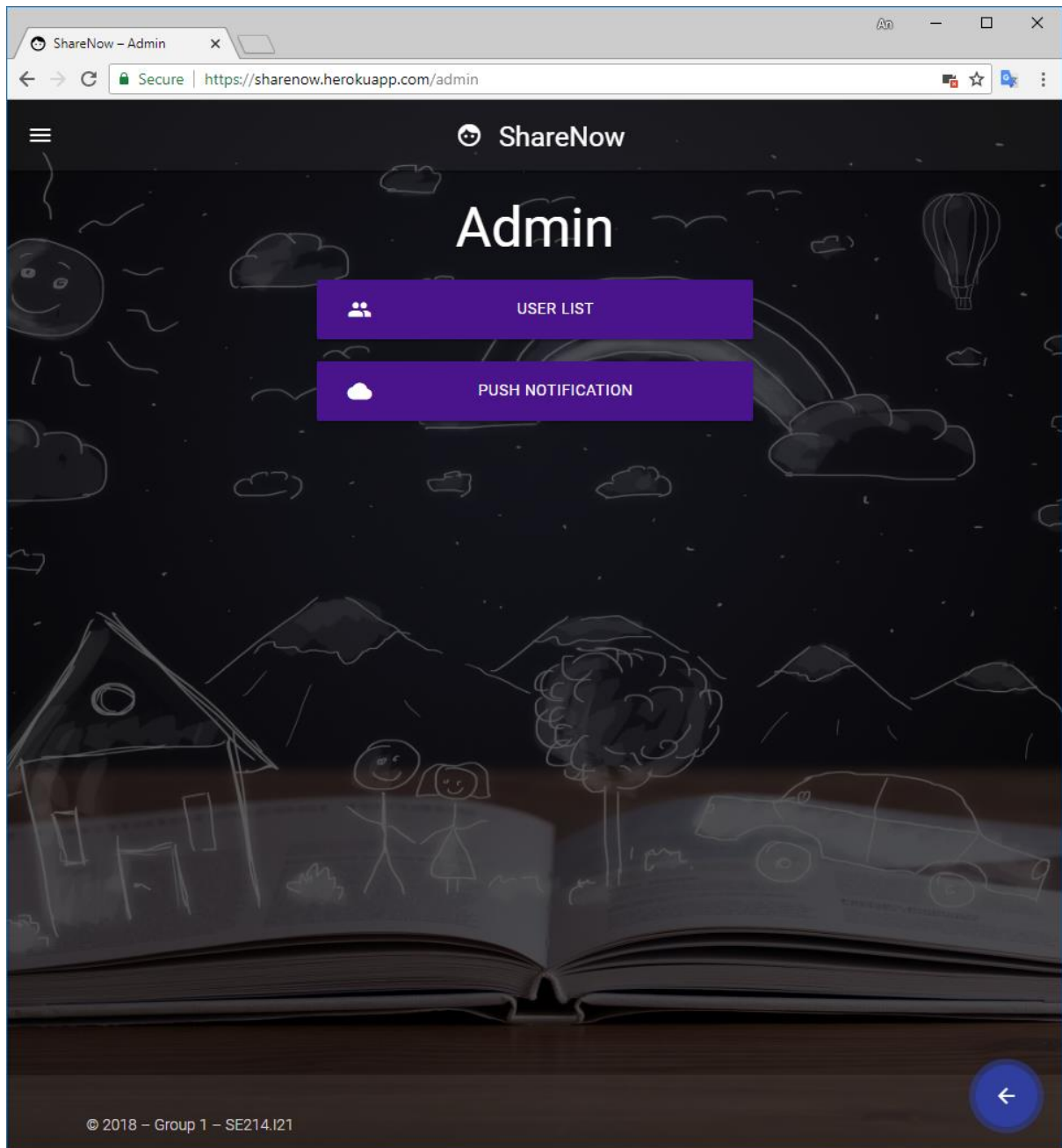
📍

© 2018 - Group 1 - SE214.I21

#### d. Màn hình My Post:



### e. Màn hình Admin:



## PHẦN 5: DEPLOYMENT

### 1. Tổng quan:

Dependency	Version
Git	2.16.2.windows.1
Node.js	8.9.4
npm	6.0.0
Express	4.16.3
MongoDB	3.4.14
Mongoose	5.0.12
React	16.3.0
Redux	3.7.2

### 2. Cài đặt chi tiết:

#### a. Git:

- Download trực tiếp từ trang chủ và cài đặt.

Homepage: <https://git-scm.com/>

#### b. Node.js & npm:

- Download trực tiếp từ trang chủ và cài đặt.

Homepage: <https://nodejs.org/en/>

- Lưu ý: Nên chọn NodeJS version 8.0.0 trở lên để có thể sử dụng các syntax mới nhất của ES6, ES7.

- Node Package Manager (npm) sẽ được cài đặt chung khi cài đặt Node.js.

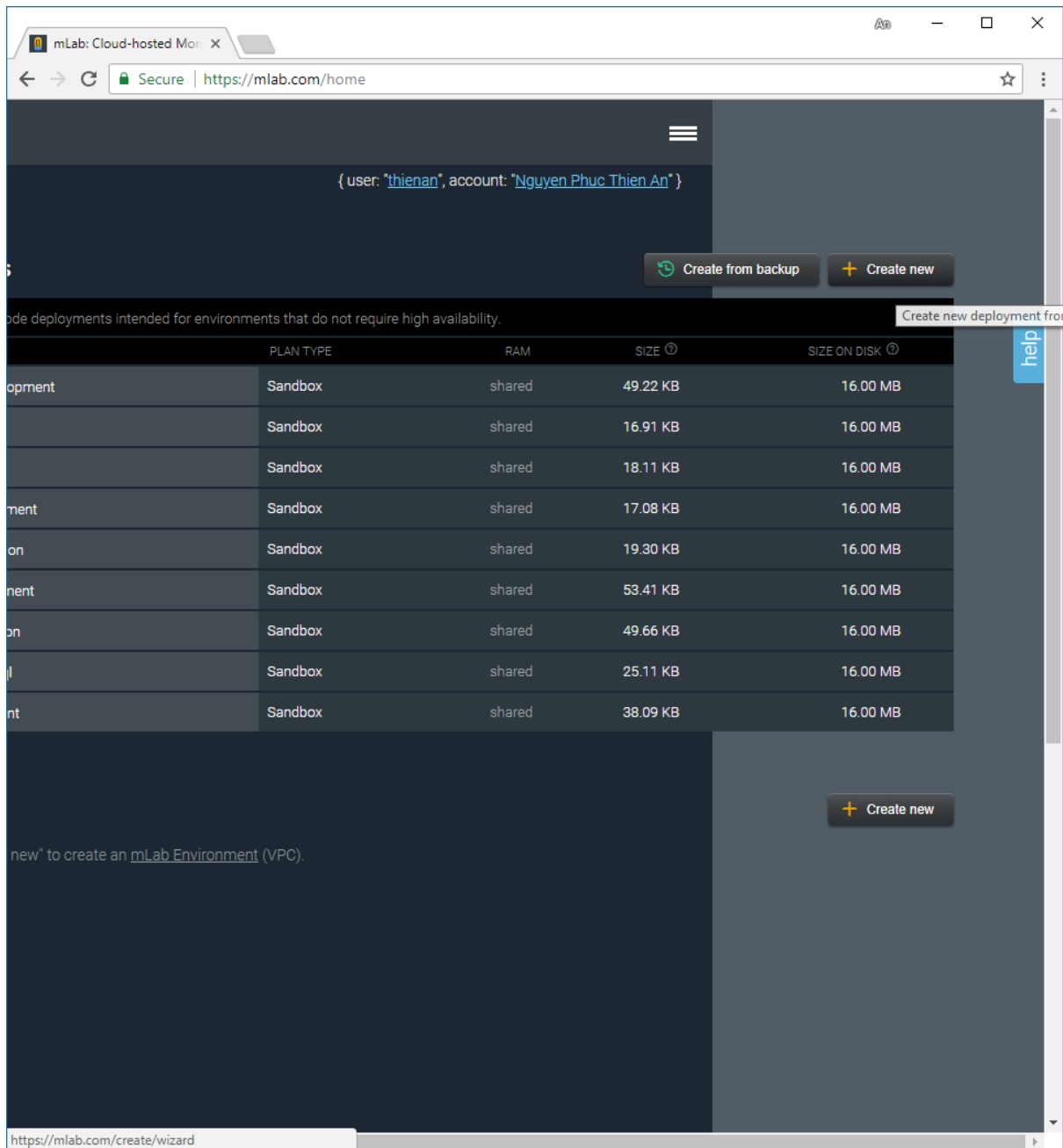
Homepage: <https://www.npmjs.com/>

### c. MongoDB:

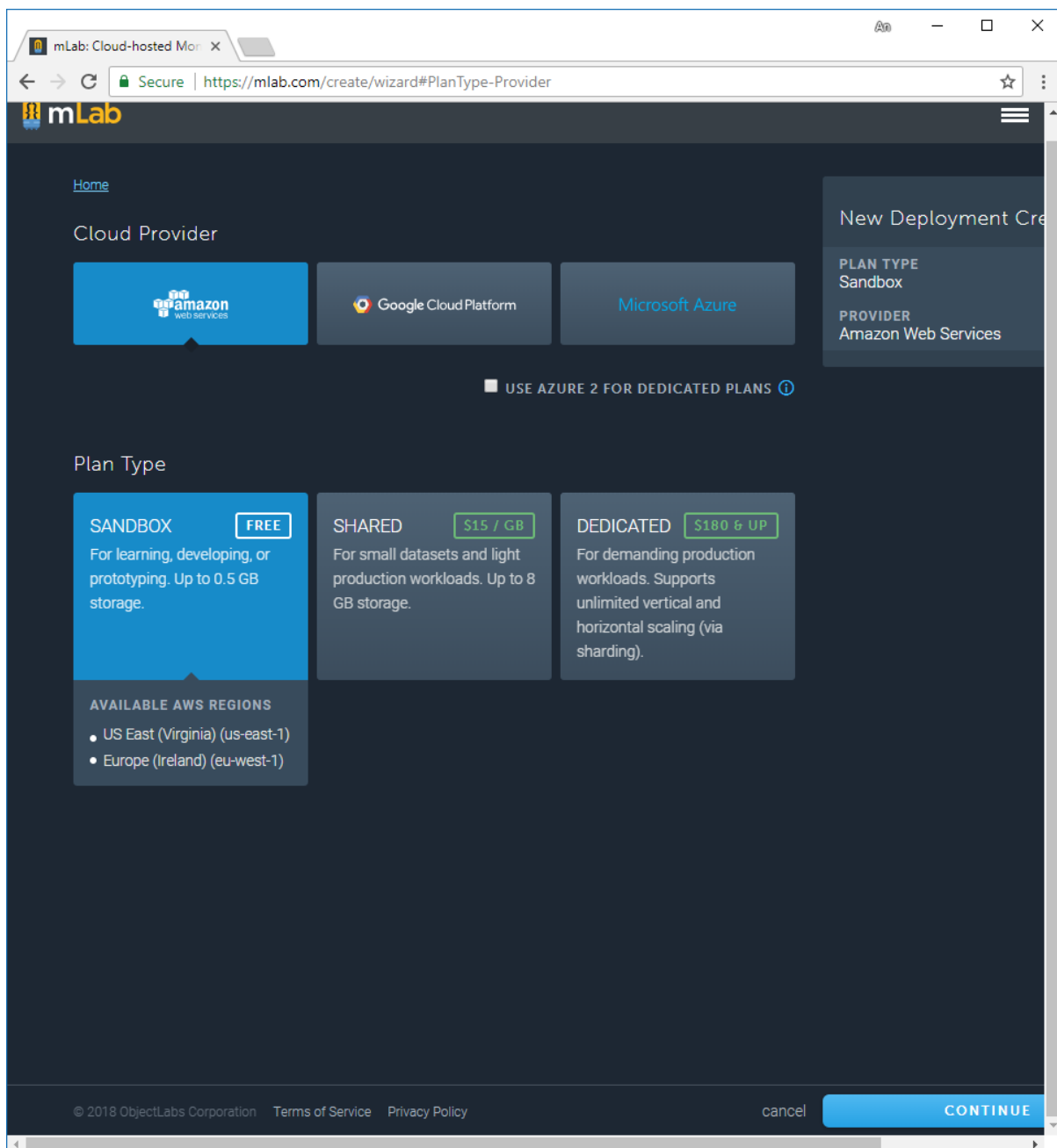
- Sử dụng dịch vụ của mLab.

