CLOUD COMPUTING
REPORT ASSIGNMENT 2

STUDENT NAME: TRAN QUANG HUY
STUDENT ID: GCD18457
Class: GCD0703

ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 16: Cloud Computing		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Trần Quang Huy	**Student ID**	GCD18457
Class	GCD0703	**Assessor name**	Tran Trong Minh
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	Huy

Grading grid

P5	P6	P7	P8	M3	M4	D2	D3

☐ Summative Feedback:	☐ Resubmission Feedback:

Grade:	Assessor Signature:	Date:

Signature & Date:

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF PICTURES

INTRODUCTION

Traditionally, businesses have to manage all the activities related to procurement, delivery, and commissioning of their IT infrastructure. Nowadays, most of the IT-related services are outsourced to vendors who provide these computing services through the internet. These vendors have a huge collection of computing resources (called Cloud Computing) that provide computing services like storage, database servers, software, and networking. These services are flexible and economical, which means that a user pays only for the type of service they use and for how much time they use.

Cloud computing is perhaps the most flamboyant technological innovation of the 21st century. This is because it has seen the fastest adoption into the mainstream than any other technology in the domain. This adoption has been fueled mainly by the ever-increasing number of smartphones and mobile devices that can access the internet. Cloud computing is not just for organizations and businesses; it's also useful for the average person as well. It enables us to run software programs without installing them on our computers; it enables us to store and access our multimedia content via the internet, it enables us to develop and test programs without necessarily having servers and so on. Cloud computing is a 21st-century marvel that holds its importance in almost every field the company can think of.

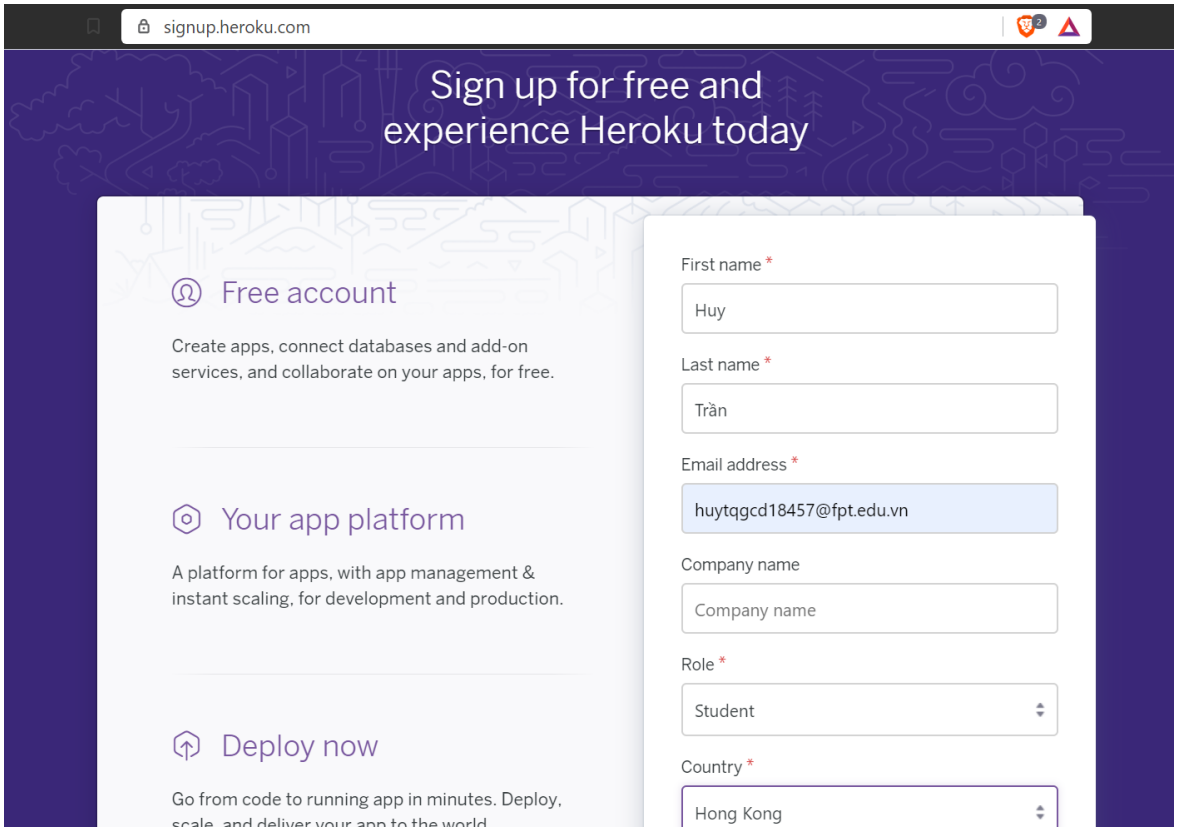
The development of Cloud Computing solutions using the service provider's framework and open source tools

1. The configuration Cloud Computing platform with a cloud service provider's framework (P5)

1.1. Heroku registration and configuration

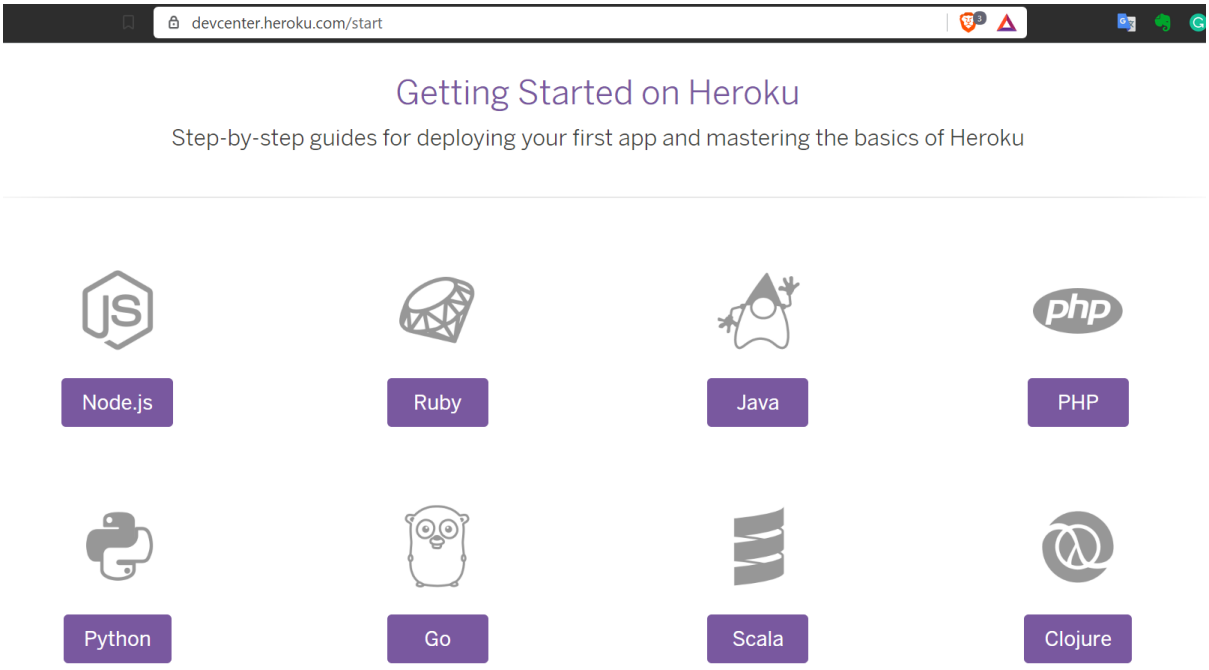
Picture 1. Heroku logo

When starting the project, go to Heroku.com and create an account. At first glance, the dashboard is incredibly simple and user friendly.



Picture 2. Sign up on Heroku

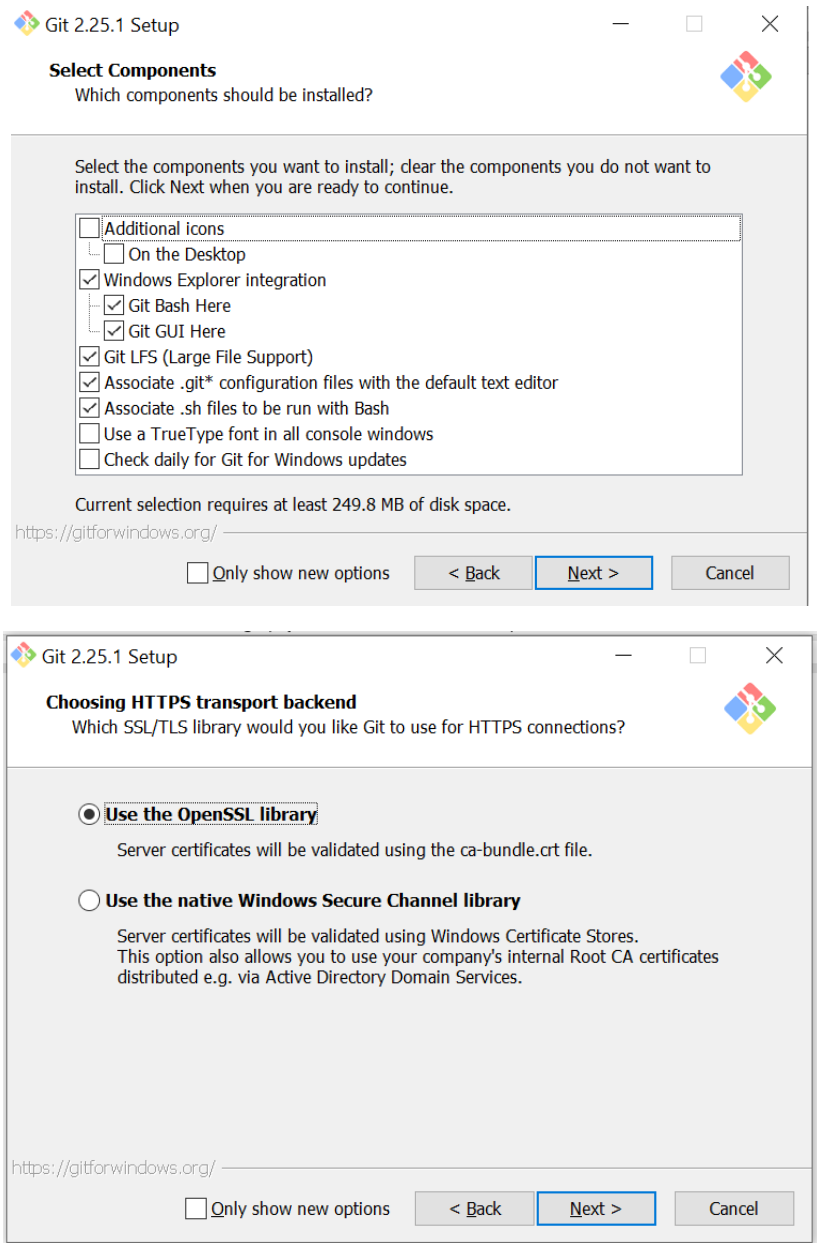
It gives users great Getting Started with Heroku dialog where users can find the instructions for each type of app that can deploy.



Picture 3. Getting started on Heroku

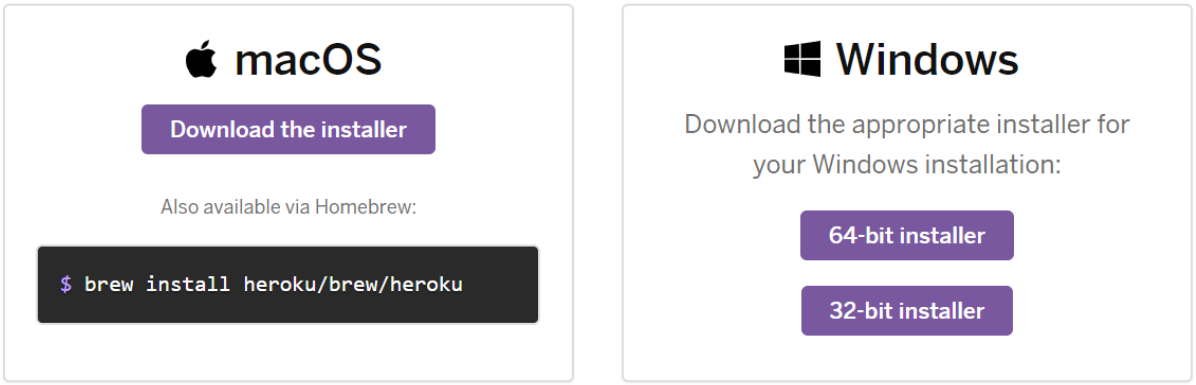
The Heroku CLI requires Git, the popular version control system.

Picture 4. Download Git

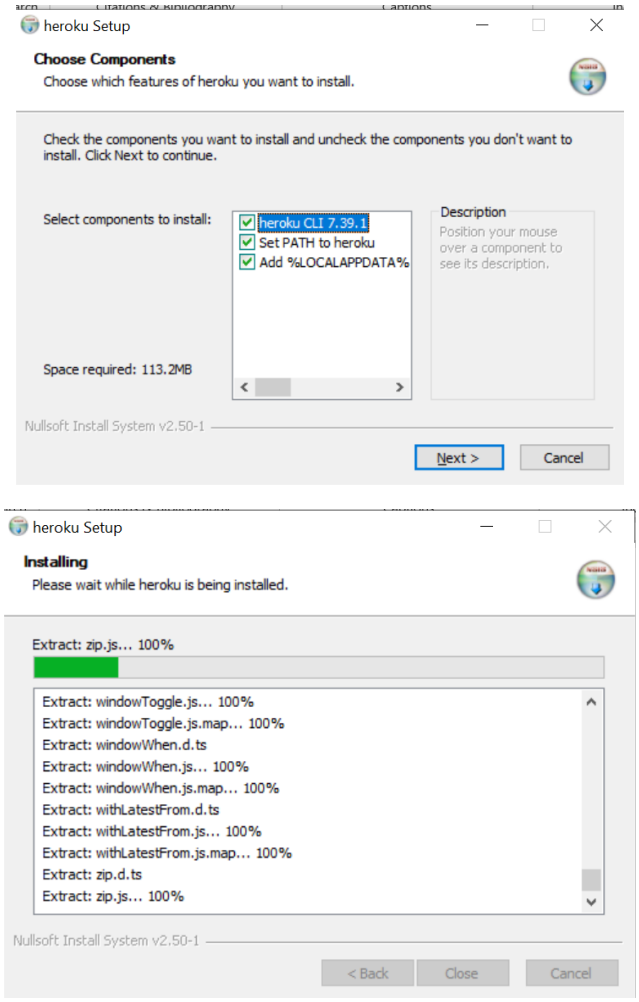


Picture 5. Git setup

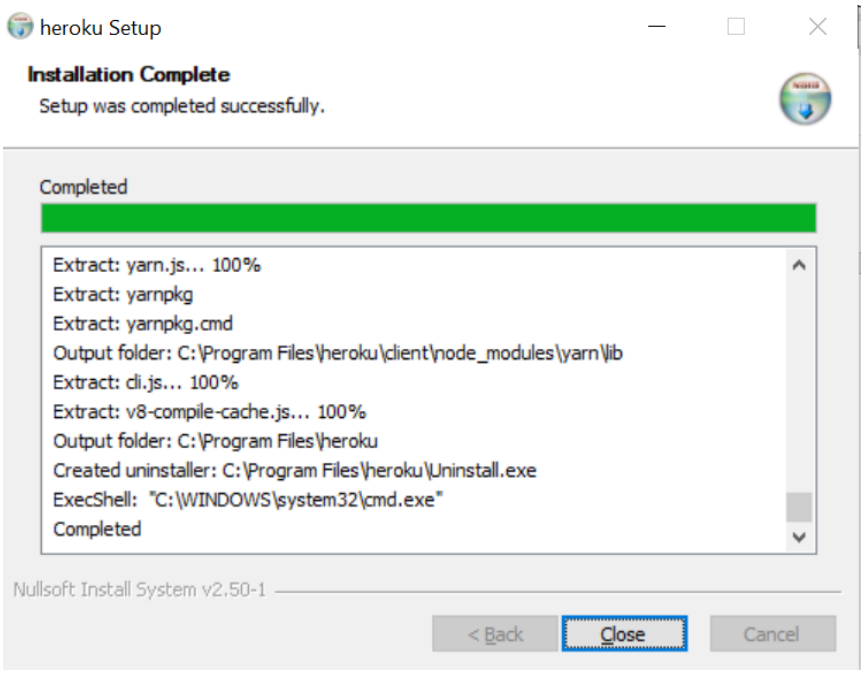
This step will install the Heroku Command Line Interface (CLI). The CLI to manage and scale applications, provision add-ons, view application logs and run applications locally.