Homepage: <https://mlab.com/>

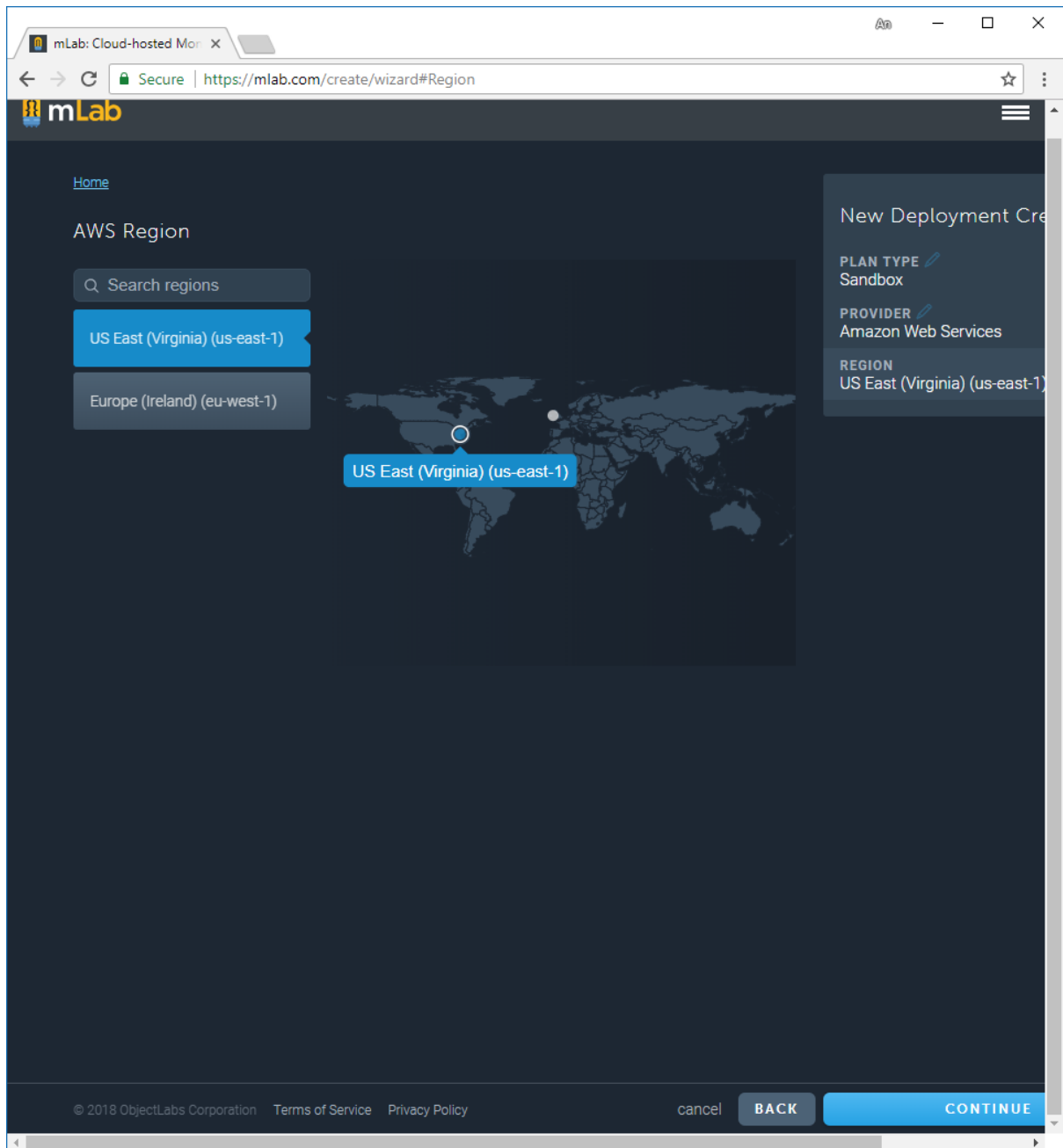
- Sau khi đăng ký tài khoản và đăng nhập, chọn **Create New** để tạo mới database.



- Chọn 1 nhà cung cấp bất kỳ (ở đây là AWS), plan **Sandbox** (Free) và chọn **Continue**.



- Chọn nơi đặt database server và chọn **Continue**.



- Nhập tên database muốn tạo và chọn **Continue**.

mLab: Cloud-hosted MongoDB

Secure | https://mlab.com/create/wizard#Options

mLab

[Home](#)

### Final Details

MONGODB VERSION 3.4.14 (MMAPv1)

DATABASE NAME

**New Deployment Created**

PLAN TYPE  
Sandbox

PROVIDER  
Amazon Web Services

REGION  
US East (Virginia) (us-east-1)

PLAN LINE & SIZE  
- 0.5 GB

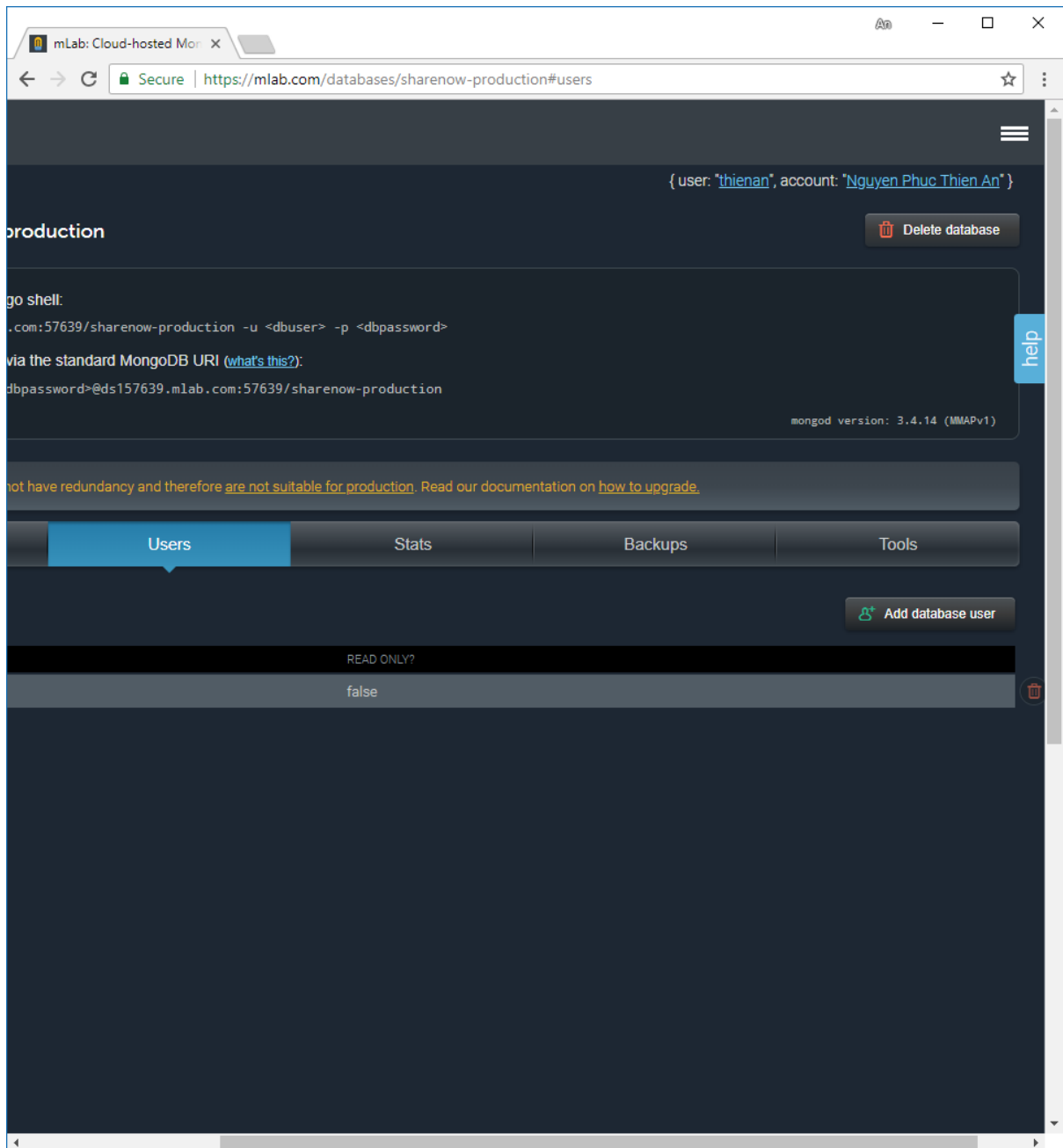
MONGODB VERSION  
3.4.14 (MMAPv1)