Picture 6. Download Heroku CLI



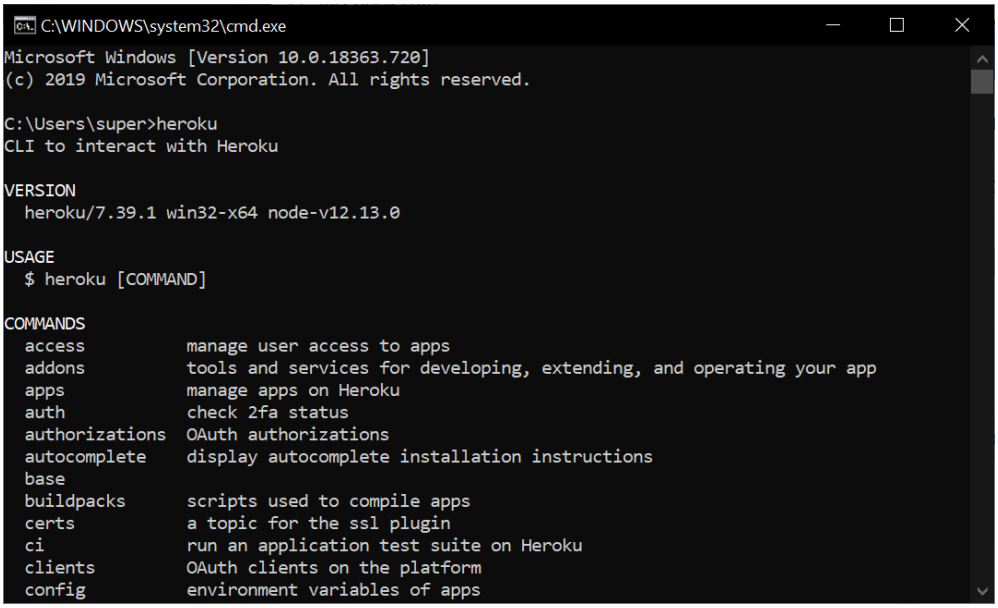
Picture 7. Setup Heroku



Picture 8. Complete install Heroku CLI

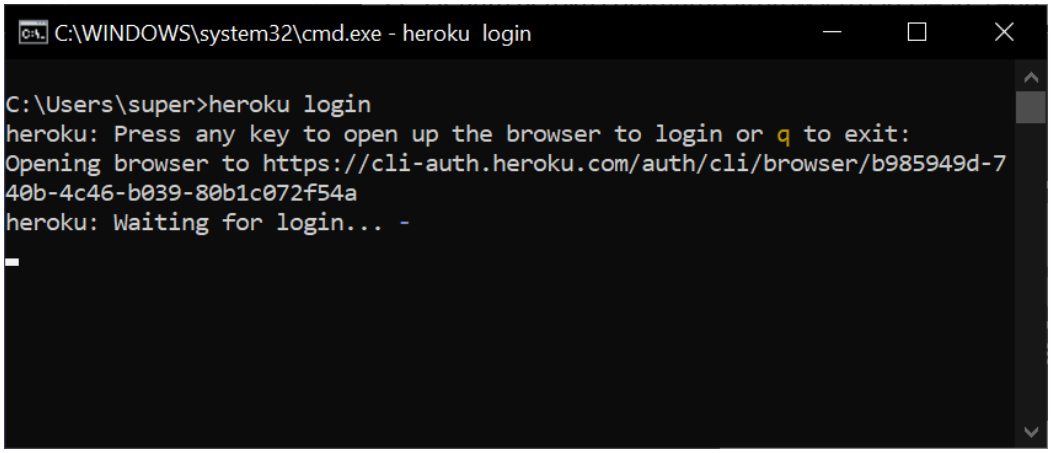
Checking the Heroku CLI: On Windows, start the Command Prompt (cmd.exe) or Powershell to access the command shell.

```
$ heroku
```

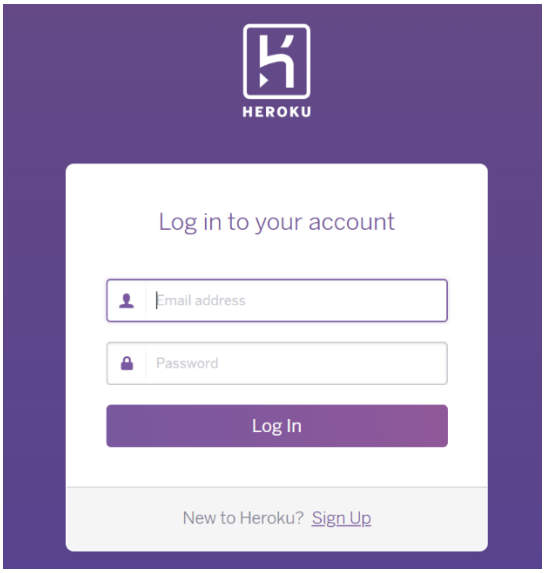


Picture 9. Heroku CLI

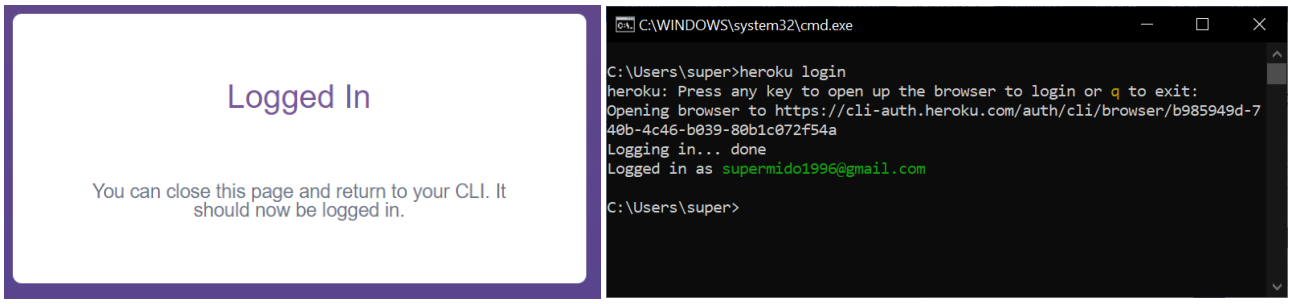
Use the Heroku login command to log in to the Heroku CLI:



Picture 10. Heroku login



Picture 11. Heroku require login on a browser



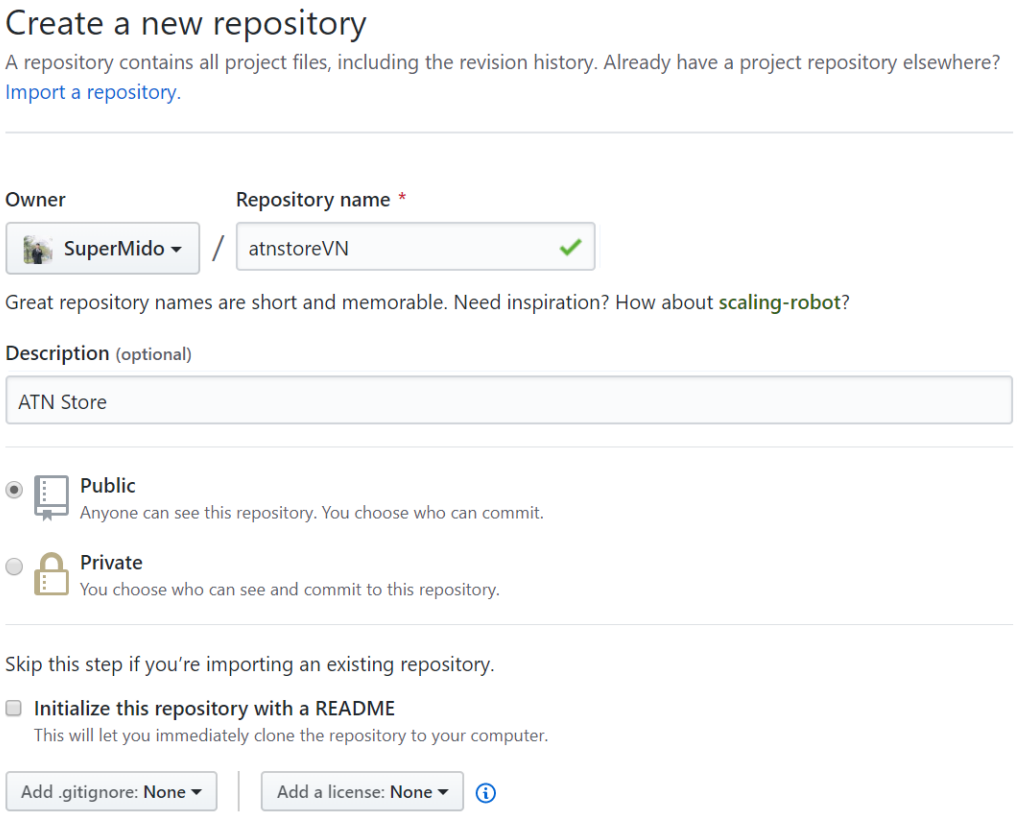
Picture 12. Heroku login successful

1.2. GitHub repository configuration

Picture 13. GitHub logo

A repository is like a folder for the project. The project's repository contains all of the project's files and stores each file's revision history. Users can also discuss and manage the project's work within the repository.

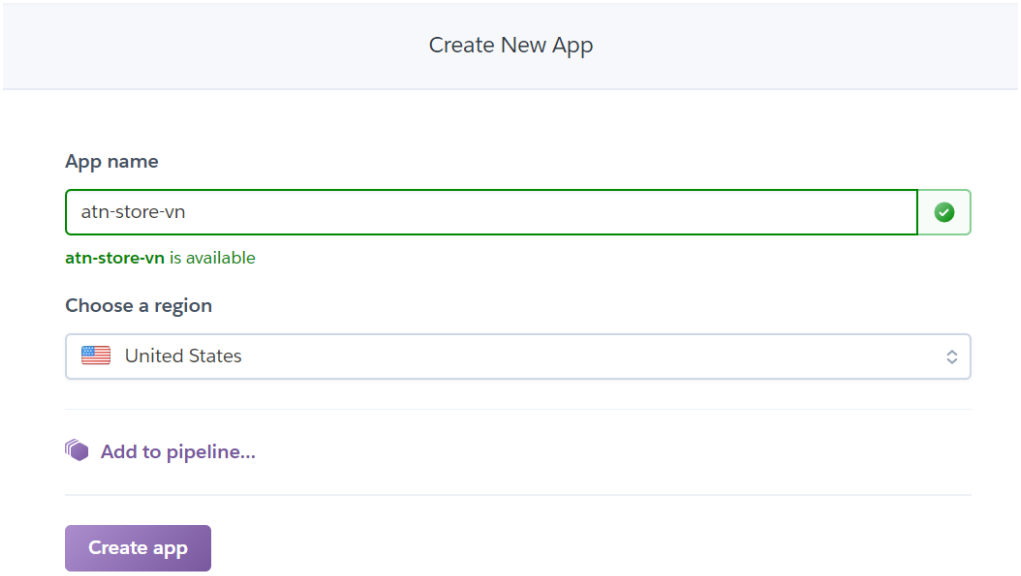
Users can own repositories individually, or share ownership of repositories with other people in an organization.



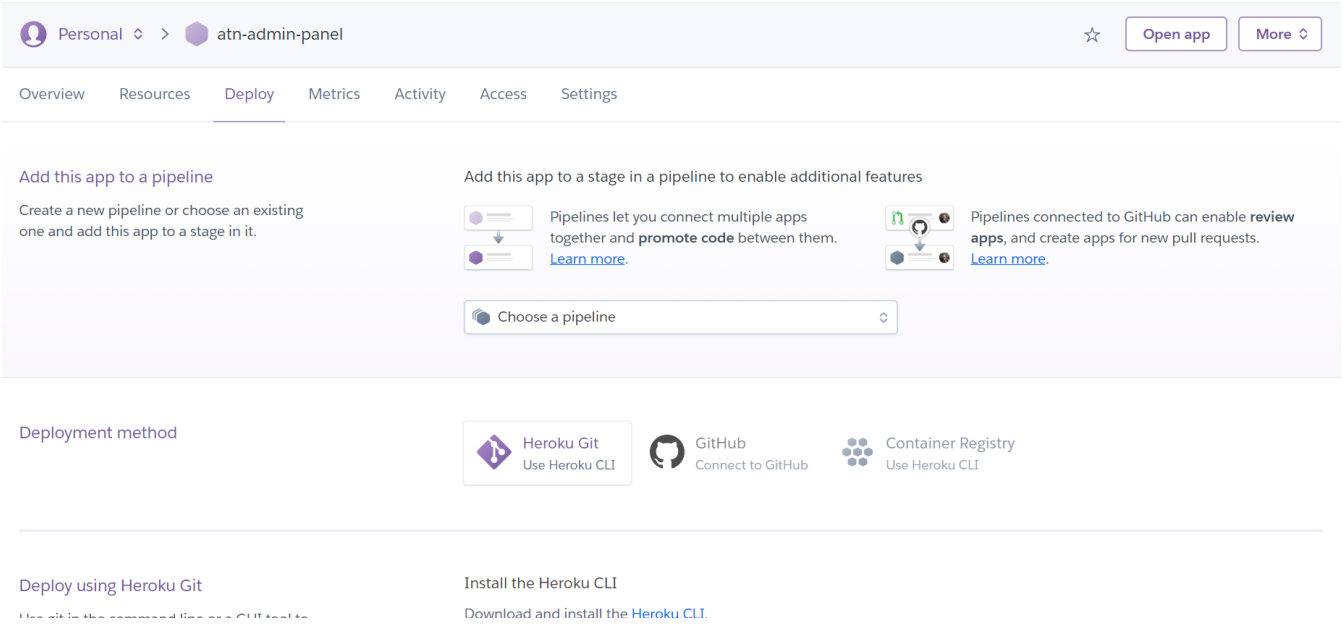
Picture 14. Create a new repository on GitHub

1.3. Heroku cloud application registration for ATN Store

With just the click of a button, users can deploy a preconfigured app that has everything it needs, including code, configurations, and add-ons. By creating the first Heroku app with a button, the user can get a flavor for how easy the platform is to use and also get a real, functioning Node.js app that users can explore and modify to learn more.