DATABASE NAME  
sharenow-production

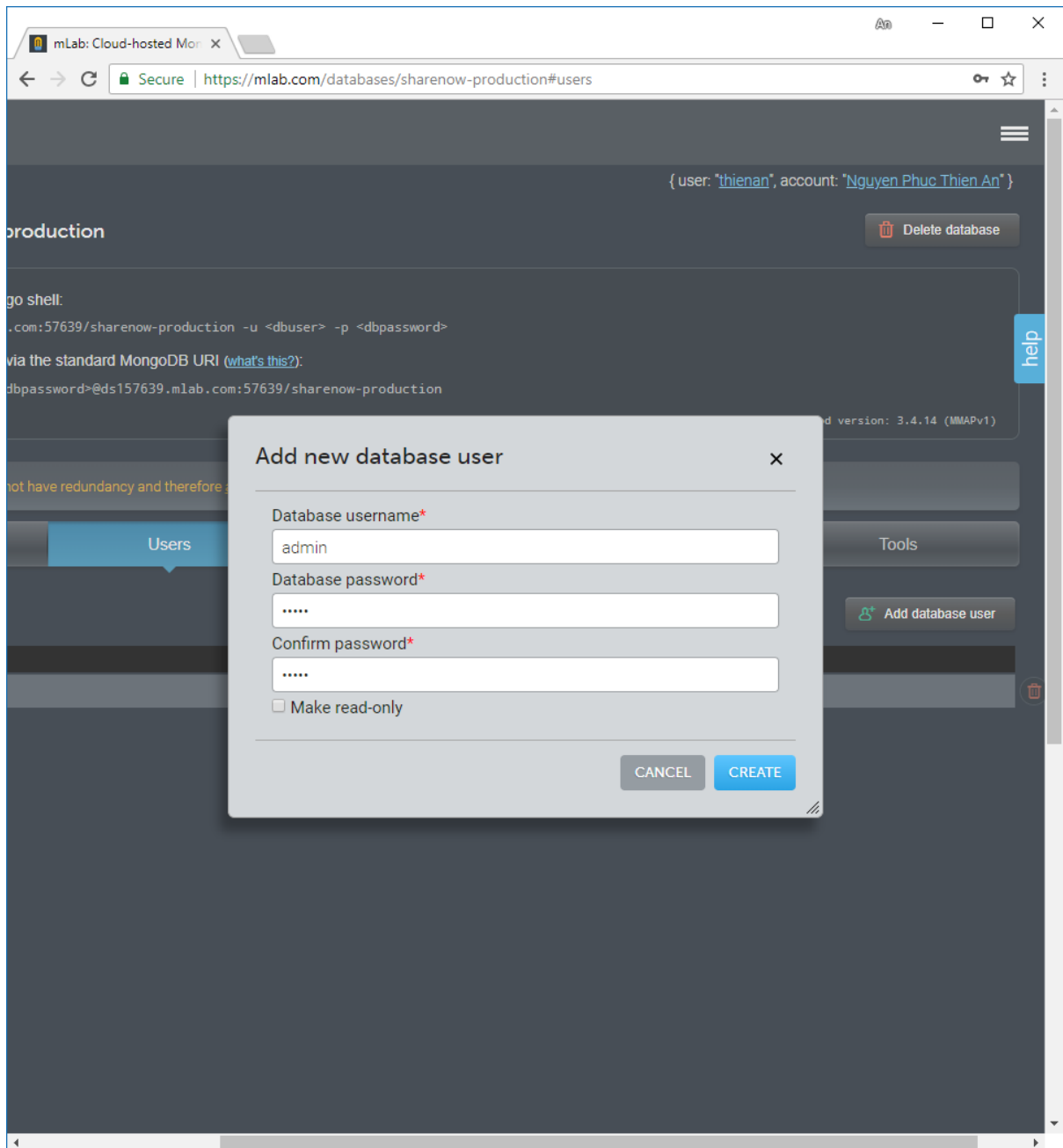
© 2018 ObjectLabs Corporation [Terms of Service](#) [Privacy Policy](#) [cancel](#) **BACK** **CONTINUE**



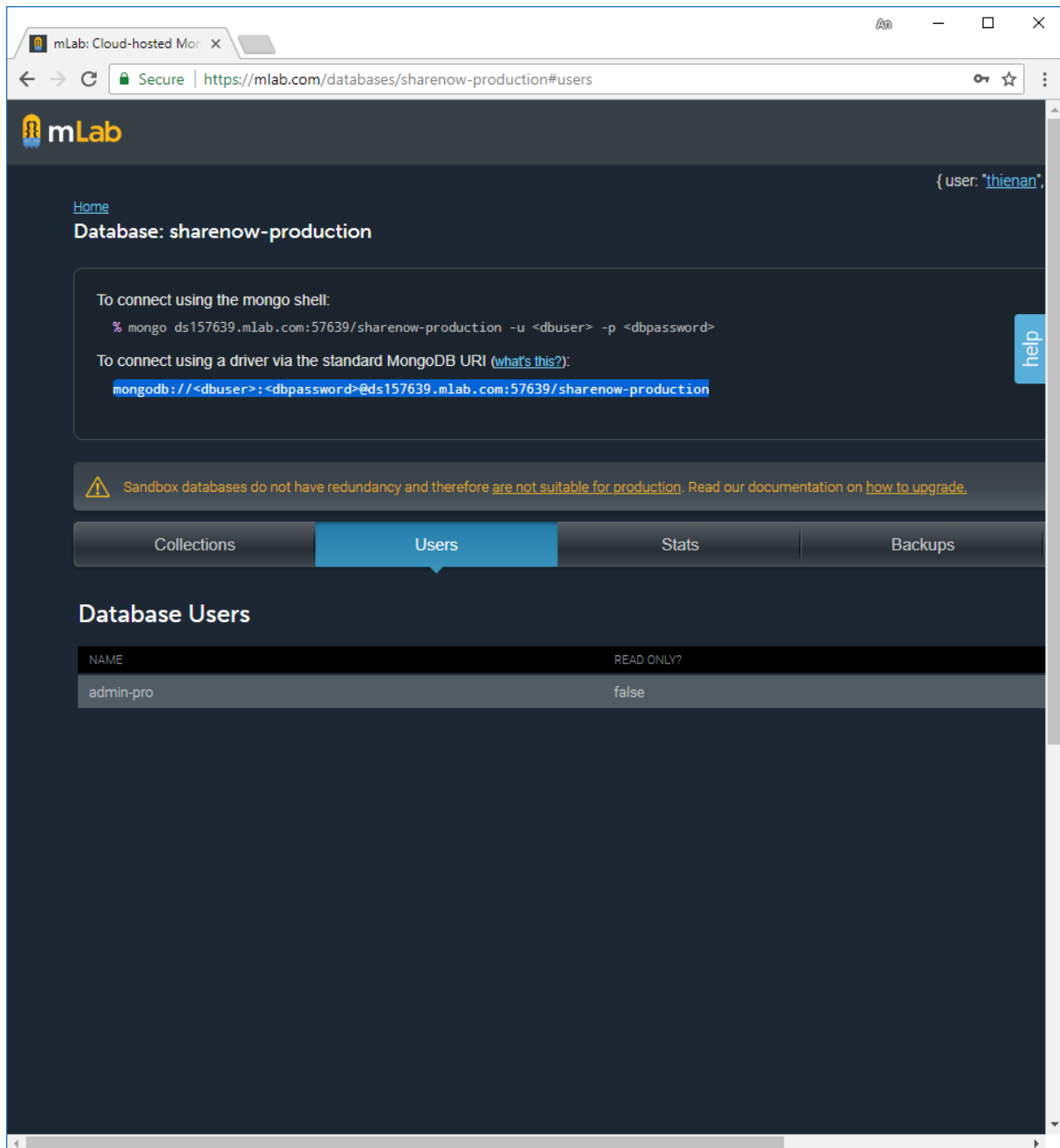
- Sau khi đã tạo thành công database, chọn tab **Users** và chọn **Add database user**.



- Nhập *username*, *password* cho tài khoản admin và chọn **Create**.



- Ta có **MongoDB URI** dùng để kết nối tới database server.
- Thay <dbuser> và <dbpassword> bằng *username* và *password* của tài khoản admin.
- Ví dụ tài khoản admin có *username* là admin, *password* là admin thì **MongoDB URI** sẽ là: `mongodb://admin:admin@ds157639.mlab.com:57639/sharenow-production`



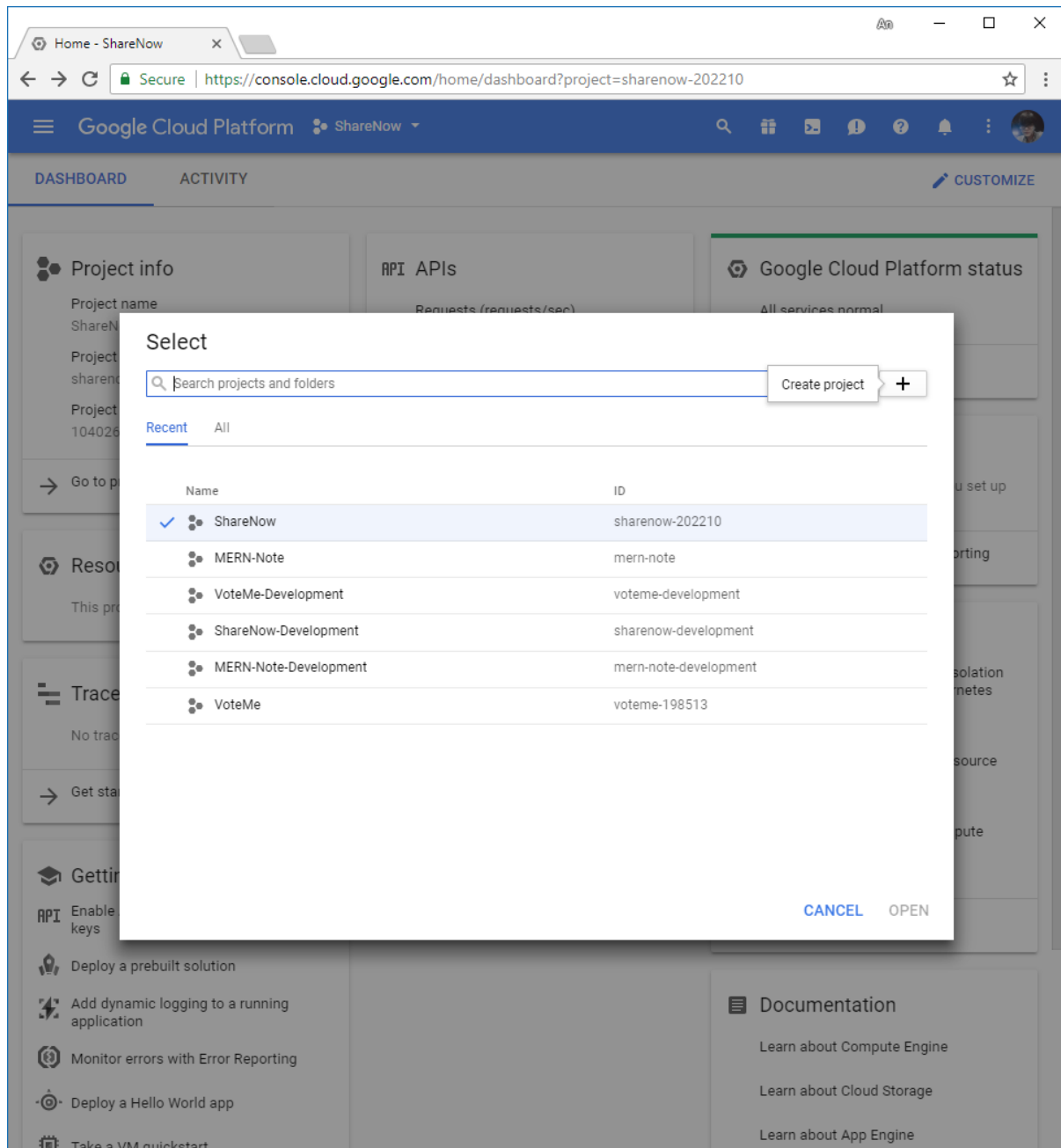
- Lưu lại **MongoDB URI** để sử dụng ở phần sau.