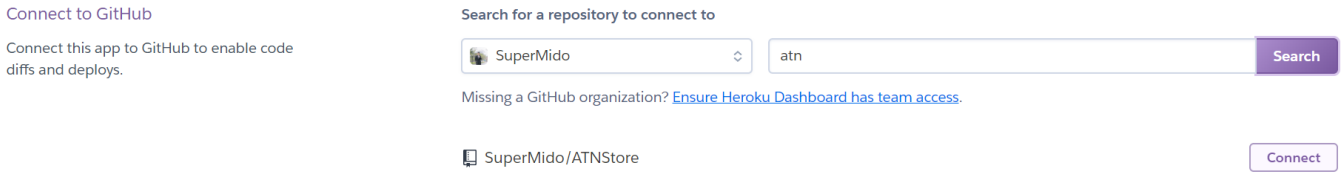


Picture 15. Create a new app on Heroku



Picture 16. Heroku app manager

The user will change the source code for the application and deploy that change using one of the easiest Heroku app deployment methods, GitHub integration. GitHub is a web-based source code management hosting service and it can be configured to automatically deploy code changes to Heroku.



Picture 17. Heroku connect to GitHub

The user now has source code for the application in GitHub repository, and have connected it to Heroku for automatic deployment whenever anything changes in that source code.

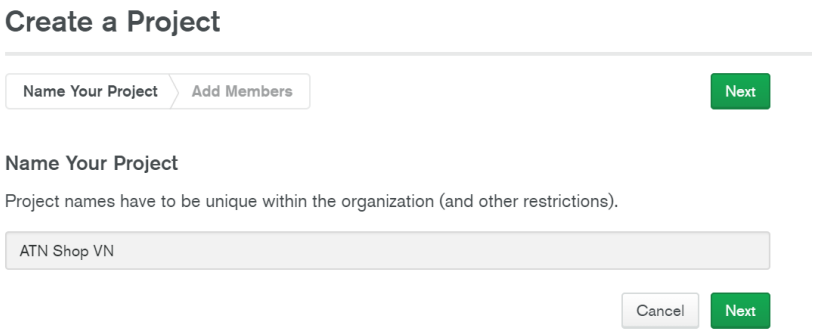
1.4. MongoDB Atlas database configuration

MongoDB Atlas delivers the world's leading database for modern applications as a fully automated cloud service with the operational and security best practices built-in. Easily deploy, operate, and scale databases across the leading cloud platforms.

Picture 18. MongoDB atlas logo

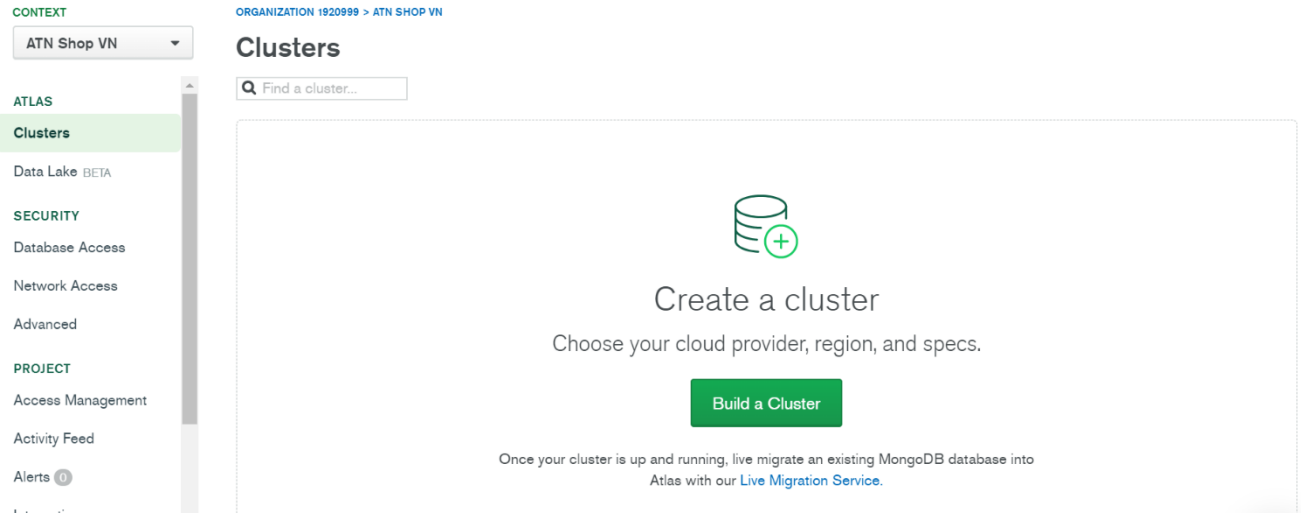
MongoDB Atlas is a fully-managed cloud database developed by the same people that build MongoDB. Atlas handles all the complexity of deploying, managing, and healing deployments on the cloud service provider of AWS, Azure, and GCP.

When working with MongoDB atlas, the user needs to create a project:



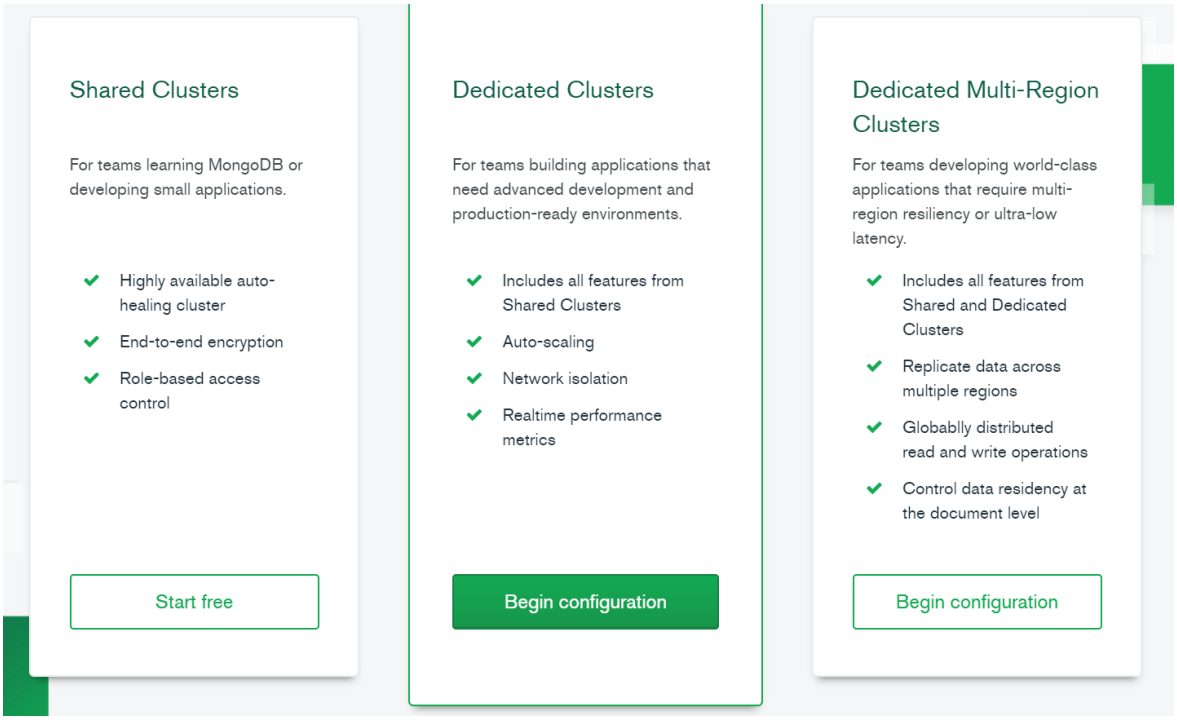
Picture 19. Create a Project with atlas

Then to configure the database, the project needs at least one cluster



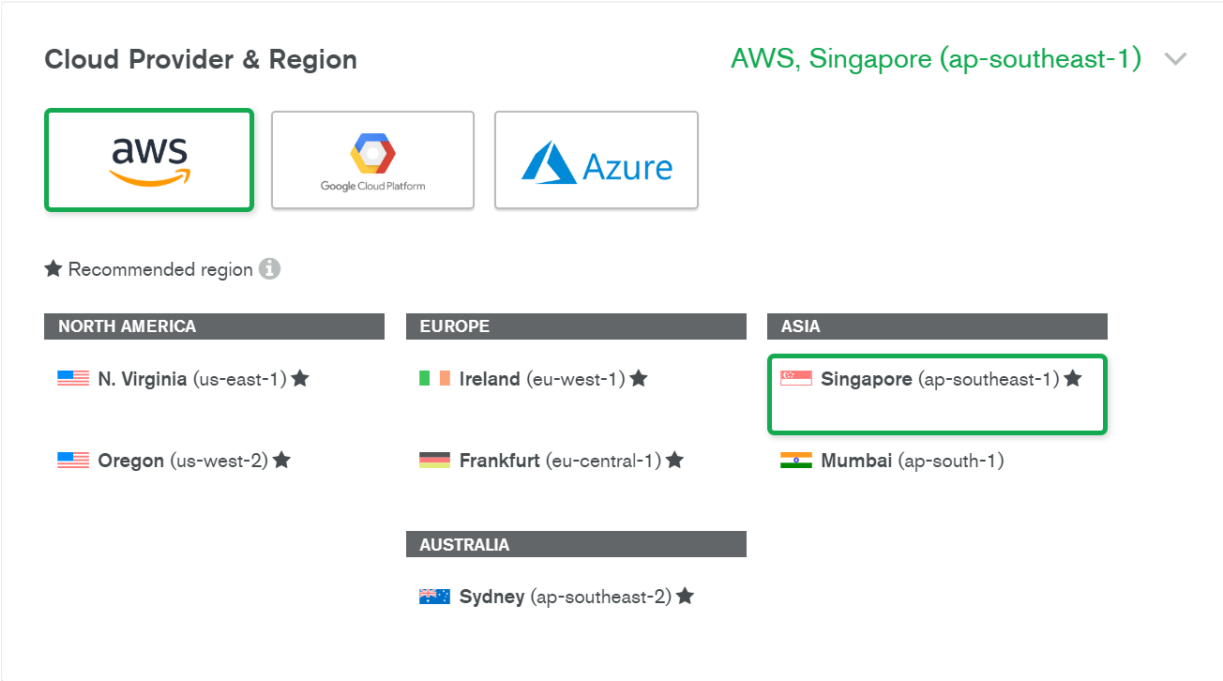
Picture 20. Create a cluster with MongoDB Atlas

In the ATN Project will start with Shared Clusters



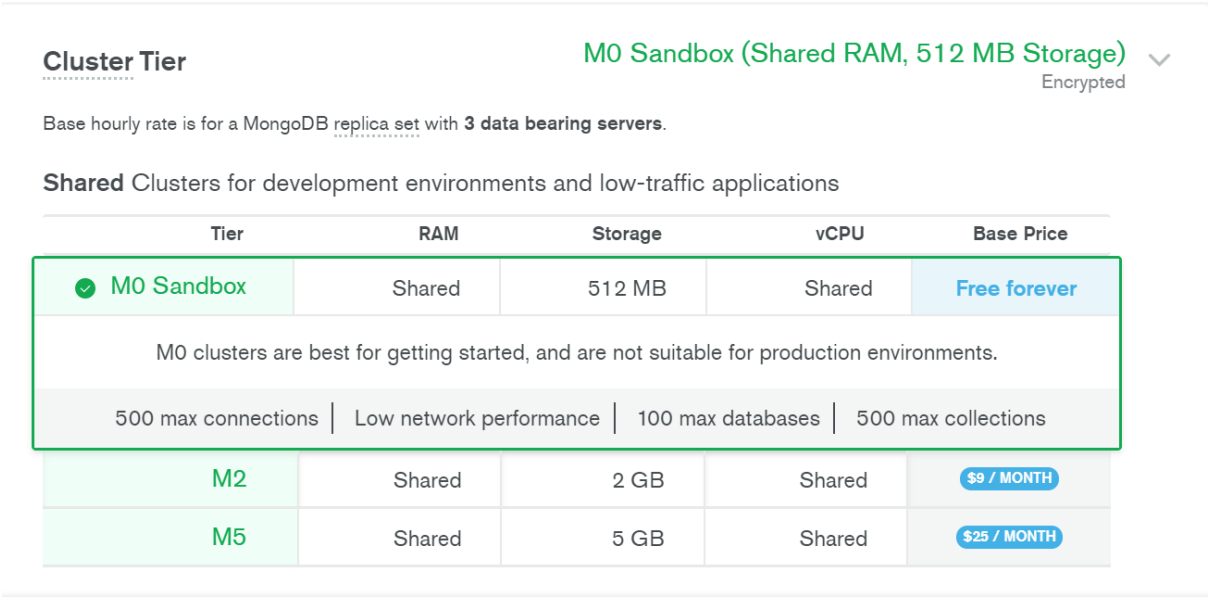
Picture 21. Kind of clusters on MongoDB atlas

Cloud provider and region for ATN project are AWS and Singapore (ap-southeast-1) for the fastest connection from application to MongoDB



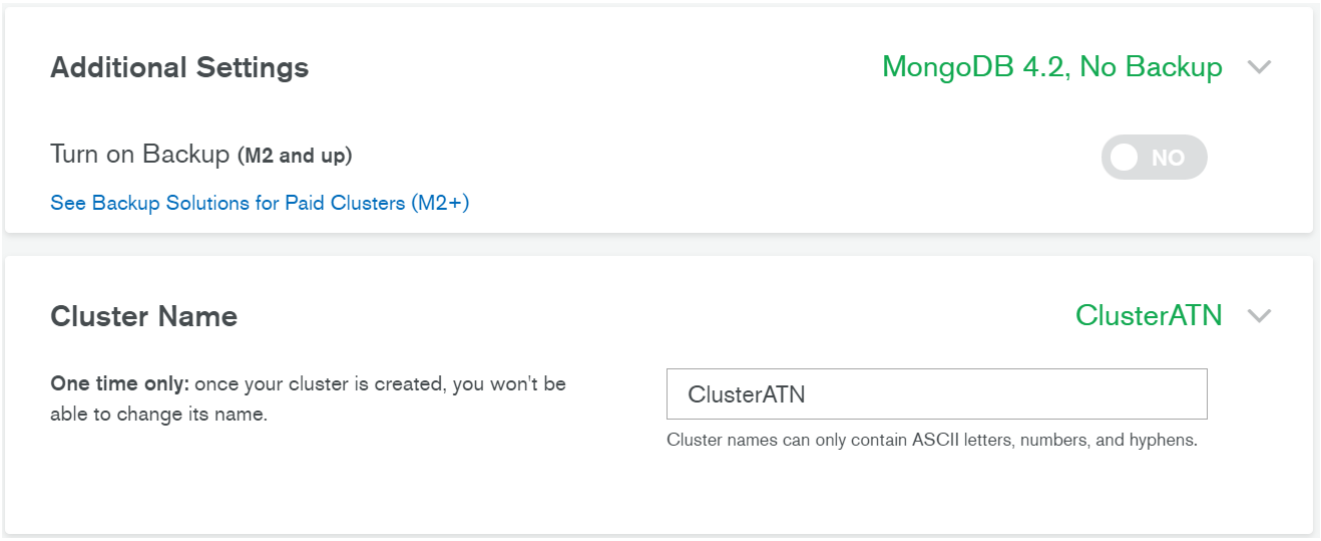
Picture 22. Configure cloud provider and region on MongoDB Atlas

There are many Cluster tier for users to choose, in some cases of situation, the company will pay more money to get more storage. But in this case, will be M0 SandBox



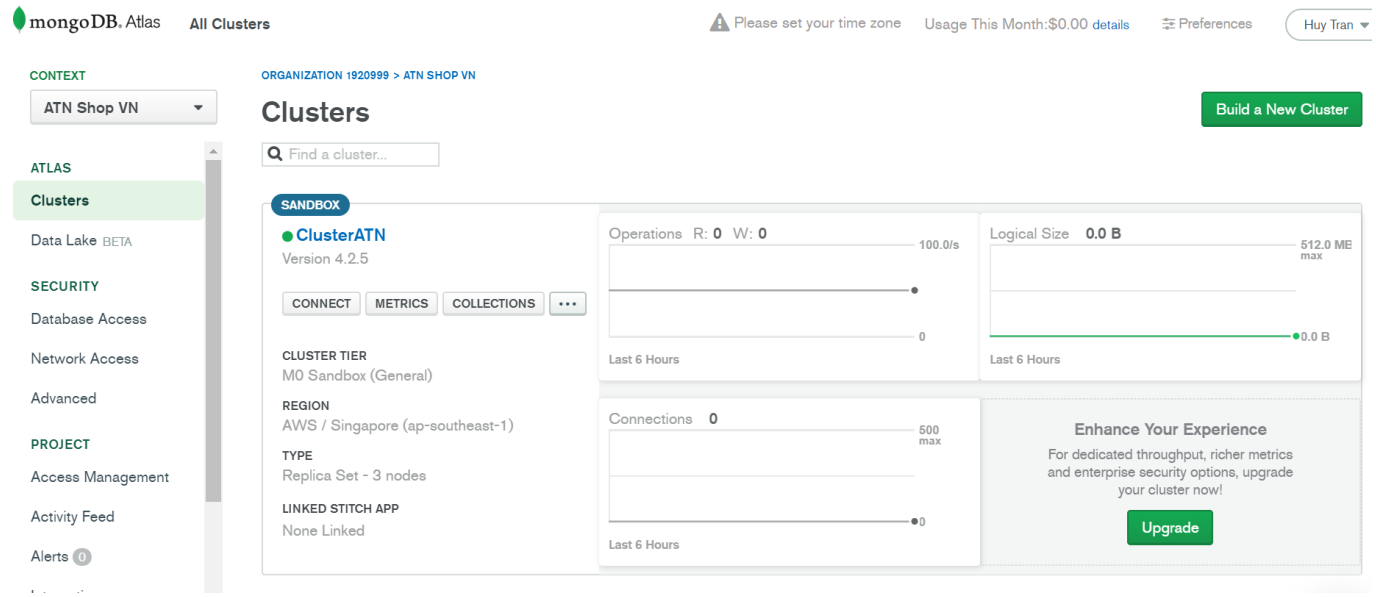
Picture 23. Cluster Tier on MongoDB Atlas

Users can set up the back-up with (M2 Cluster Tier), but ATN Store will consider for this case.



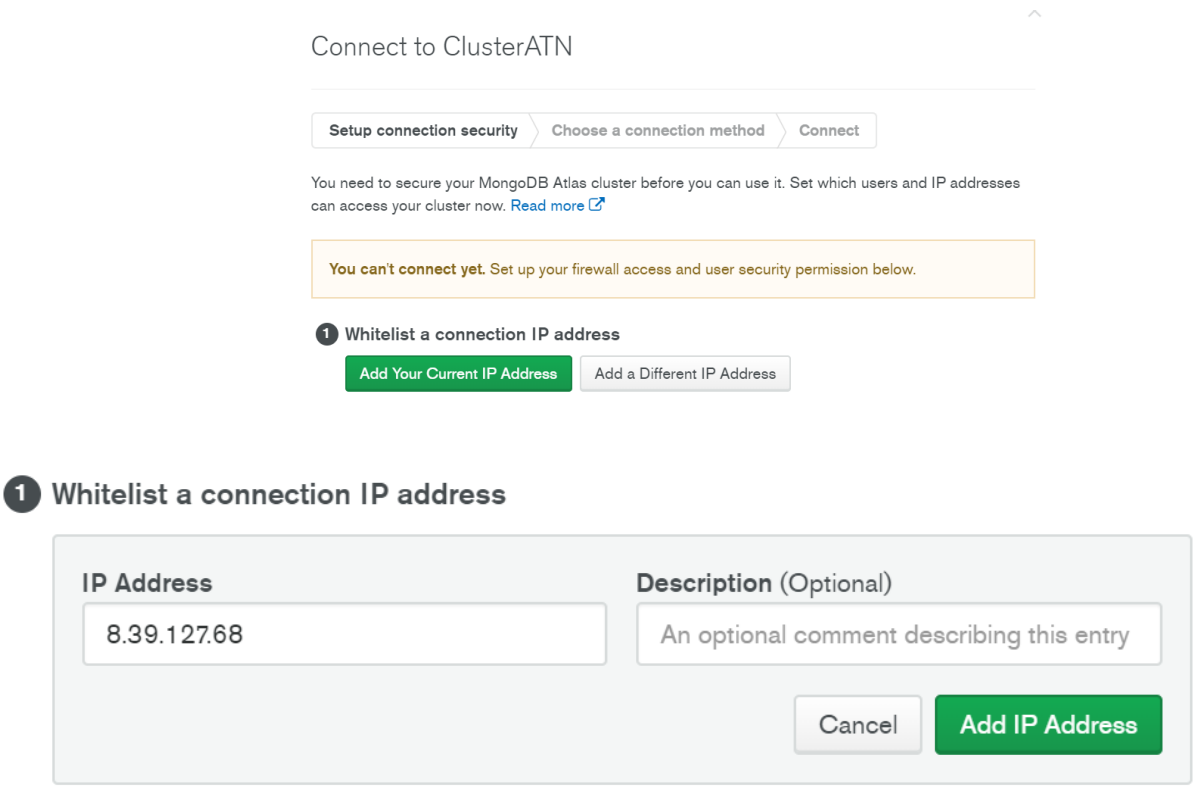
Picture 24. Addition settings MongoDB Atlas

Checking the configuration about MongoDB Atlas at a dashboard:



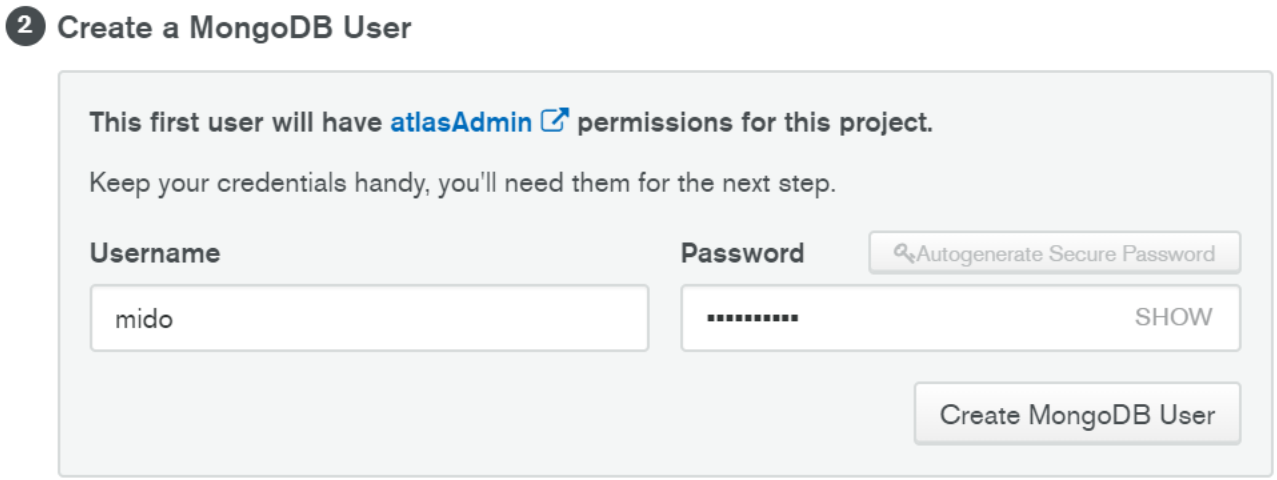
Picture 25. Dashboard MongoDB Atlas

The important in the configuration is set up the connection from Client to Server, so the first thing is Add a Whitelist IP



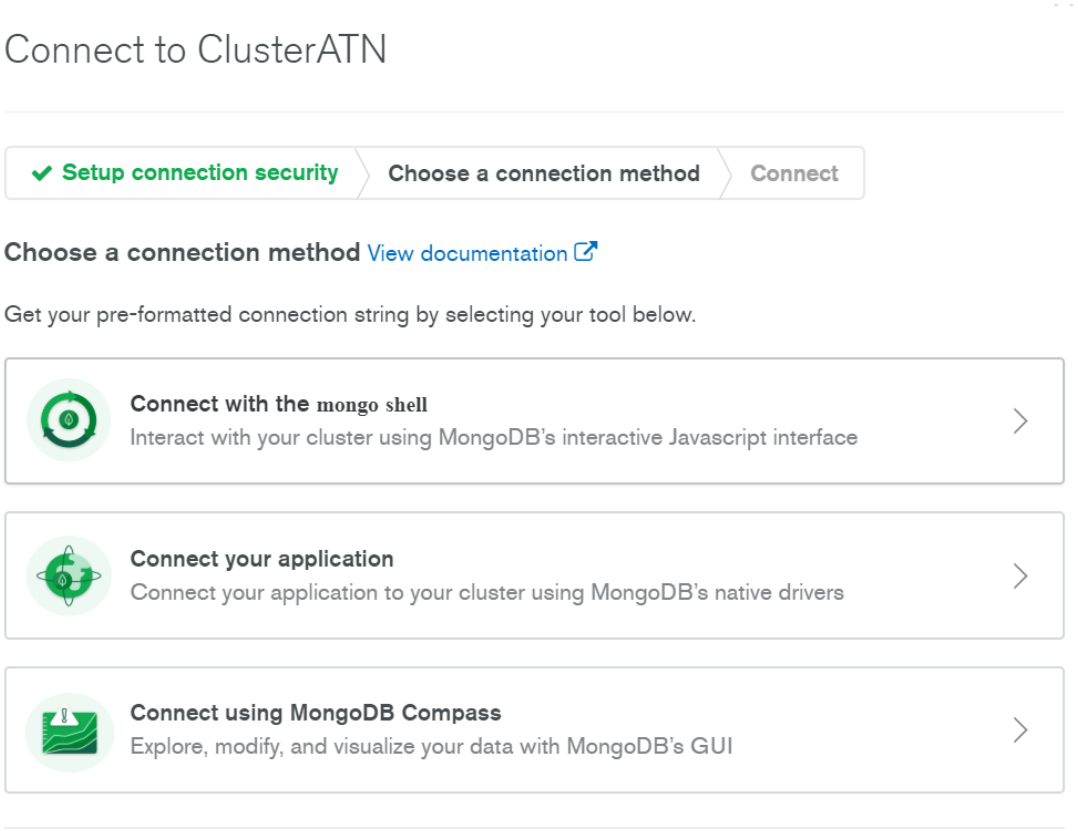
Picture 26. Add whitelist for MongoDB Atlas

Create a MongoDB User:



Picture 27. Create a MongoDB User

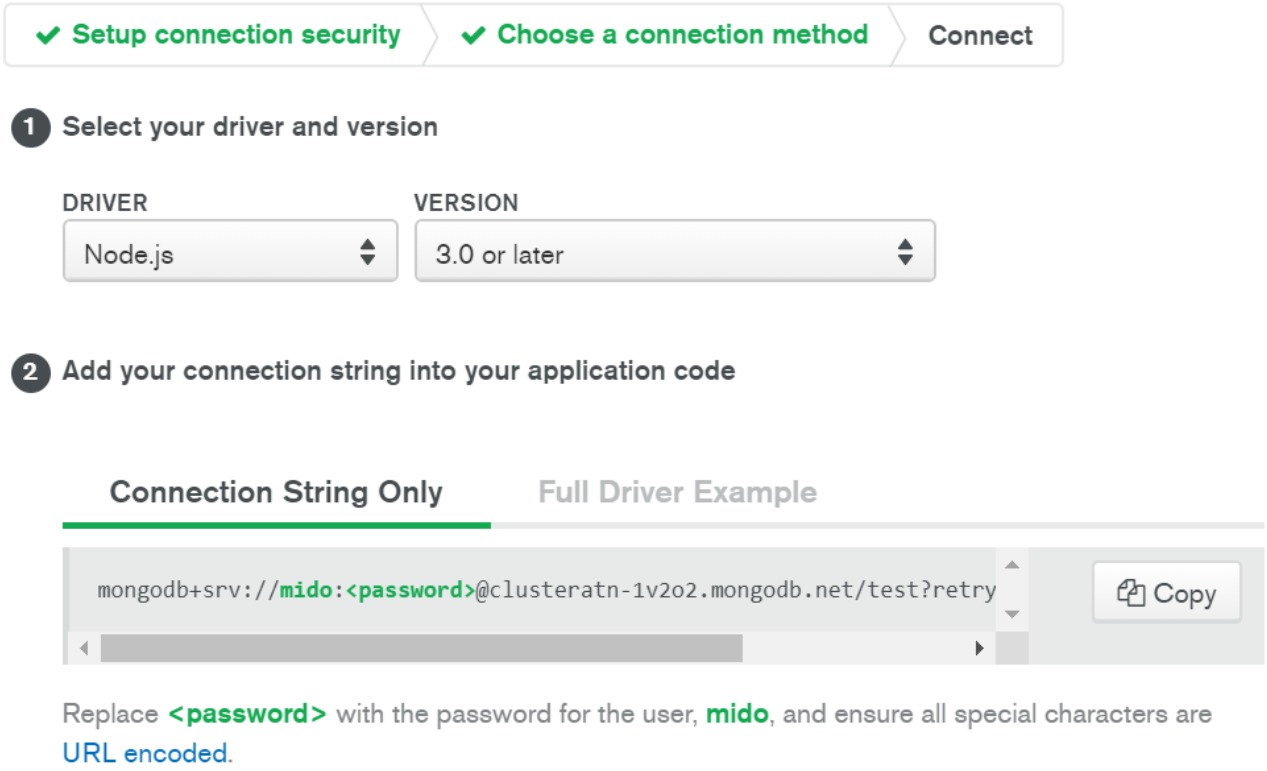
Connect application to the cluster using MongoDB's native drivers



Picture 28. Connect to Cluster

The ATN application is built with Node.js version 12.16, so the selection for connection method:

- **Driver**: Node.Js
- **Version**: 3.0 or later



Picture 29. MongoDB connection method

2. The implementation cloud platform using the open-source (P6)

2.1. Back-end development

- **Node.Js:**

Picture 30. NodeJs logo

JavaScript is a programming language and **data format** (JSON) has changed web development drastically. Integrating Node.js with it to do things on the server as well as in the browser is a trend lately. These two sentences, we feel, have to be illuminated and explained for everyone to grasp. So in this project, NodeJs is the best way to build the system.