#### d. Google OAuth:

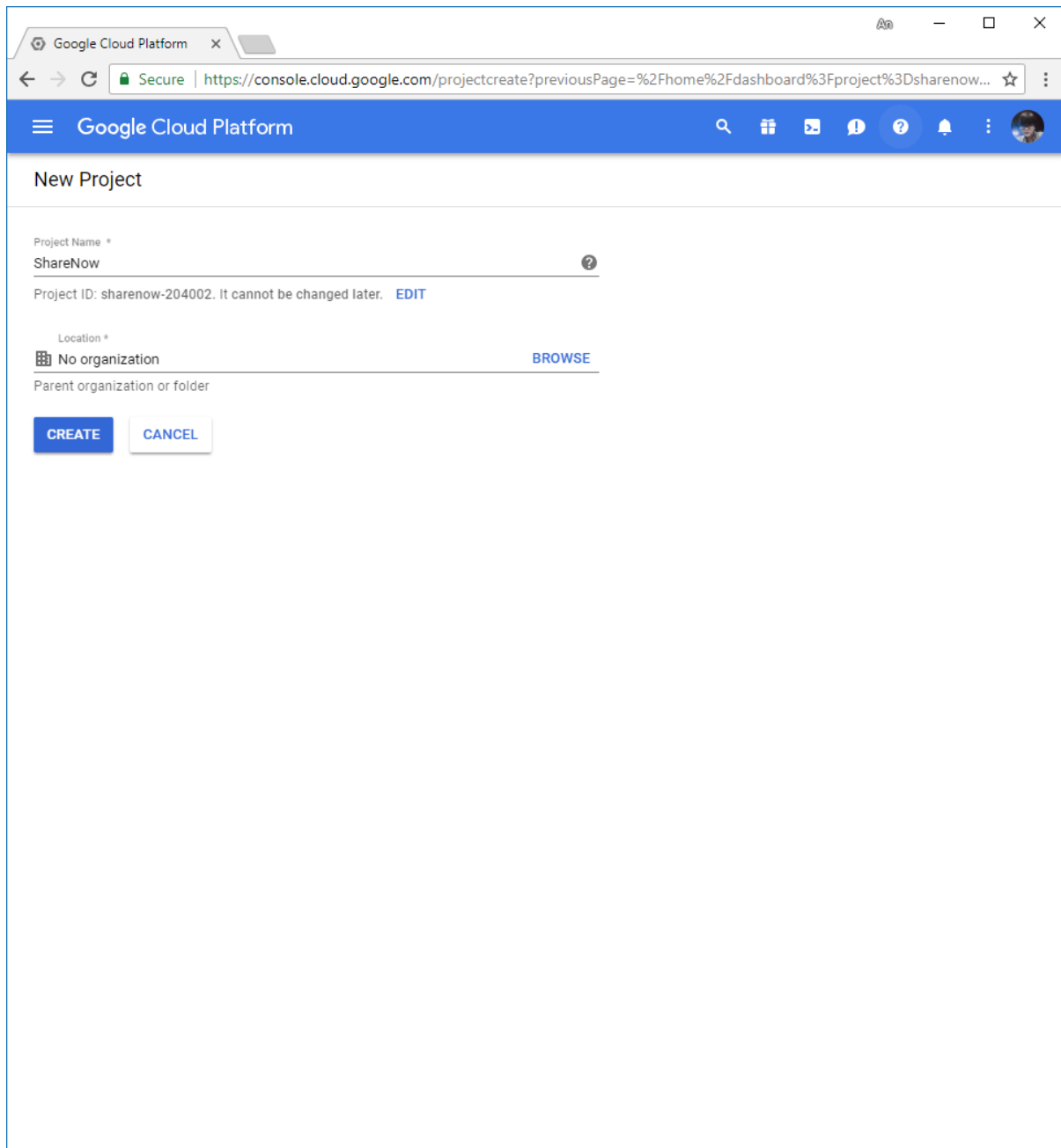
- Đăng nhập **Google Cloud** bằng tài khoản Google.

Google Cloud: <https://console.cloud.google.com/>

- Chọn **Create project** để tạo project mới.



- Nhập tên project vào phần **Project Name** và chọn **Create**.



Google Cloud Platform

Secure | <https://console.cloud.google.com/projectcreate?previousPage=%2Fhome%2Fdashboard%3Fproject%3Dsharenow...>

### New Project

Project Name \*  
ShareNow ?

Project ID: sharenow-204002. It cannot be changed later. [EDIT](#)

Location \*  
No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

- Chờ một lúc để project được khởi tạo. Sau khi project đã được khởi tạo, dưới góc trái phần **Getting Started**, chọn **Enable APIs and get credentials like keys**.

The screenshot shows the Google Cloud Platform (GCP) console dashboard for a new project named 'ShareNow'. The browser address bar shows the URL: <https://console.cloud.google.com/home/dashboard?organizationId=0&project=sharenow-202210>. The dashboard is divided into several sections:

- Left Sidebar:** Contains navigation links for 'Go to project settings', 'Resources' (stating 'This project has no resources'), 'Trace' (stating 'No trace data from the past 7 days'), and 'Get started with Stackdriver Trace'. Below these is the 'Getting Started' section, which lists several tasks: 'Enable APIs and get credentials like keys' (highlighted), 'Deploy a prebuilt solution', 'Add dynamic logging to a running application', 'Monitor errors with Error Reporting', 'Deploy a Hello World app', 'Take a VM quickstart', 'Create a Cloud Storage bucket', 'Create a Cloud Function', and 'Install the Cloud SDK'. At the bottom of this section is a link to 'Explore all tutorials'.
- Center:** Features a graph showing 'api/request\_count:consumed\_api:REDUCE\_SUM(sharenow-202210)' with a value of 0.033. Below the graph is a link to 'Go to APIs overview'.
- Right Sidebar:** Includes a message 'No sign of any errors. Have you set up Error Reporting?' with a link to 'Learn how to set up Error Reporting'. Below this is a 'News' section with articles about container security, Kubernetes best practices, and Jenkins on Google Compute Engine. At the bottom is a 'Documentation' section with links to learn about Compute Engine, Cloud Storage, and App Engine.

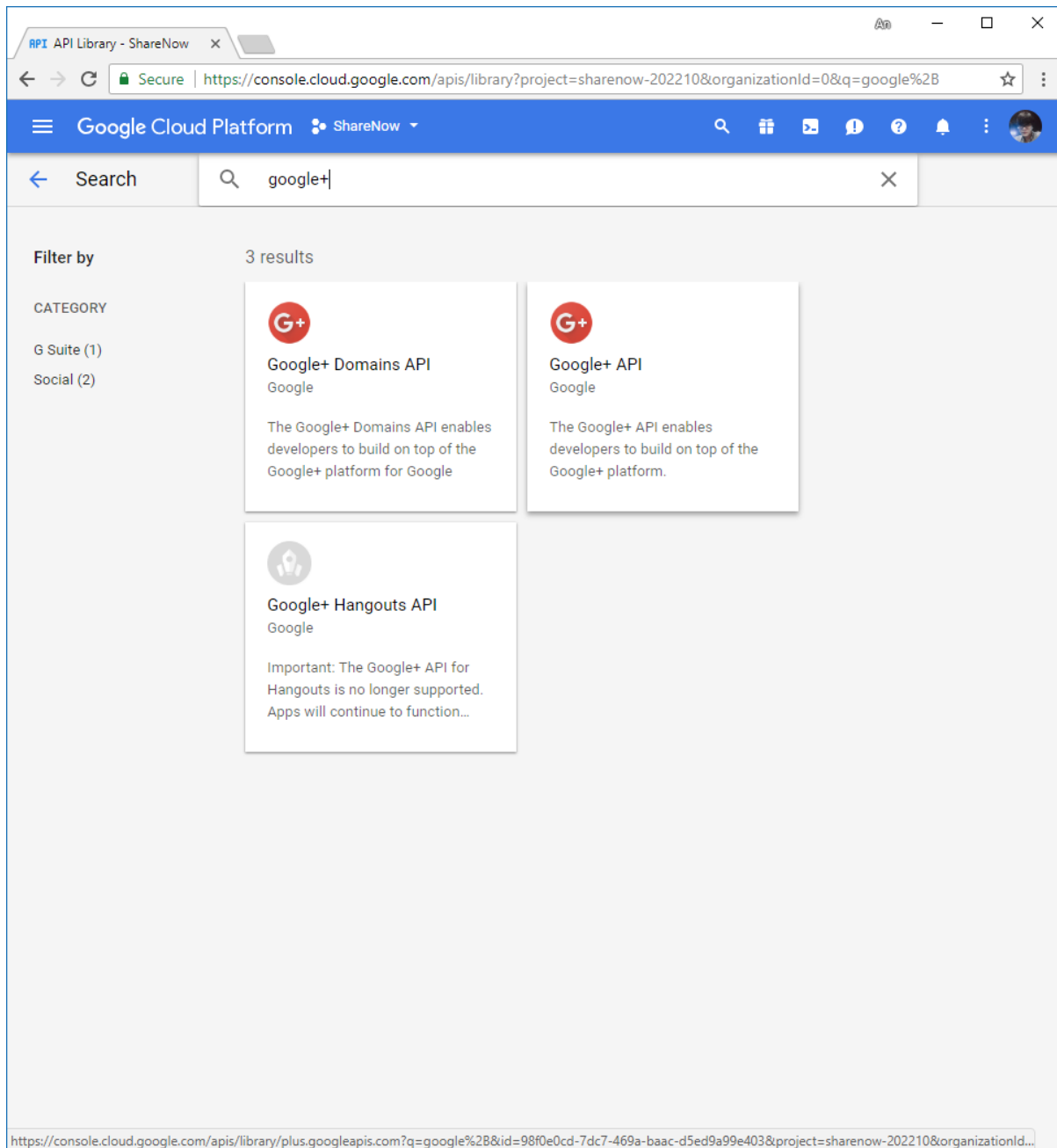
The bottom of the screenshot shows a partial URL: <https://console.cloud.google.com/getting-started?projec...>

- Chọn **ENABLE APIS AND SERVICES**.

The screenshot displays the Google Cloud Platform API Dashboard for the project 'ShareNow'. The interface includes a left sidebar with navigation options: Dashboard, Library, and Credentials. The main content area is titled 'Dashboard' and features a '+ ENABLE APIS AND SERVICES' button. Below this, the 'Enabled APIs and services' section indicates that some APIs are enabled automatically. It provides a time range selector (1 hour, 6h, 12h, 1 day, 2d, 4d, 7d, 14d, 30d) and three monitoring charts: Traffic (Requests/sec), Errors (Percent of requests), and Median latency (Milliseconds). The Traffic chart shows a single request at May 13, 9:09 AM. The Errors and Median latency charts show no data for the selected time period. At the bottom, a table lists the enabled APIs.

API	Requests	Errors	Error ratio	Latency, median	Latency, 98%	
Google+ API	1	0	0%	—	—	Disable

- Tìm và chọn **Google+ API**.





- Nếu API chưa được enable thì chọn **Enable**.

The screenshot shows the Google Cloud Platform API Library page for the Google+ API. The page is titled "Google+ API" and is part of the "Google" category. It features a "MANAGE" button, a "TRY THIS API" button, and a status indicator showing "API enabled". The page also includes an overview, about Google, tutorials, and terms of service.