There are many benefits of Node.Js:

- Node JS is a server framework and is free
- It runs on Windows, Linux, Mac OS, etc.
- Node utilizes JavaScript on the server
- It contains tasks and executes them upon set events
- Generate dynamic content
- Create, open and read, or delete files on the server
- Gather and modify data in the database

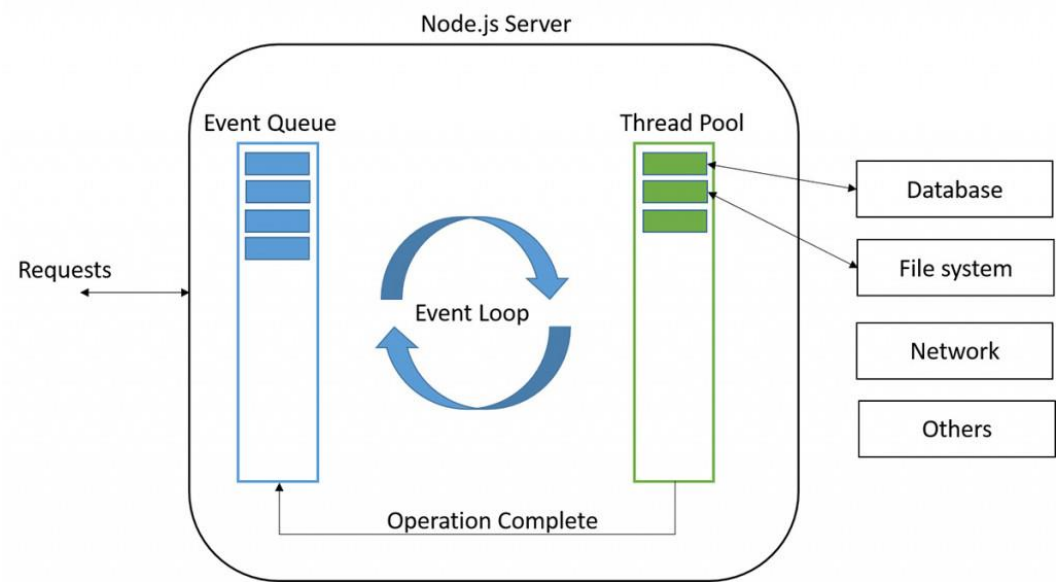


Figure 1. How Node.js Server work

- **ExpressJs:**

ExpressJS is a prebuilt **NodeJS** framework that can help ATN company in creating server-side web applications faster and smarter. Simplicity, minimalism, flexibility, scalability is some of its characteristics and since it is made in NodeJS itself, it inherited its performance as well.

In short, ExpressJS did for NodeJS what Bootstrap did for HTML/CSS and responsive web design.

It made coding in NodeJS a piece of cake and gave programmers some additional features to extend their server-side coding. ExpressJS is hands down the most famous NodeJS framework- so much so that when most people talk about NodeJS they surely mean NodeJS+ExpressJS. (algoworks, n.d.)

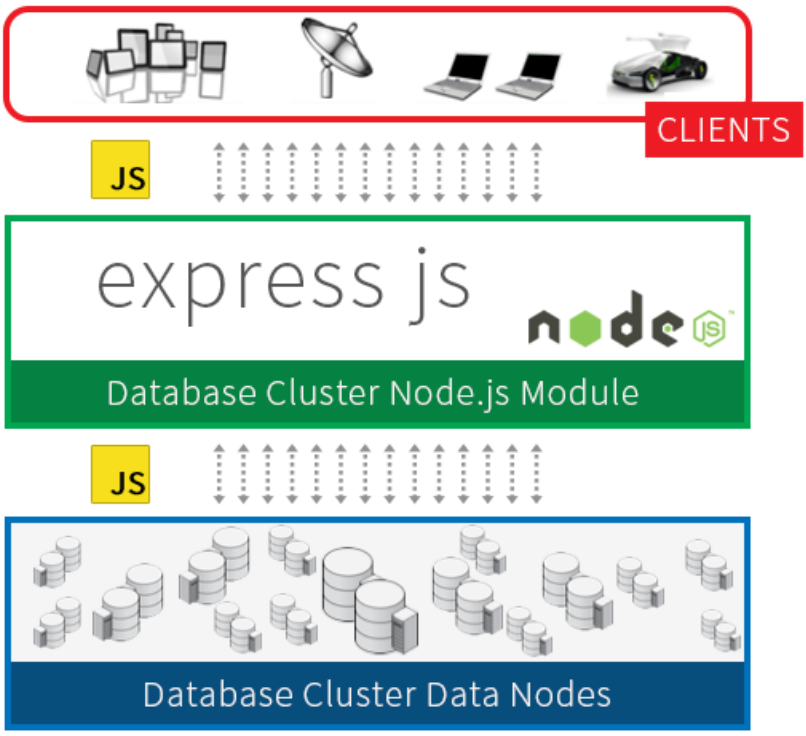
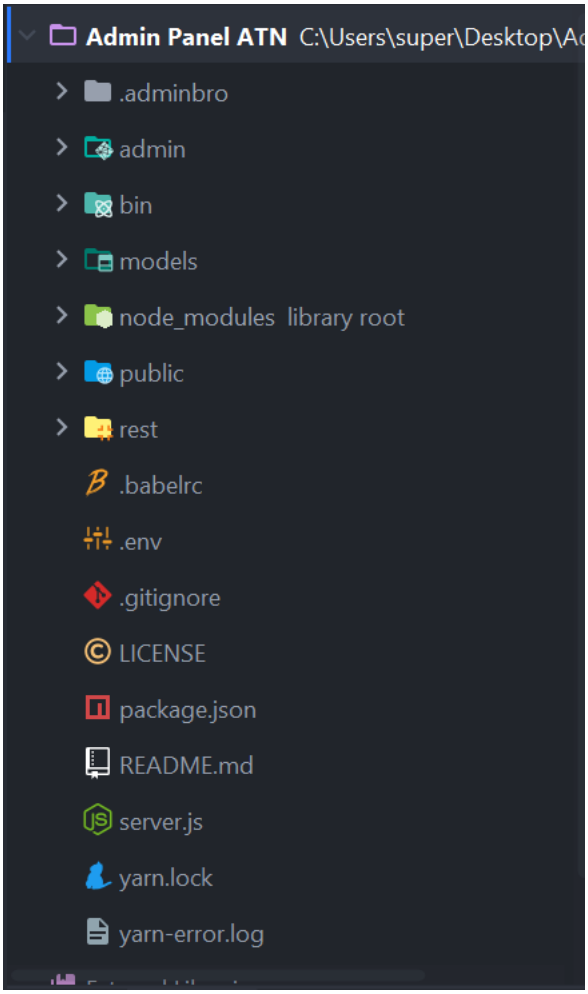


Figure 2. How ExpressJs work

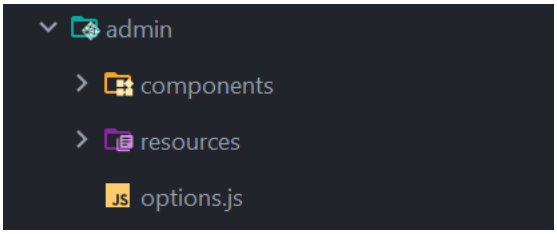
The only point developer is trying to make here is that can cut down programming time in half through ExpressJS. Also since NodeJS and ExpressJS are written in JavaScript, a very easy language to learn and manipulate, the framework is highly scalable and flexible.

In ATN Admin Panel Project will be built seem like the MVC model with Models, views, and controller:



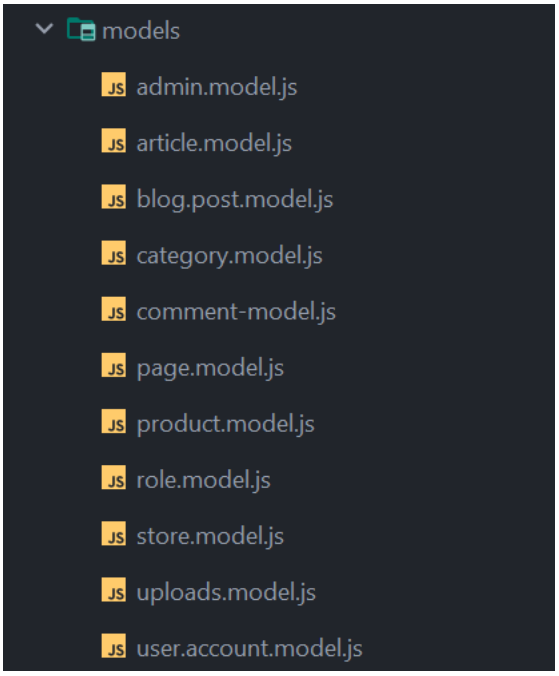
Picture 31. The constructor of the ATN Admin Panel Project

- **Admin**: To store the component and option for front-end (View)



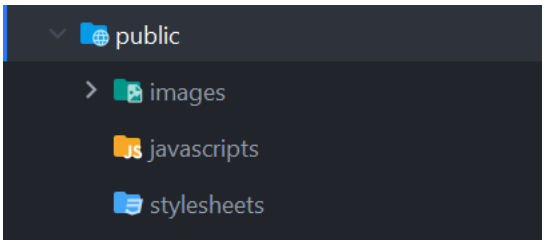
Picture 32. Admin folder

- **Models**: To store the models (Database)



Picture 33. models folder

- **Public**: To store the image, CSS, font



Picture 34. Public folder

- **Server.js**: The main system, will connect the database and render the interface of ATN application

```
const mongoose = require('mongoose')
const AdminBro = require('admin-bro')
const AdminBroExpress = require('admin-bro-expressjs')
const AdminBroOptions = require('./admin/options')
require('dotenv').config()

const express = require('express')
const app = express()

const UserAccountModel = require('./models/user.account.model')

const adminBro = new AdminBro(AdminBroOptions)

const router = AdminBroExpress.buildAuthenticatedRouter(adminBro, {
  authenticate: async (email, password) => {
    const user = await UserAccountModel.findOne({ email })
    if (user){
      if(password === user.auth.password)
        return user
    }
    return null
  },
  cookieName: 'adminbro',
  cookiePassword: 'somepassword',
})


app.use(adminBro.options.rootPath, router)
app.use(express.static('./public'));
app.get('/', router)

const run = async () => {
  await mongoose.connect(process.env.MONGO_URL, {
    useNewUrlParser: true,
    useUnifiedTopology: true
  })
  app.listen(process.env.PORT, () => console.log(` Admin Panel is under localhost:${process.env.PORT}` ))
}

run()
```

- **Package.json**: will store information about the package, that will make the cloud easy to install and run.

```
{
 "name": "example",
 "version": "1.0.0",
 "description": "",
 "main": "index.js",
 "scripts": {
  "start": "nodemon --inspect server.js"
 },
 "author": "",
 "license": "ISC",
 "dependencies": {
  "@material-ui/core": "^4.8.2",
  "admin-bro": "^2.2.4",
  "admin-bro-expressjs": "^2.0.4",
  "admin-bro-mongoose": "^0.5.1",
  "admin-bro-theme-dark": "^1.0.0",
  "axios": "^0.18.0",
  "bcrypt": "^3.0.6",
  "body-parser": "^1.18.3",
  "boom": "^7.3.0",
  "chokidar": "^3.2.2",
  "cookie-parser": "^1.4.4",
  "dotenv": "^8.2.0",
  "express": "^4.17.1",
  "express-brute": "^1.0.1",
  "express-formidable": "^1.2.0",
  "express-session": "^1.17.0",
  "faker": "^4.1.0",
  "inert": "^5.1.2",
  "mongoose": "^5.8.0",
  "nodemon": "^1.18.8",
  "ora": "^4.0.2",
  "pg": "^7.7.1",
  "react": "^16.13.1",
  "react-dom": "^16.13.1",
  "recharts": "^1.8.5",
  "reflect-metadata": "^0.1.13",
  "styled-components": "^5.1.0",
  "typeorm": "^0.2.20",
  "typescript": "^3.7.3"
 },
 "devDependencies": {
  "@babel/core": "^7.8.3",
  "@babel/plugin-proposal-decorators": "^7.6.0",
  "@babel/preset-env": "^7.8.3",
  "@babel/preset-react": "^7.8.3",
  "@babel/preset-typescript": "^7.8.3",
  "@babel/register": "^7.8.3",
  "@babel/runtime": "^7.8.3",
  "babel-plugin-transform-class-properties": "^6.24.1"
 }
}
```

- **Database schema design:**

The database of ATN will build with **NoSQL databases**. NoSQL databases store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.

The database of ATN Company should be designed with at least 3 collections that related to each other to create an efficient and clear database architecture that avoids repeating data or anomaly data.