**Google+ API**  
Google

The Google+ API enables developers to build on top of the Google+ platform.

[MANAGE](#) [TRY THIS API](#) ✓ API enabled

**Type**  
[APIs & services](#)

**Last updated**  
6/30/17, 4:36 AM

**Category**  
[Social](#)

**Service name**  
plus.googleapis.com

**Overview**  
The Google+ API enables developers to build on top of the Google+ platform.

**About Google**  
Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

**Tutorials and documentation**  
[Learn more](#)

**Terms of service**  
By using this product you agree to the terms and conditions of the following license(s): [Google APIs Terms of Service](#), [Google+ API](#)

**More solutions to explore**

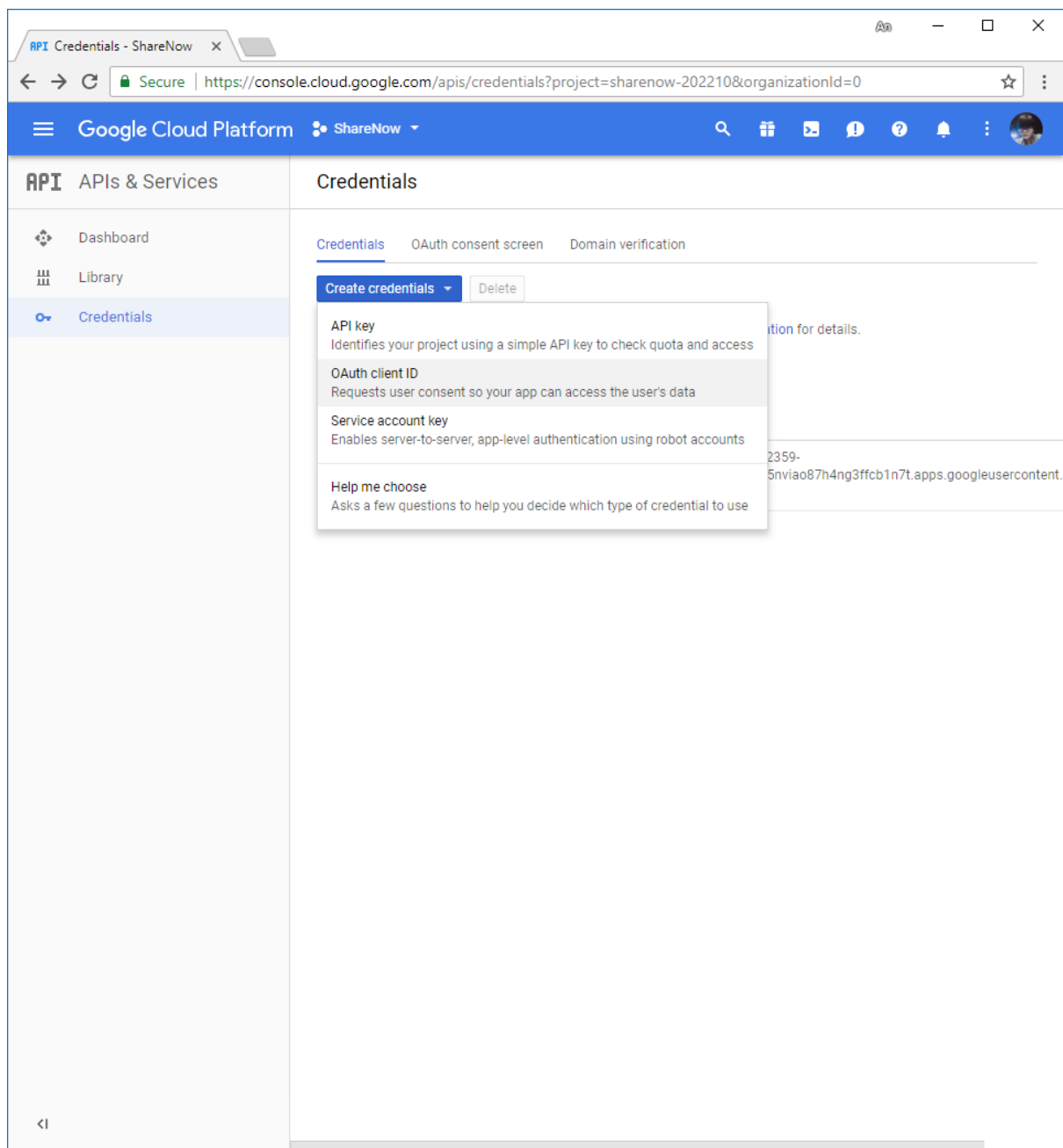
- Gmail API**  
Google
- Google Drive API**  
Google

- Quay về trang **APIs and Services**, chọn phần **Credentials**.

The screenshot shows the Google Cloud Platform console interface. The left sidebar is titled 'APIs & Services' and contains links to 'Dashboard', 'Library', and 'Credentials'. The 'Credentials' link is selected. The main content area is titled 'Credentials' and has tabs for 'Credentials', 'OAuth consent screen', and 'Domain verification'. The 'Credentials' tab is active. Below the tabs, there is a 'Create credentials' button and a 'Delete' button. A message states: 'Create credentials to access your enabled APIs. Refer to the API documentation for details.' Below this, there is a section for 'OAuth 2.0 client IDs' with a table containing one entry.

<input type="checkbox"/>	Name	Creation date	Type	Client ID
<input type="checkbox"/>	Web Client	Apr 25, 2018	Web application	1040267862359-oeqj02js6je5nviao87h4ng3ffcb1n7t.apps.googleusercontent.com

- Chọn **Create credentials** → **OAuth Client ID**.



- Nhập các thông tin cần thiết, đổi “<https://sharenow.herokuapp.com>” thành URL thích hợp và chọn **Create** (hoặc **Save**).

OAuth client - ShareNow X

Secure | <https://console.cloud.google.com/apis/credentials/oauthclient/1040267862359-oeqj02js6je5nviao87h4ng3ffcb1n...>

Google Cloud Platform ShareNow

Client ID for Web application DOWNLOAD JSON RESET SECRET DELETE

Client ID	10402
Client secret	wOSJ
Creation date	Apr 25, 2018, 5:08:31 PM

Name ?

Web Client

Restrictions

Enter JavaScript origins, redirect URIs, or both

**Authorized JavaScript origins**  
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard ([https://\\*.example.com](https://*.example.com)) or a path (<https://example.com/subdir>). If you're using a nonstandard port, you must include it in the origin URI.

<https://sharenow.herokuapp.com> X

<https://www.example.com>

**Authorized redirect URIs**  
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

<http://sharenow.herokuapp.com/api/login/google/callback> X

<https://sharenow.herokuapp.com/api/login/google/callback> X

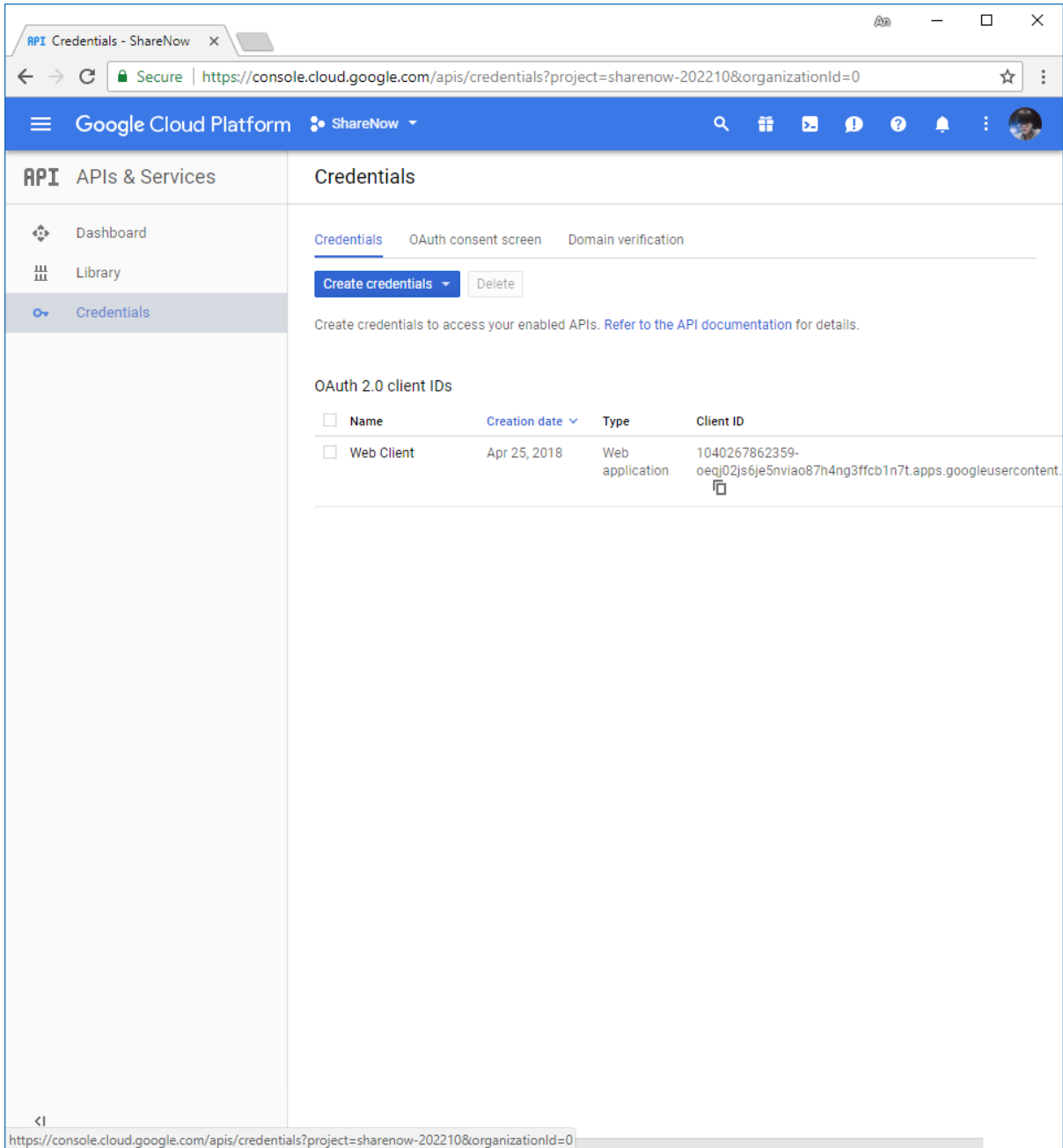
<https://www.example.com/oauth2callback>

Save Cancel

- Lưu lại **Client ID** và **Client secret** để sử dụng ở phần sau.

## e. Google Maps API:

- Vào phần **Credentials** tương tự như trên.

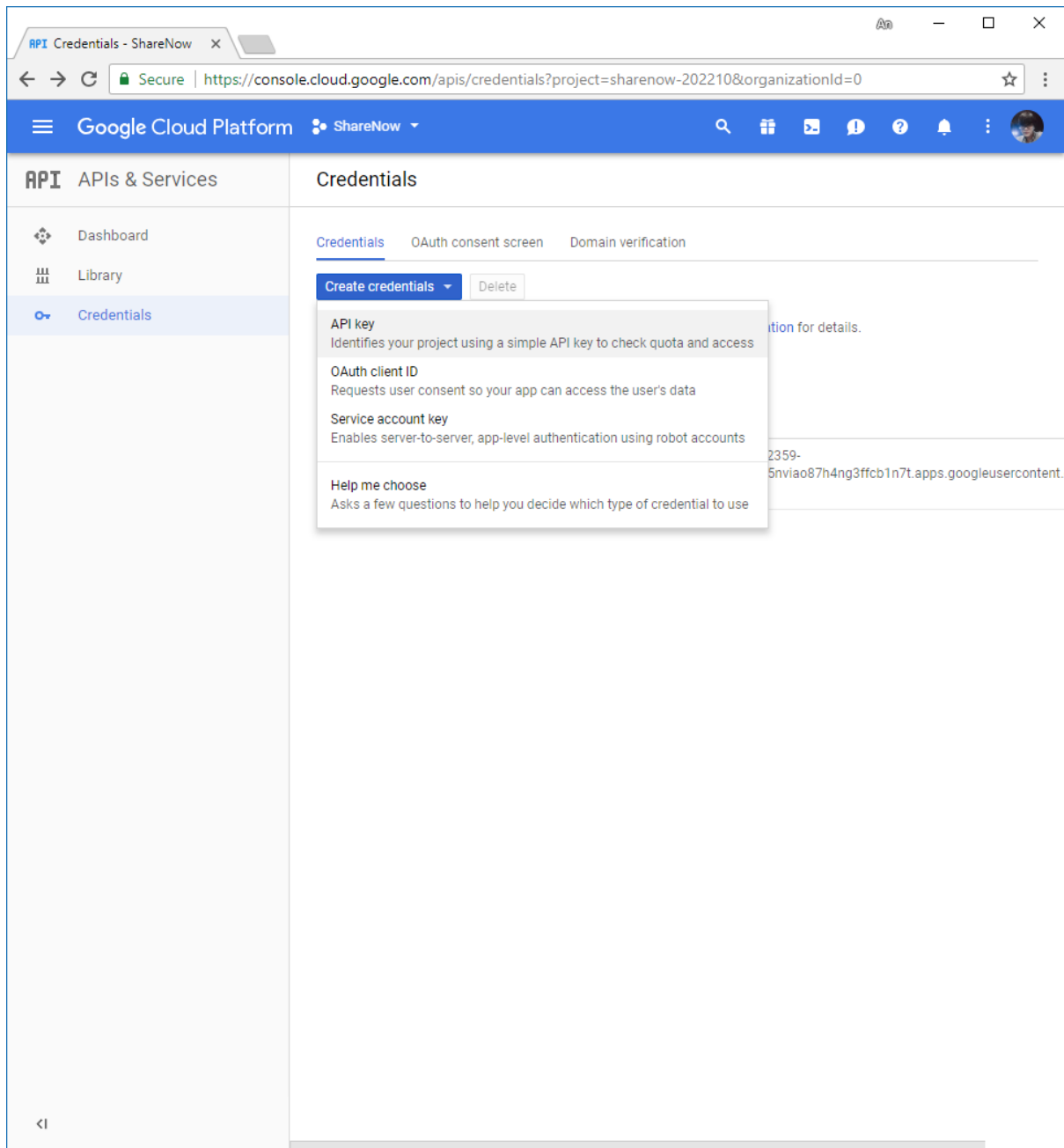


The screenshot shows the Google Cloud Platform console interface. The left sidebar contains the 'APIs & Services' menu with 'Dashboard', 'Library', and 'Credentials' (selected). The main content area is titled 'Credentials' and includes tabs for 'Credentials', 'OAuth consent screen', and 'Domain verification'. Below the tabs are buttons for 'Create credentials' and 'Delete'. A message states: 'Create credentials to access your enabled APIs. Refer to the API documentation for details.' Under the 'OAuth 2.0 client IDs' section, there is a table with one entry:

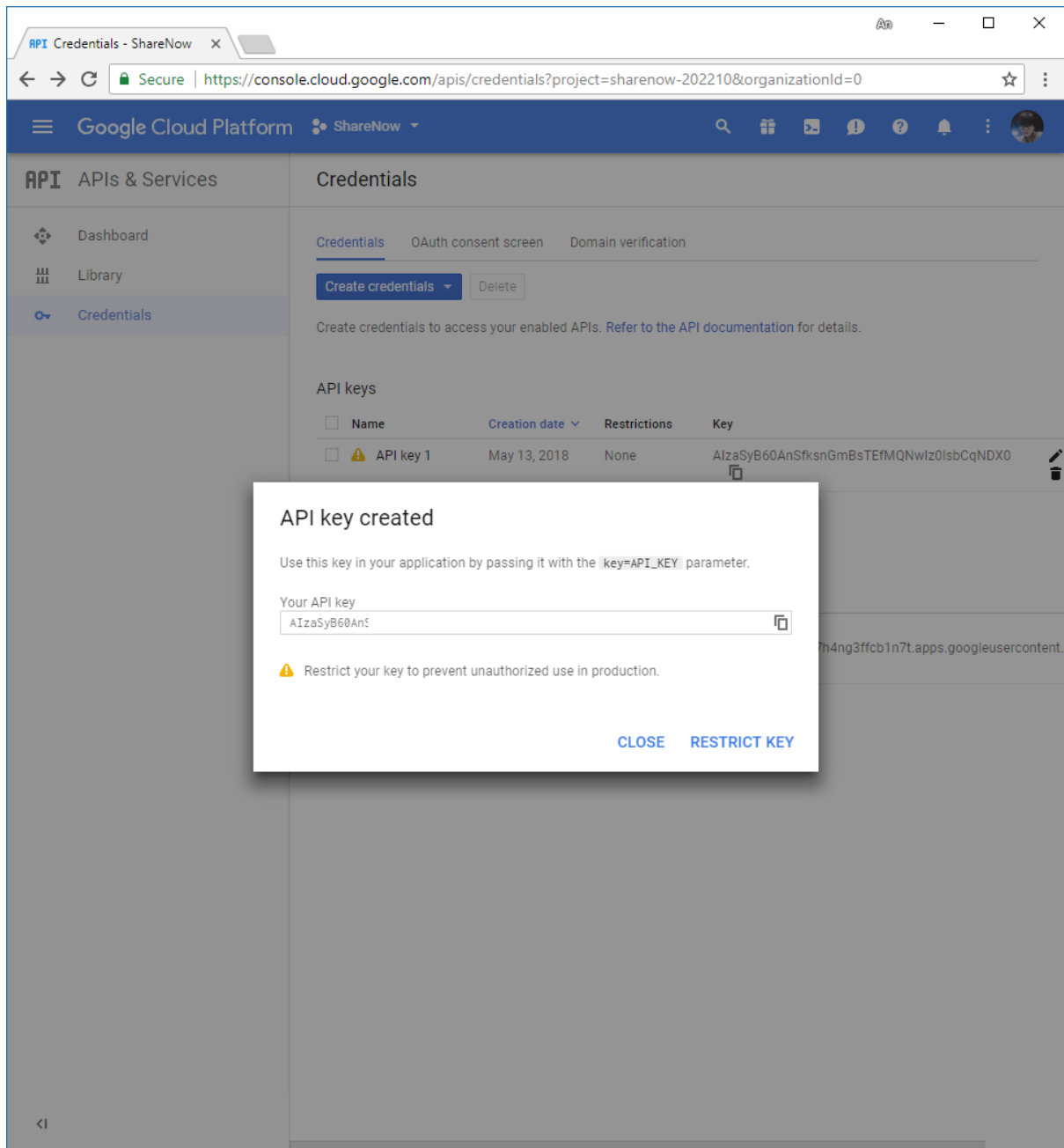
<input type="checkbox"/>	Name	Creation date	Type	Client ID
<input type="checkbox"/>	Web Client	Apr 25, 2018	Web application	1040267862359-oegj02js6je5nviao87h4ng3ffcb1n7t.apps.googleusercontent.com

The browser's address bar shows the URL: <https://console.cloud.google.com/apis/credentials?project=sharenow-202210&organizationId=0>.

- Chọn **Create credentials** → **API Key**.



- Đợi một lúc để **API Key** được tạo.



- Lưu lại **API Key** để sử dụng ở phần sau.

## f. Cloudinary API:

- Dùng **Cloudinary** để upload ảnh, do **Heroku** plan Free không cho phép upload file trực tiếp lên server.

Homepage: <https://cloudinary.com/>

- Sau khi đăng ký và đăng nhập, xuất hiện bên góc trái phần **Account Details**.

The screenshot shows the Cloudinary Management console dashboard. The top navigation bar includes the Cloudinary logo, a 'Welcome' message, and a '1/5' indicator. The main dashboard area is divided into several sections:

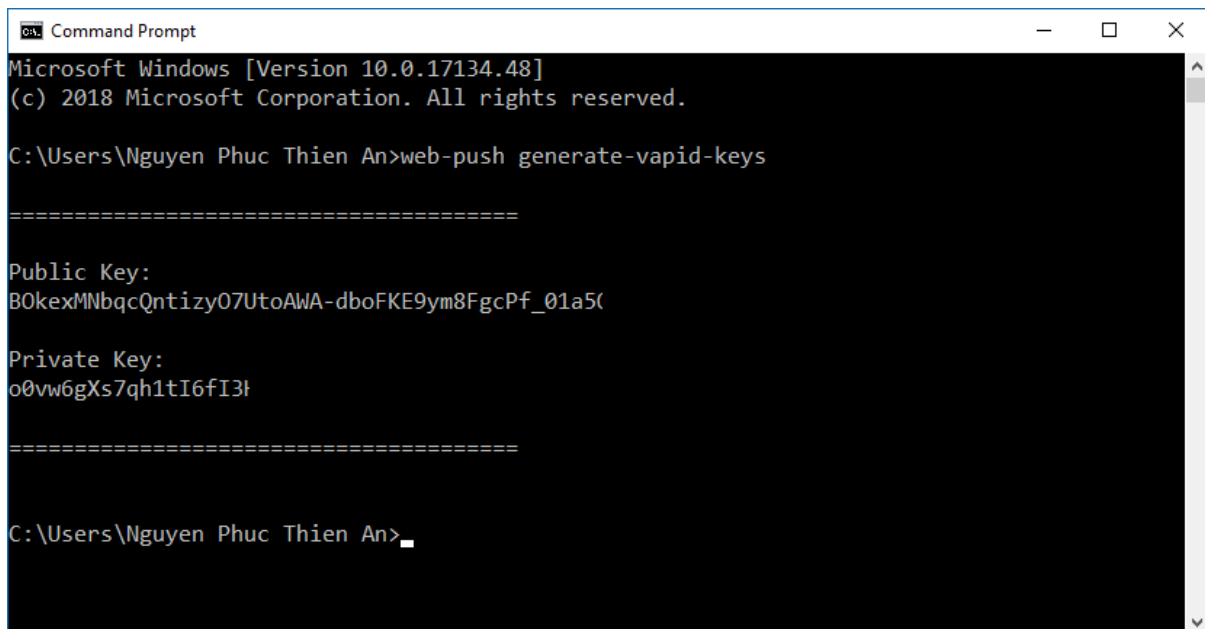
- Dashboard:** Shows the current plan as 'Free Plan' with an 'Upgrade plan' button. It also displays usage statistics: 10 GB storage (0%), 20,000 transformations (0%), and 20 GB bandwidth (0%).
- Account Details:** A section on the left containing account information: Cloud name: 'the', API Key: '924', API Secret: 'va3', and Environment variable: 'CLOUDINARY\_URL=cloudinary://924688771491672:va3kZ-N-\_Y34it'. It also includes a 'Download: YML, PY' link.
- Image & Video Resources:** A section showing 0 resources and 2 transformation types (+2). It also shows 0 derived resources.
- Transformations:** A section showing 2 transformations (0% of plan limit) and a 'Breakdown' link.
- Images & Videos:** A section showing 0 images and videos (0% of plan limit).
- Storage:** A section showing 0 bytes (0% of plan limit) and a 'Breakdown' link.
- Bandwidth:** A section showing 0 bytes (0% of plan limit).
- Recent Blog Posts:** A section on the right listing recent blog posts, including 'Sendify Lightens its Heavy Load of Image Manipulation and Storage with Cloudinary', 'Add the 360 Product Viewer to Your Commerce Site with Cloudinary', 'Introducing Boomerang Video Effect with Cloudinary', and 'HTML5 Video Tags Won't Do for Your Videos'.
- Quick Links:** A section on the right with a link to 'Blog Posts & Tutorials'.