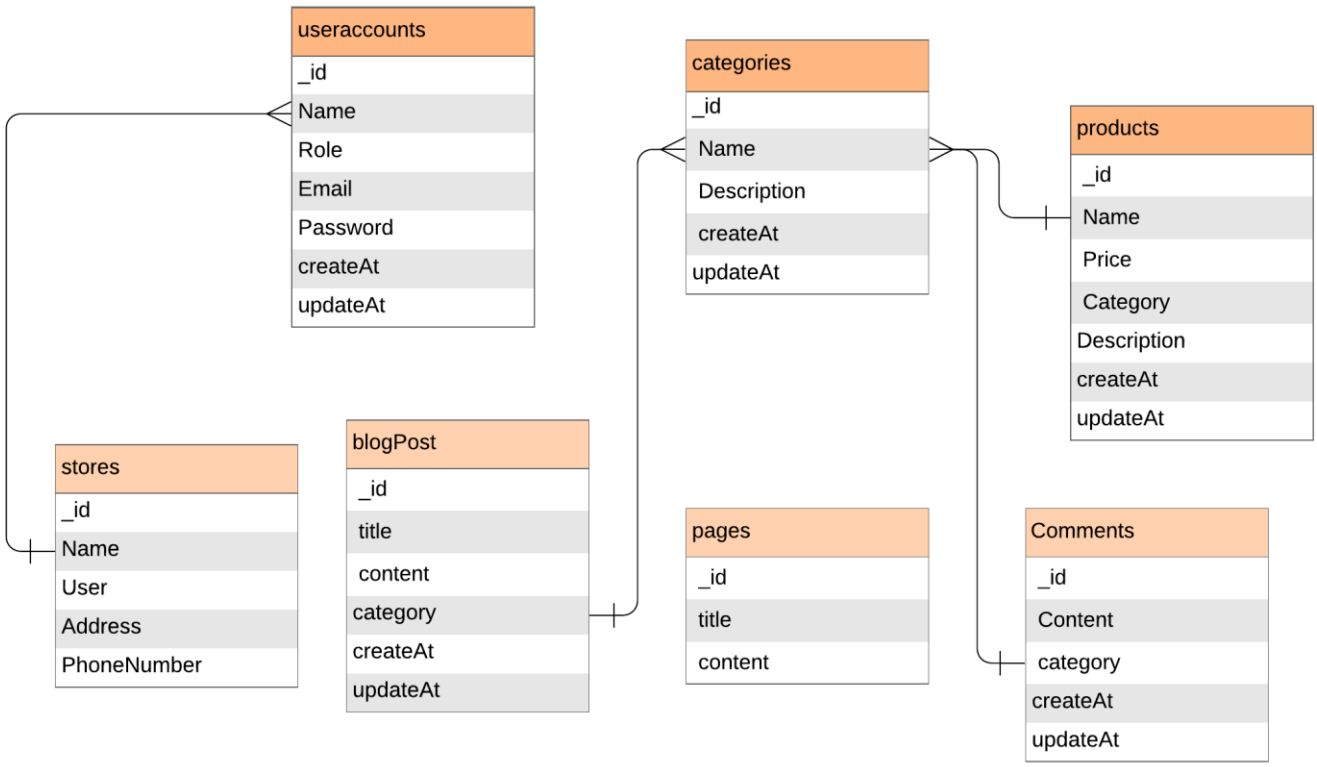


Figure 3. The entity-relationship diagram of ATN database

In order to communicate between the web application interface and the data in the database, the back-end needs some models for each collection of the database:

- **Useraccounts model:**

To create a **User account collections** in the database, the model will help the ATN system to create a collection and type of data will store.

```
const mongoose = require('mongoose')
const { Schema } = mongoose
const EMAIL_REGEXP = /^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$/i;

const validateEmail = email => EMAIL_REGEXP.test(email);

const UserSchema = new Schema({
  name: String,
  email: {
    type: String,
    lowercase: true,
    trim: true,
    required: true,
    validate: [validateEmail, 'Please fill a valid email address']
  },
  auth: {
    password: {
      type: String,
      required: true,
    },
  },
  role: {
    type: String,
    enum: ['Admin', 'Store Manager', 'Sales'],
    required: true
  }
}, { timestamps: true })

const UserAccount = mongoose.model('UserAccount', UserSchema)

module.exports = UserAccount
```

- **Categories model:**

To create **category collections** in the database, the model will help the ATN system to create collections and type of data will store.

```
const mongoose = require('mongoose')

const { Schema } = mongoose

const CategorySchema = new Schema({
  Name: {
    type: String,
    required: true,
  },
  createdAt: Date,
  Description: String,
}, { timestamps: true })

const Category = mongoose.model('Category', CategorySchema)

module.exports = Category
```

- **Products model:**

To create **product collections** in the database, the model will help the ATN system to create collections and type of data will store.

In this **product model,** the data category will link with the **category** from the **category model**.

```
const mongoose = require('mongoose')

const { Schema } = mongoose

const ProductSchema = new Schema({
  Name: String,
  Description: String,
  Price: Number,
  category: {
    type: Schema.Types.ObjectId,
    ref: 'Category',
  }
}, { timestamps: true })

const Product = mongoose.model('Product', ProductSchema)

module.exports = Product
```

- **Stores model:**

To create **Stores collections** in the database, the model will help the ATN system to create collections and type of data will store.

In this **Stores model,** the data of User will link with **User** from the **User Account model**.

```
const mongoose = require('mongoose')

const { Schema } = mongoose

const StoreSchema = new Schema({
  Name: {
    type: String,
    required: true,
  },
  Address: {
    type: String,
    required: true,
  },
  PhoneNumber: {
    type: String,
    required: true,
  },
  User: {
    type: Schema.Types.ObjectId,
    ref: 'UserAccount',
  },
})

const Store = mongoose.model('Store', StoreSchema)

module.exports = Store
```

2.2. Front-end development

In ATN's project, **React JS** is used in web development to build interactive elements on websites.

Picture 35. React logo

There are design patterns and the MVC pattern (Model-View-Controller), that React works as the V in MVC. With React, the user can design all the front-end parts of the application. That means the user can create easily all the interface of the application.

When creating an ATN React app, it's made of two parts: **components**, that are the pieces that contain HTML code and what developer want the user to see, and an HTML document where all components will be rendered.

- **Rendering and optimizatio**n: when ATN React app is loaded in the browser, all the components that developers created are rendered. The only parts of the application that reloads (rendered again) are the ones whose state changes.
- **All the power of HTML, CSS, and Javascript together inside the component**: when developers create a React component, not only the user can use HTML and CSS as usually do, but can also integrate Javascript in a very nice way. The developer can define methods inside components that can use in every part the developer want.
- Every **component** is a Class that ATN can **instantiate**: That means that developer component can receive arguments which developer can use to customize what's shown in the application.
- It's **asynchronous**. There are no blocking actions in the application. If something hasn't finished loading, other components will continue working or loading.

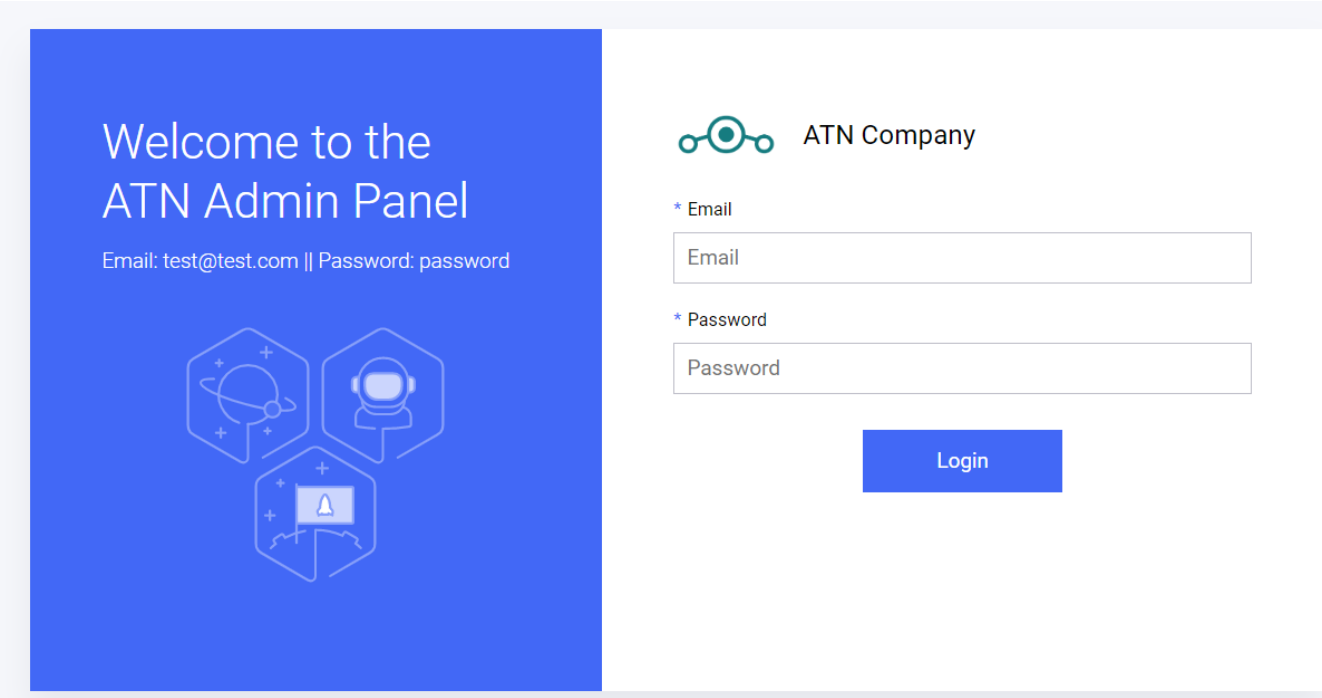
The ATN's Front-end will have 3 main pages such as **Login**, **main menu dashboard**, and **input(view) data**.

- **Login page:**

In ATN Admin Panel will require users need to login to the system with **email** and **password** that the administrator created before.

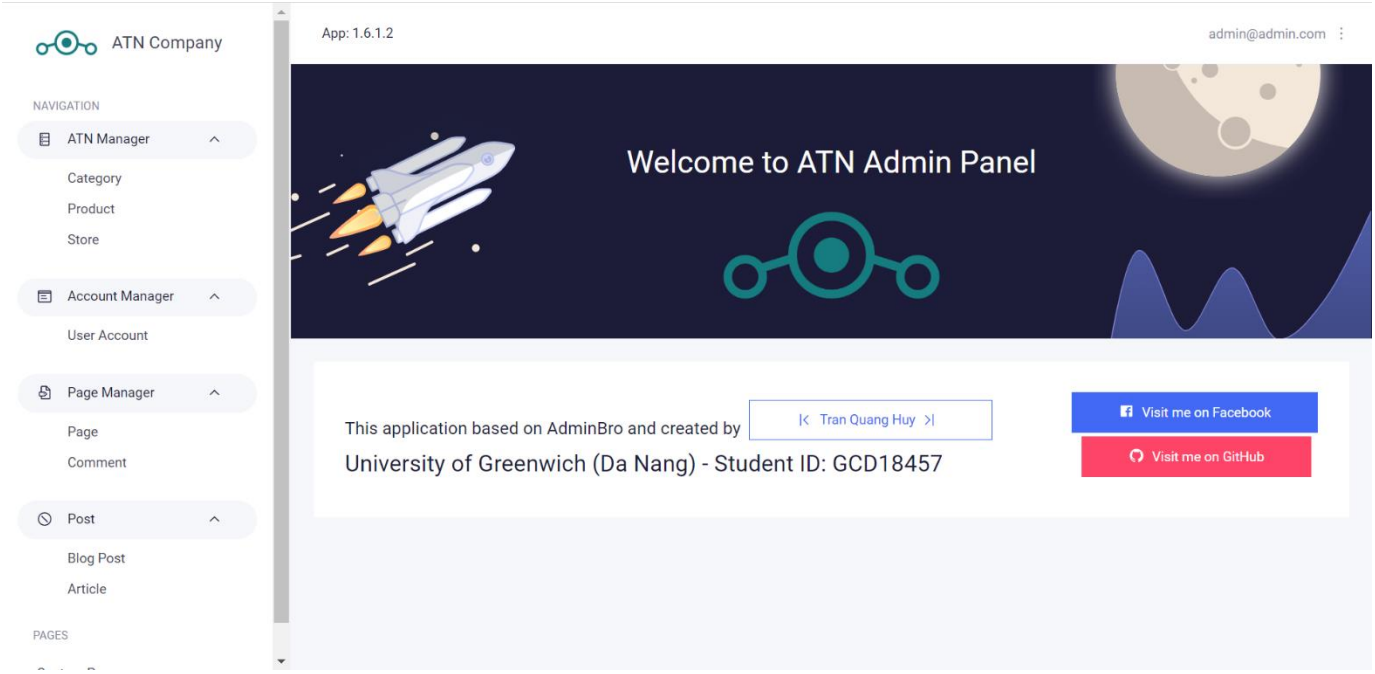
This Login Page will send data to the database to confirm that correct email and password. After that, if email and password are correct the login will be successful.

```
const router = AdminBroExpress.buildAuthenticatedRouter(adminBro, {
 authenticate: async (email, password) => {
  const user = await UserAccountModel.findOne({ email })
  if (user){
   if(password === user.auth.password)
    return user
  }
  return null
 },
 cookieName: 'adminATN',
 cookiePassword: 'LongPassword',
})
```



Picture 36. ATN login page

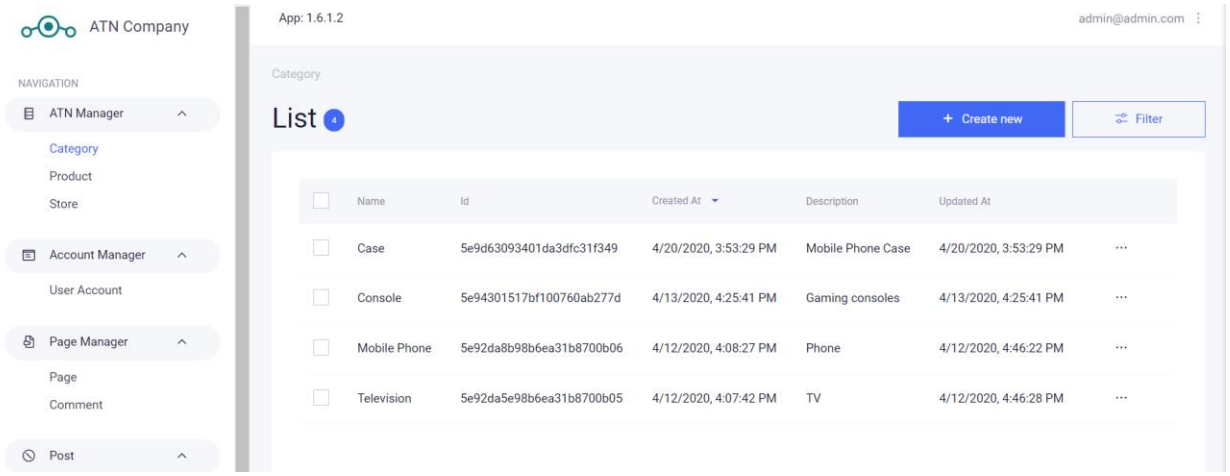
- Main menu and Dashboard:



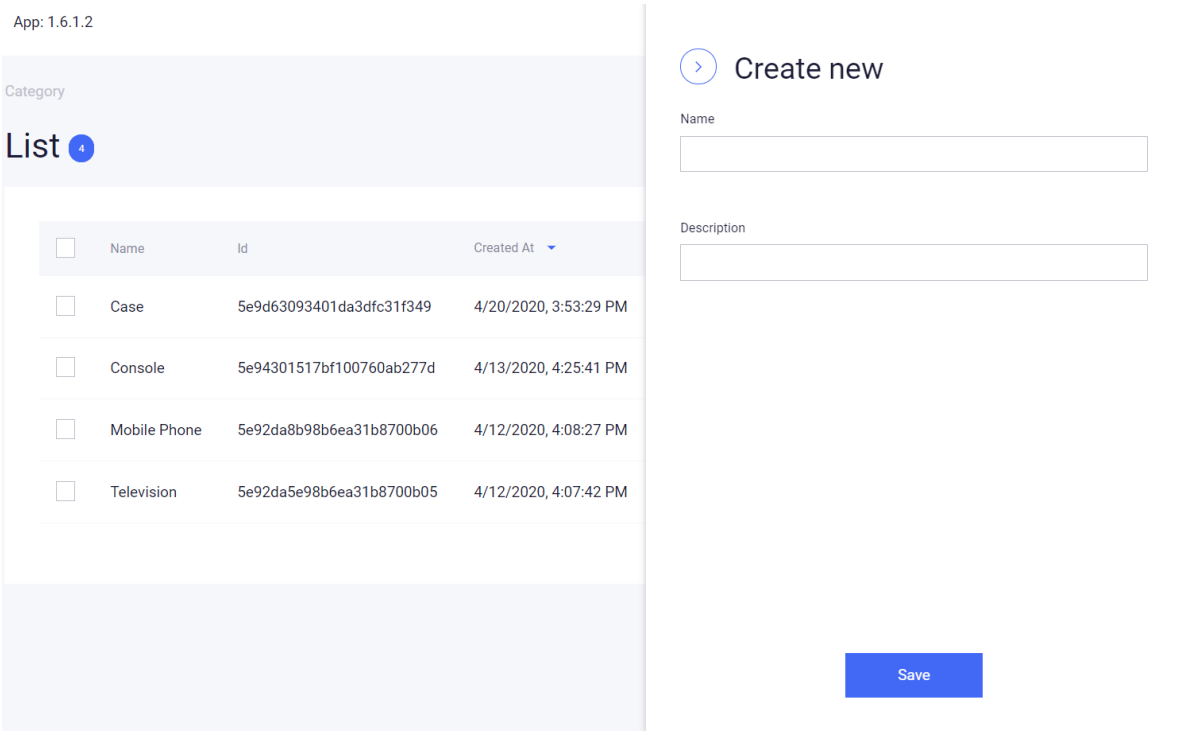
Picture 37. Main menu and dashboard

In main menu and dashboard, there are 2 main areas such as navigation bar and the dashboard

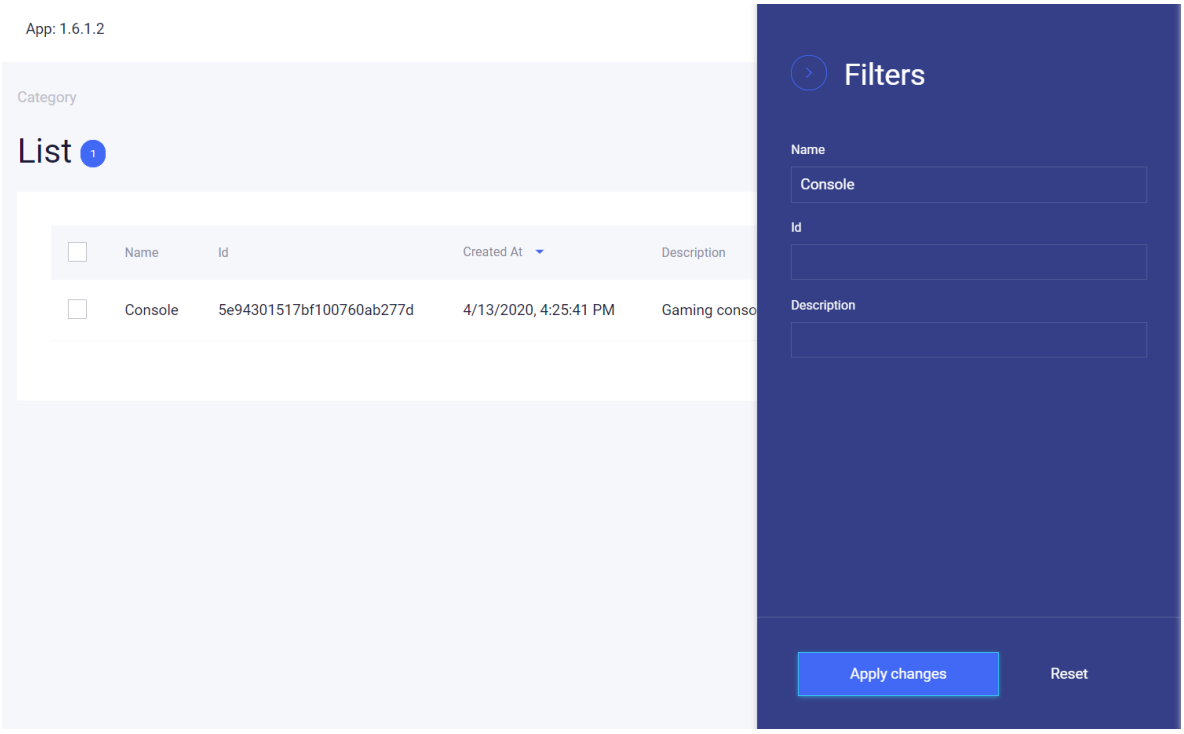
- **Navigation bar**: User can input or view information about each category, product, store
    - **Category:** To storage information about the category, the user can create or filter to find category.



Picture 38. ATN Admin Panel Category

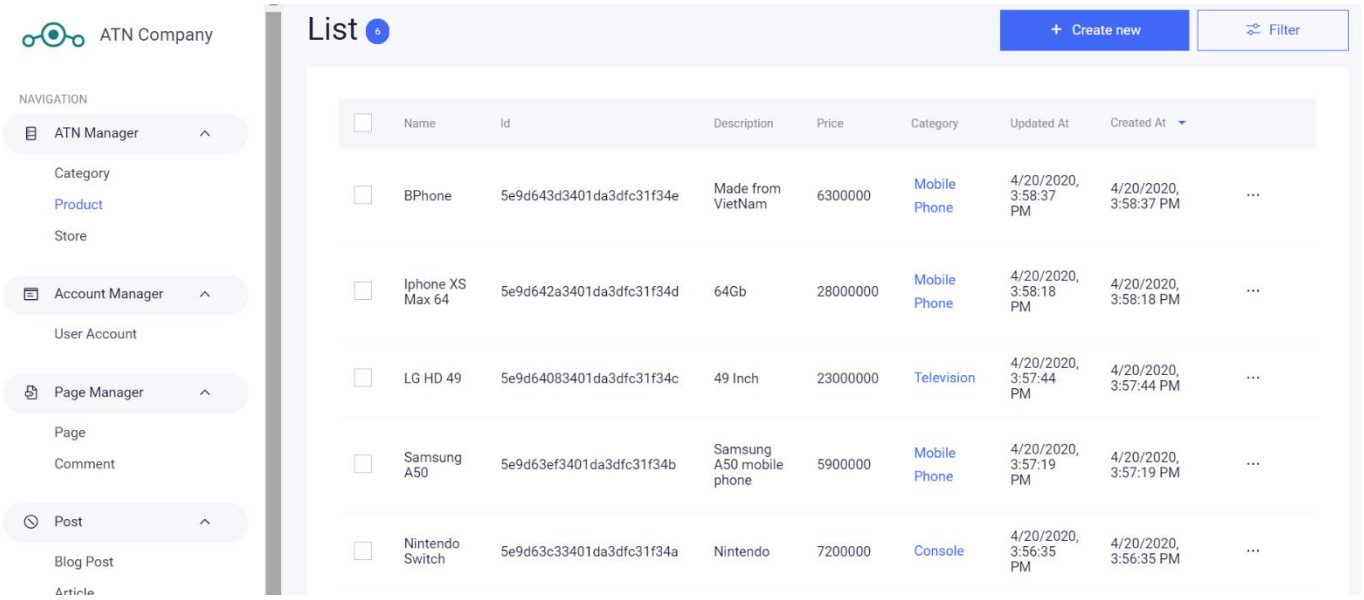


Picture 39. Create a new category

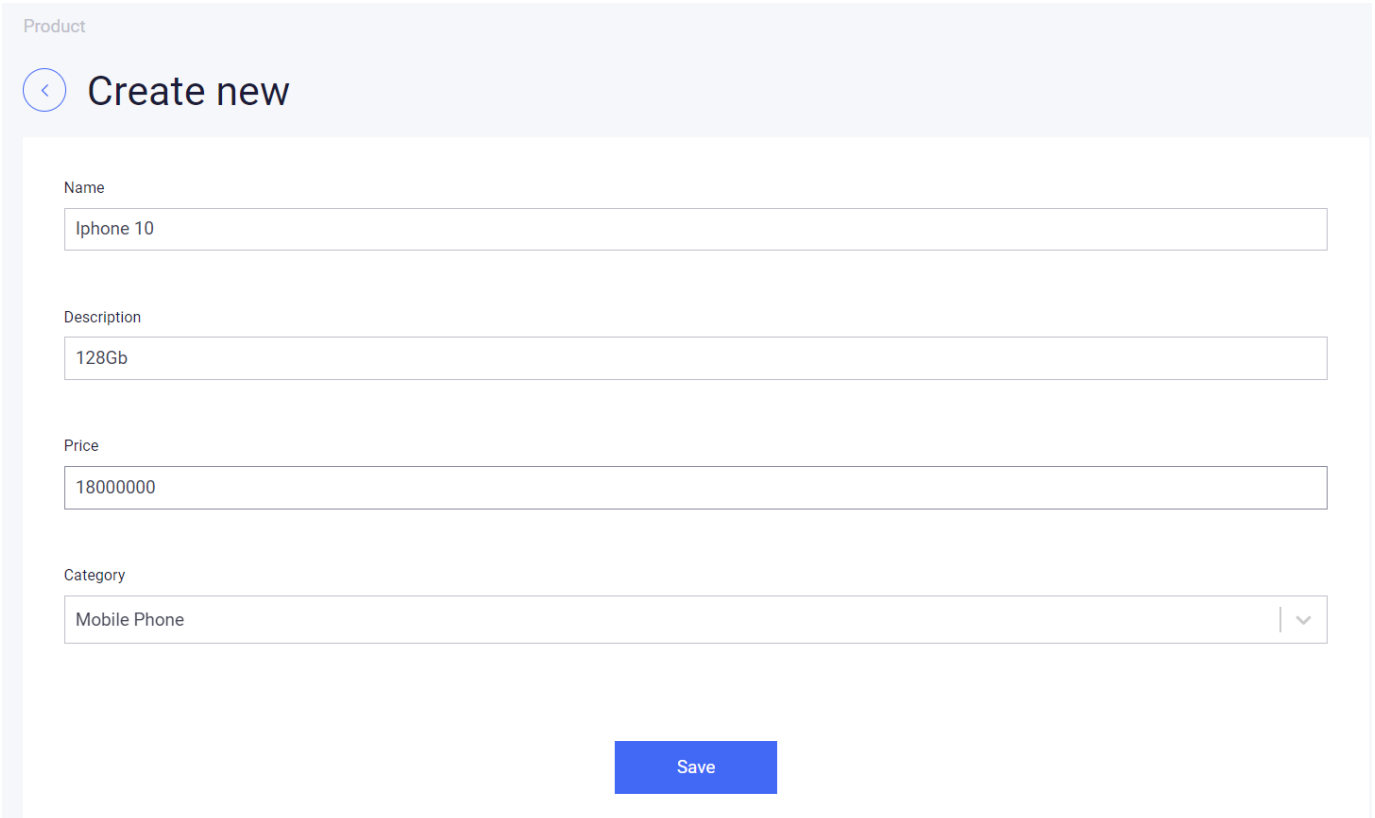


Picture 40. Filters in category

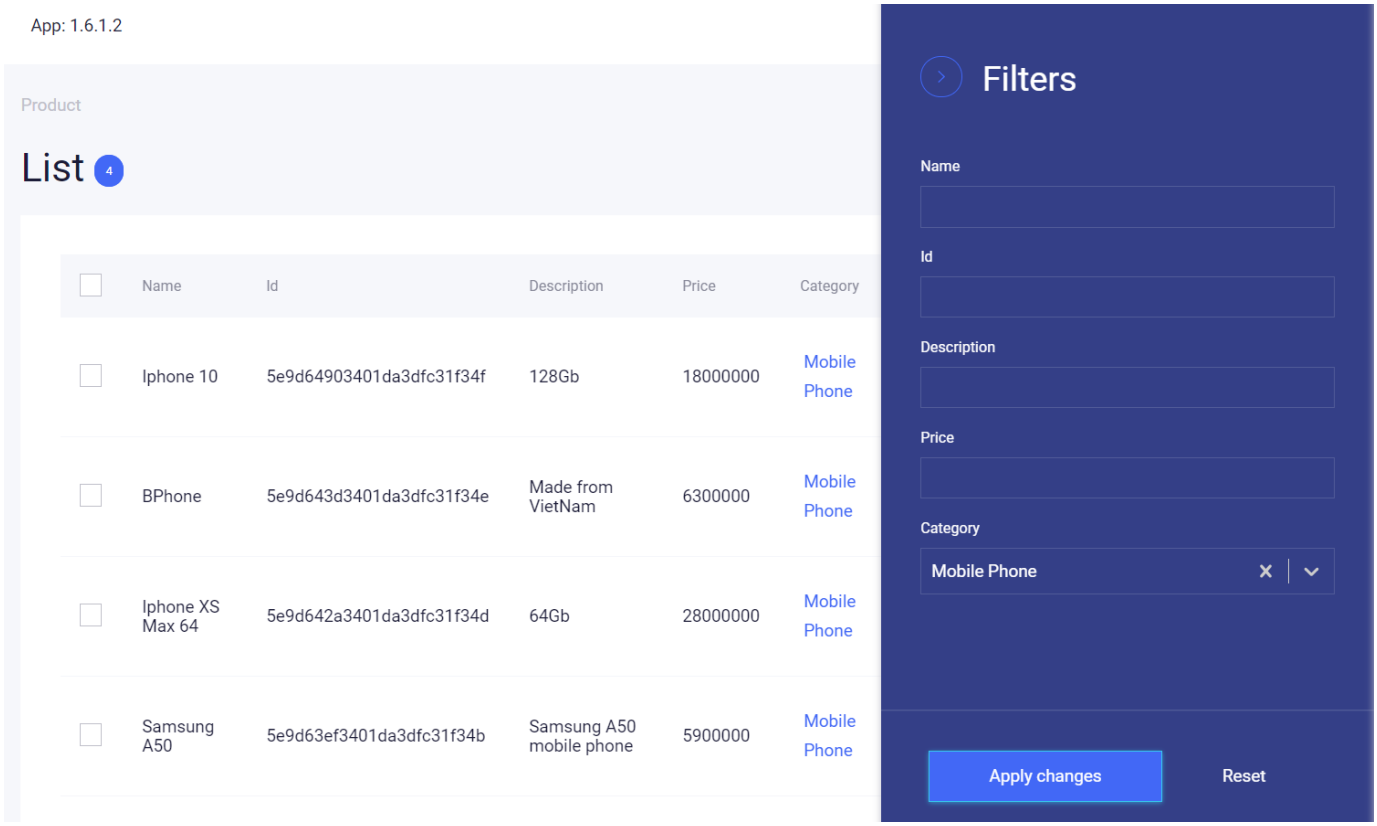
- **Product:** To storage information about products, users can create or filter to find categories.