- Lưu lại **Cloud name**, **API Key**, **API Secret** để sử dụng ở phần sau.



#### g. VAPID keys:

- Dùng **VAPID** (Voluntary Application Server Identification for Web Push) để push notification đến người dùng.
- Mở **Command Prompt** (cmd)
- Gõ lệnh ***npm install --global web-push***
- Chờ một lúc để web-push được cài đặt.
- Gõ tiếp lệnh ***web-push generate-vapid-keys*** để tạo VAPID keys.



```
Command Prompt
Microsoft Windows [Version 10.0.17134.48]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Nguyen Phuc Thien An>web-push generate-vapid-keys

=====

Public Key:
B0kexMNBqcQntizy07UtoAWA-dboFKE9ym8FgcPf_01a5C

Private Key:
o0vw6gXs7qh1tI6fI3t

=====

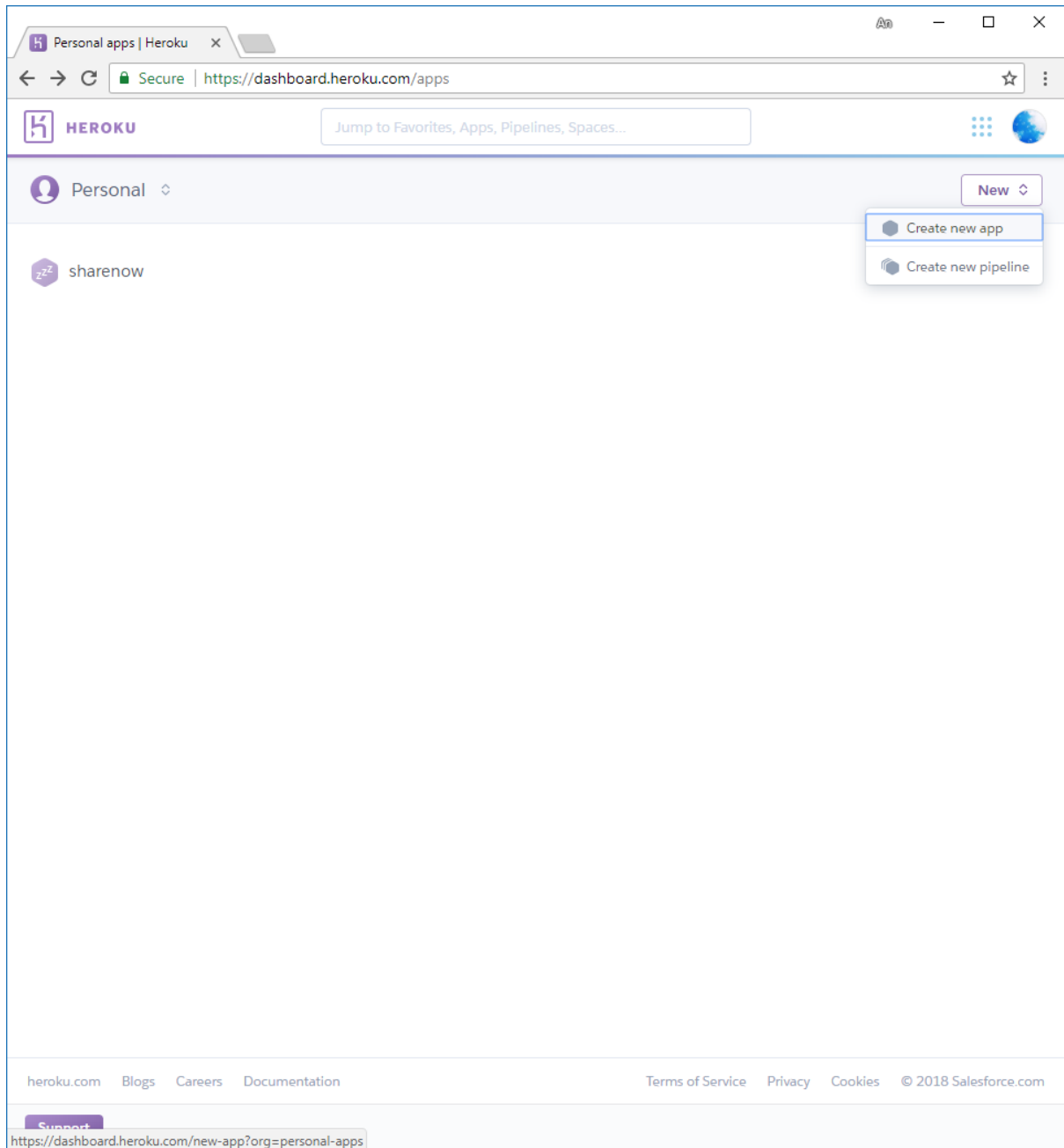
C:\Users\Nguyen Phuc Thien An>
```

- Lưu lại **Public Key** và **Private Key** để sử dụng ở phần sau.

## h. Heroku:

Homepage: <https://heroku.com/>

- Sau khi đăng ký và đăng nhập, chọn **Create new app**.



- Nhập tên App vào phần **App name** và chọn **Create app**.

Create New App | Heroku X

Secure | [https://dashboard.HEROKU.com/new-app](https://dashboard.heroku.com/new-app)

HEROKU Jump to Favorites, Apps, Pipelines, Spaces...

Create New App

App name

sharenow

Choose a region

Europe

Add to pipeline...

Create app

heroku.com Blogs Careers Documentation Terms of Service Privacy Cookies © 2018 Salesforce.com

Support

- Chọn tab **Settings**, chọn **Real Configs Vars**.

The screenshot shows the Heroku dashboard for an application named 'sharenow'. The browser address bar indicates the URL is <https://dashboard.heroku.com/apps/sharenow/settings>. The page has a top navigation bar with the Heroku logo and a search bar. Below this is a sub-navigation bar with tabs: Overview, Resources, Deploy, Metrics, Activity, Access, and Settings (which is currently selected). The main content area is divided into several sections:

- Name:** The application name is 'sharenow', with an 'Edit' link next to it.
- Config Vars:** A section with a 'Reveal Config Vars' button. Below the button, it states: 'Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.'
- Info:** A table-like section showing application details:

Region	United States
Stack	heroku-16
Framework	Node.js
Slug size	65.4 MiB of 500 MiB
Heroku Git URL	<a href="https://git.heroku.com/sharenow.git">https://git.heroku.com/sharenow.git</a>
- Buildpacks:** A section with an 'Add buildpack' button. Below the button, it lists the current buildpack: 'heroku/nodejs' with a red 'X' icon next to it. A link 'Find new buildpacks on Heroku Elements' is also present.

- Nhập vào các **Environment variables**.

The screenshot shows the Heroku dashboard for an application named 'sharenow'. The 'Config Vars' section is active, displaying a list of environment variables. Each variable has a name, a value, and a 'Show' button. To the right of each variable, there are icons for editing (pencil) and deleting (X). A 'Hide Config Vars' button is located at the top right of the list. Below the list, there are input fields for 'KEY' and 'VALUE', and an 'Add' button.

KEY	VALUE	Show	Edit	Delete
APP_NAME	Sh...	Show		
BASE_URL	ht...	Show		
CLOUDINARY_API_KEY	36...	Show		
CLOUDINARY_API_SECRET	ly...	Show		
CLOUDINARY_CLOUD_NAME	br...	Show		
COOKIE_KEY	uN...	Show		
GOOGLE_CLIENT_ID	10...	Show		
GOOGLE_CLIENT_SECRET	w0...	Show		
MONGO_URI	mo...	Show		
NPM_CONFIG_PRODUCTION	fa...	Show		
UPLOAD_IMAGE_URL	ht...	Show		
VAPID_PRIVATE_KEY	6p...	Show		
VAPID_PUBLIC_KEY	BJ...	Show		

KEY VALUE Add

Variable Name	Value
APP_NAME	ShareNow
BASE_URL	Root URL VD: https://sharenow.herokuapp.com
CLOUDINARY_API_KEY	Cloudinary API Key
CLOUDINARY_API_SECRET	Cloudinary API Secret
CLOUDINARY_CLOUD_NAME	Cloudinary Cloud Name
COOKIE_KEY	Tùy ý, nên chọn chuỗi dài, nhiều kí tự, khó đoán
GOOGLE_CLIENT_ID	Google OAuth Client ID
GOOGLE_CLIENT_SECRET	Google OAuth Client Secret
MONGO_URI	MongoDB URI
UPLOAD_IMAGE_URL	{RootURL}+'/image/uploads/' VD: https://sharenow.herokuapp.com/images/uploads
VAPID_PRIVATE_KEY	VAPID Private Key
VAPID_PUBLIC_KEY	VAPID Public Key

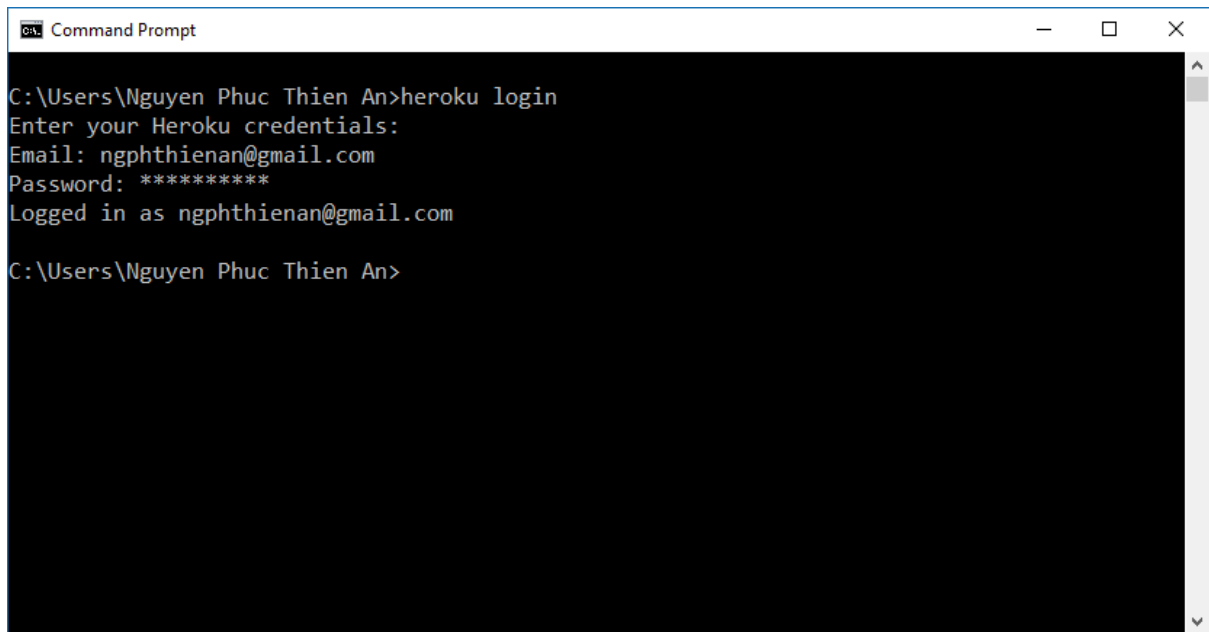
- Download **Heroku CLI** từ trang chủ và cài đặt.

Heroku CLI: <https://devcenter.heroku.com/articles/heroku-cli>

- Mở **Command Prompt** (cmd).

- **cd** tới folder ShareNow

- Gõ lệnh ***heroku login*** và đăng nhập bằng tài khoản đã đăng ký.



```
Command Prompt
C:\Users\Nguyen Phuc Thien An>heroku login
Enter your Heroku credentials:
Email: ngphthienan@gmail.com
Password: *****
Logged in as ngphthienan@gmail.com
C:\Users\Nguyen Phuc Thien An>
```

- Gõ lệnh ***git remote add heroku <Heroku Git URL>***.

VD: ***git remote add heroku https://git.heroku.com/sharenow.git***

- Heroku Git URL có thể xem ở tab **Settings**.

The screenshot shows the Heroku dashboard for an application named 'sharenow'. The browser address bar indicates the URL is <https://dashboard.heroku.com/apps/sharenow/settings>. The Heroku logo is in the top left, and a search bar is in the top right. The navigation bar shows 'Personal' and 'sharenow' with a star icon, 'Open app', and 'More' options. The 'Settings' tab is selected in the navigation menu. The main content area is divided into sections: 'Name' (sharenow), 'Config Vars' (with a 'Reveal Config Vars' button), 'Info' (a table of application details), and 'Buildpacks' (with an 'Add buildpack' button and a list of buildpacks).

Name	
sharenow	<a href="#">Edit</a>

**Config Vars** [Reveal Config Vars](#)

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Info	
Region	United States
Stack	heroku-16
Framework	Node.js
Slug size	65.4 MiB of 500 MiB
Heroku Git URL	<a href="https://git.heroku.com/sharenow.git">https://git.heroku.com/sharenow.git</a>

**Buildpacks** [Add buildpack](#)

Buildpacks are scripts that are run when your app is deployed. They are used to install dependencies for your app and configure your environment. [Find new buildpacks on Heroku Elements](#)

heroku/nodejs	<a href="#">X</a>
---------------	-------------------



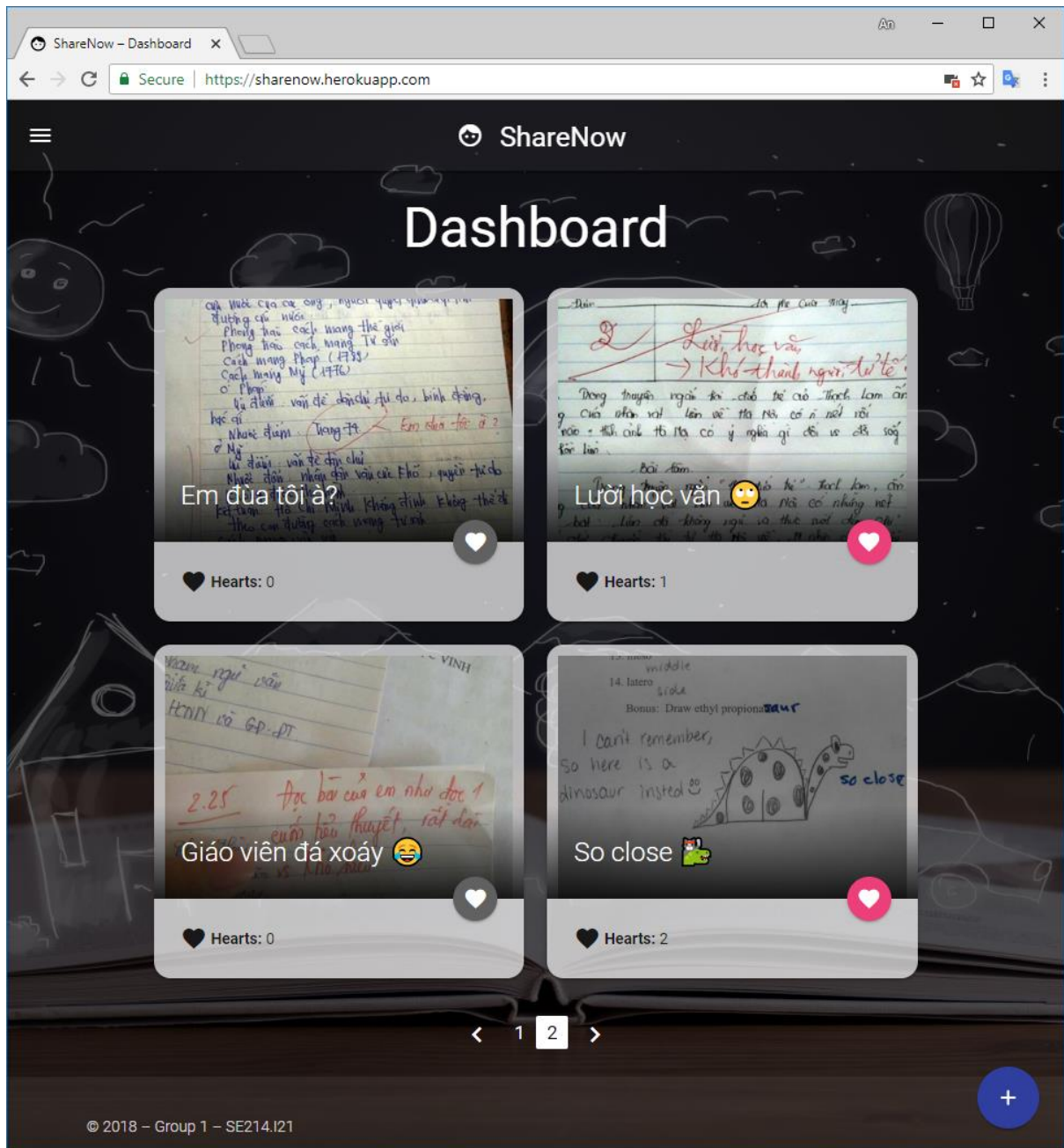
- Gõ lệnh *git push heroku master -f* để tiến hành deploy project lên Heroku server.

```
git push heroku master -f
remote:      Downloading and installing node 8.9.4...
remote:      Bootstrapping npm 5.8.0 (replacing 5.6.0)...
remote:      npm 5.8.0 installed
remote:
remote: -----> Restoring cache
remote:      Loading 2 from cacheDirectories (default):
remote:      - node_modules
remote:      - bower_components (not cached - skipping)
remote:
remote: -----> Building dependencies
remote:      Installing node modules (package.json + package-lock)
remote:      up to date in 4.54s
remote:      Running heroku-postbuild
remote:
remote:      > sharenow@1.0.0 heroku-postbuild /tmp/build_e232d3217867bb735c4d4abba7fcf5
11
remote:      > NPM_CONFIG_PRODUCTION=false npm install --prefix client && npm run build
--prefix client
remote:
```

- Đợi một lúc để Heroku bắt đầu quá trình install dependencies và build project. Khi thấy xuất hiện *Verifying deploy... done* nghĩa là project đã được deploy thành công.

```
remote:
remote: -----> Pruning devDependencies
remote:      Skipping because NPM_CONFIG_PRODUCTION is 'false'
remote:
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:      Procfile declares types      -> (none)
remote:      Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:      Done: 67.4M
remote: -----> Launching...
remote:      Released v45
remote:      https://sharenow.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sharenow.git
+ a1d4fae...122d47e master -> master (forced update)
D:\Workspace\Github\ShareNow>
```

- Vào **Project URL** để kiểm tra lại.



## PHẦN 6: PHỤ LỤC

### 1. Một số cách tối ưu React Application trên các công cụ tìm kiếm (SEO)

#### a. Thêm viewport meta tag:

```
<head>
  ...
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  ...
</head>
```

Nếu không có viewport meta tag, các thiết bị di động sẽ hiển thị các trang ở độ rộng màn hình máy tính thông thường, sau đó sẽ scale các trang cho vừa với màn hình của thiết bị di động. Viewport meta tag sẽ giúp kiểm soát chiều rộng và tỉ lệ của viewport.

#### b. Luôn có <title> tag ở mỗi trang:

Một HTML document cần phải có <title> tag để cung cấp cho người dùng một cái nhìn tổng quan về nội dung. Và khi có sự hiện diện của <title> tag thì tag này không được rỗng.

#### c. Luôn trả về đúng HTTP status code:

Khi một trang được yêu cầu, phải chắc chắn rằng server luôn trả về status code 2XX hoặc 3XX. Search engine sẽ không index một cách đúng đắn khi index web application nhưng server lại trả về status code 4XX hoặc 5XX.

#### d. Sử dụng anchor tag với miêu tả đầy đủ:

Đối với những liên kết có thể click vào, văn bản mô tả là những dấu hiệu hữu ích để người dùng và các search engine hiểu rõ hơn về nội dung của trang. Search engine càng biết nhiều về trang, thông qua nội dung, tiêu đề, liên kết, ..., chúng sẽ trả về những kết quả có độ liên quan cao hơn cho người dùng đang tìm kiếm, điều này sẽ mang lại một lượng lớn các khách truy cập tiềm năng cho trang.

### **e. Sử dụng React với React Router v4:**

React Router là một component để tạo ra liên kết giữa các component hoặc các trang khác nhau. React Router giúp xây dựng một React application với cấu trúc URL thân thiện với các search engine.

## **2. Tài liệu tham khảo:**

- Progressive Web Apps – Google Developer

<https://developers.google.com/web/progressive-web-apps/>

- Progressive Web Apps—The Next Step in Web App Development

<https://hackernoon.com/progressive-web-apps-the-next-step-in-web-app-development-372235bf9a99>

- Service Workers: an Introduction – Google Developer

<https://developers.google.com/web/fundamentals/primers/service-workers/>

- Heroku – Deploying with Git

<https://devcenter.heroku.com/articles/git>

- Best SEO practices for React websites

<https://it-consultis.com/blog/best-seo-practices-for-react-websites>