Picture 41. ATN Admin Panel Product

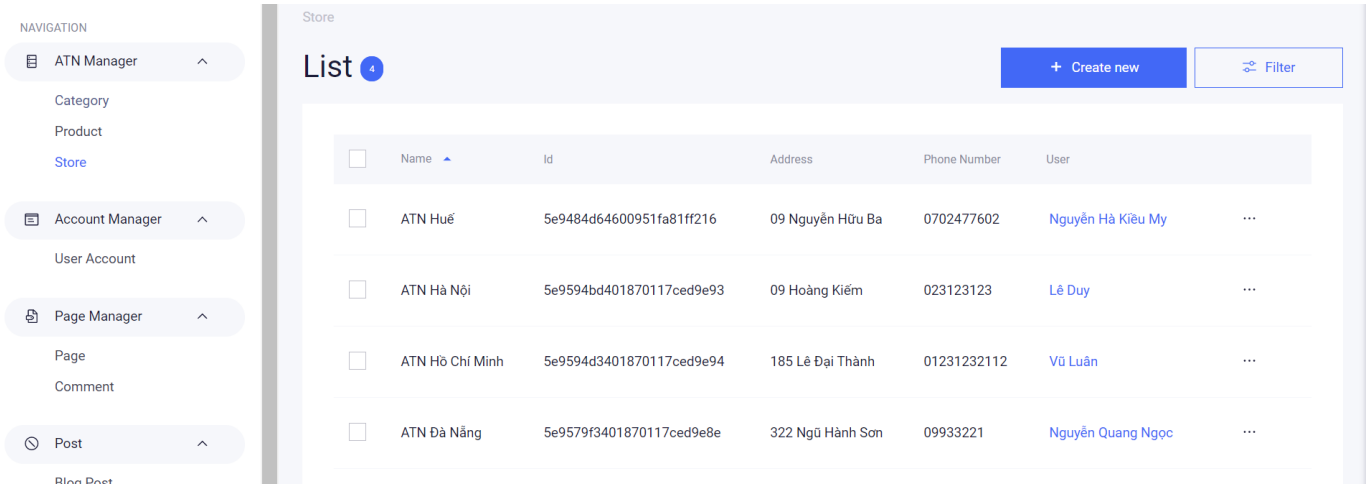


Picture 42. Create a new product

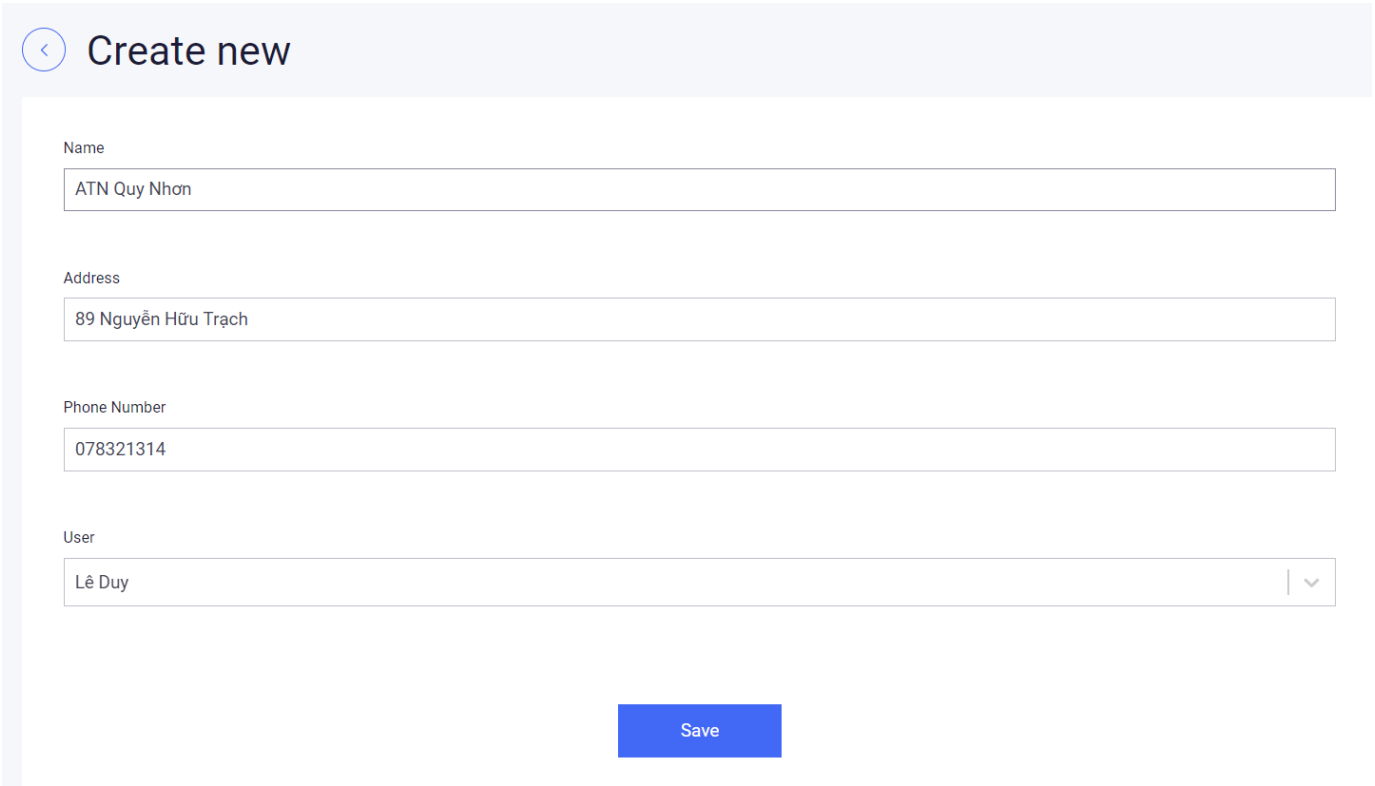


Picture 43. Filter in product

- **Store:** To storage information about the store, users can create or filter to find the store. Each store will have 1 Store manager.

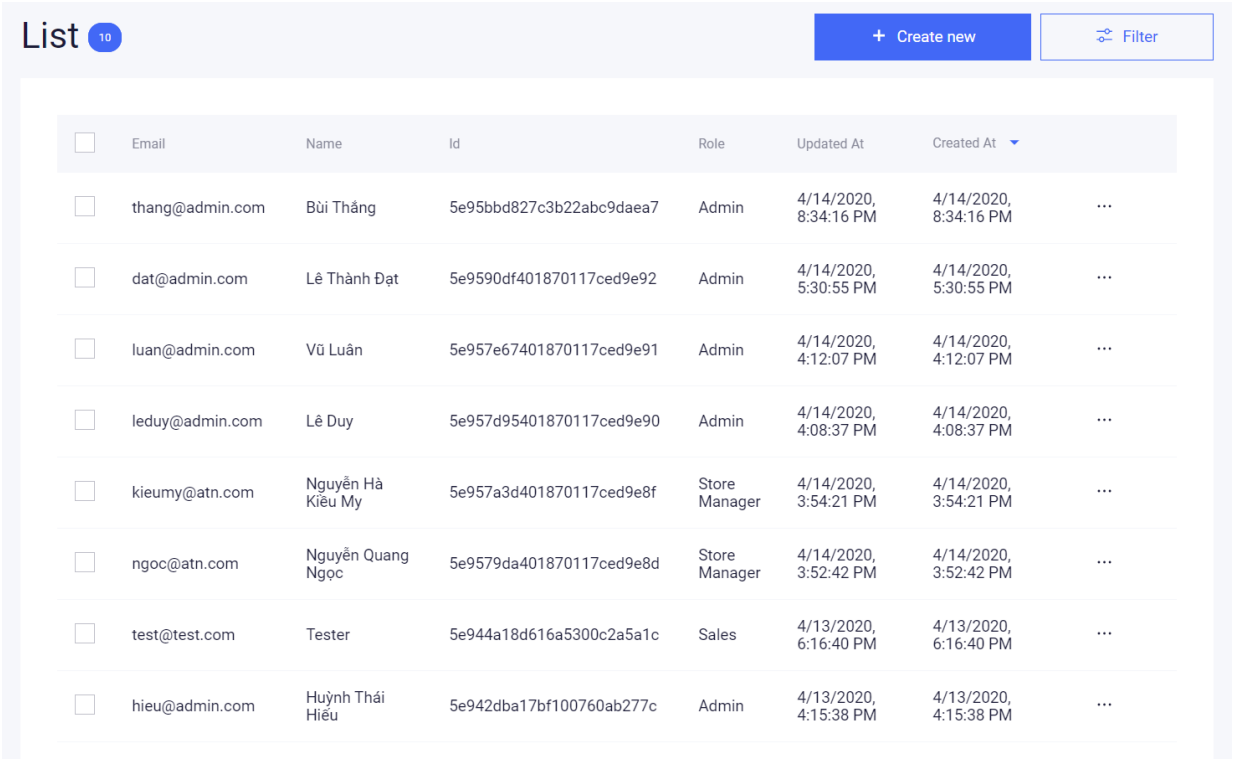


Picture 44. ATN Admin Panel Stores

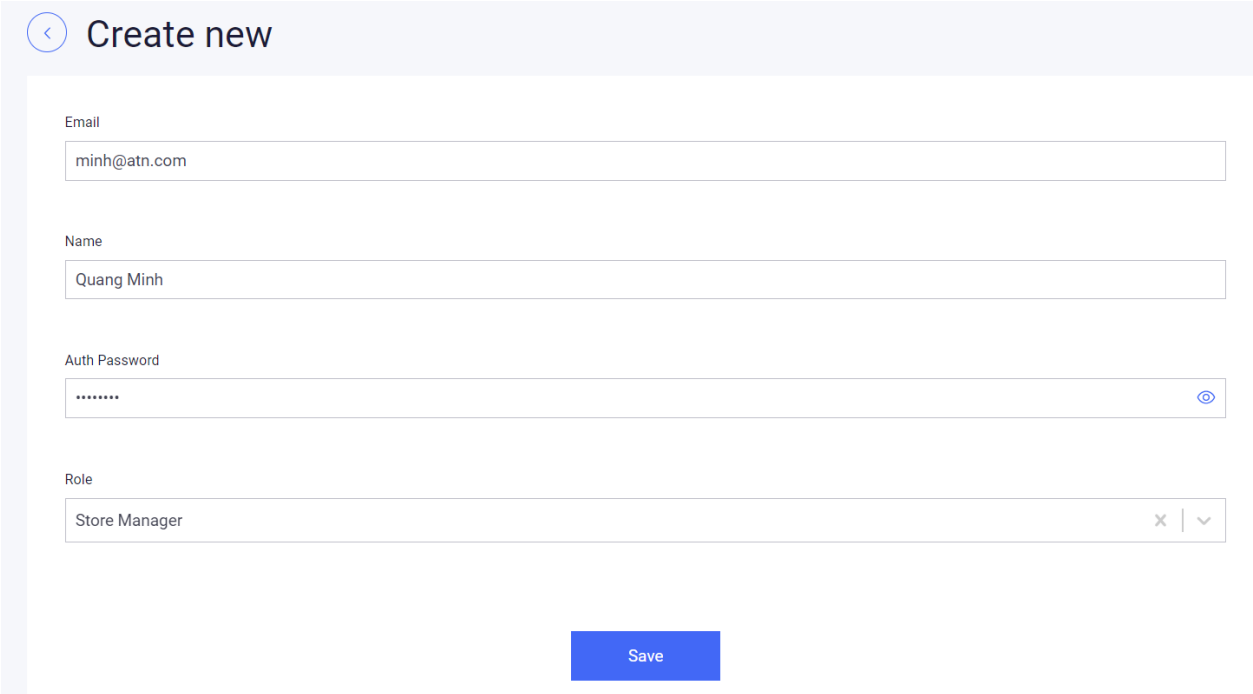


Picture 45. Create a new Store

- **User Account:** To storage information about a user account, users can create or filter to find a user account. Each User Account will be set with 1 role.

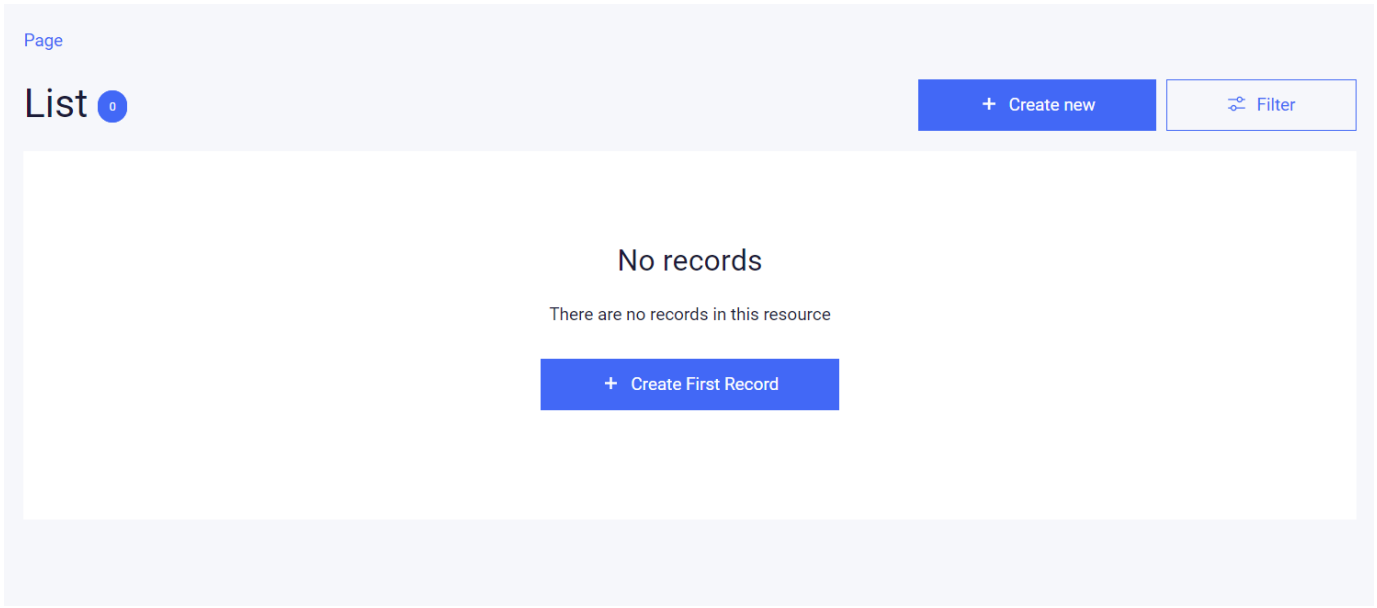


Picture 46. Admin Panel User Account



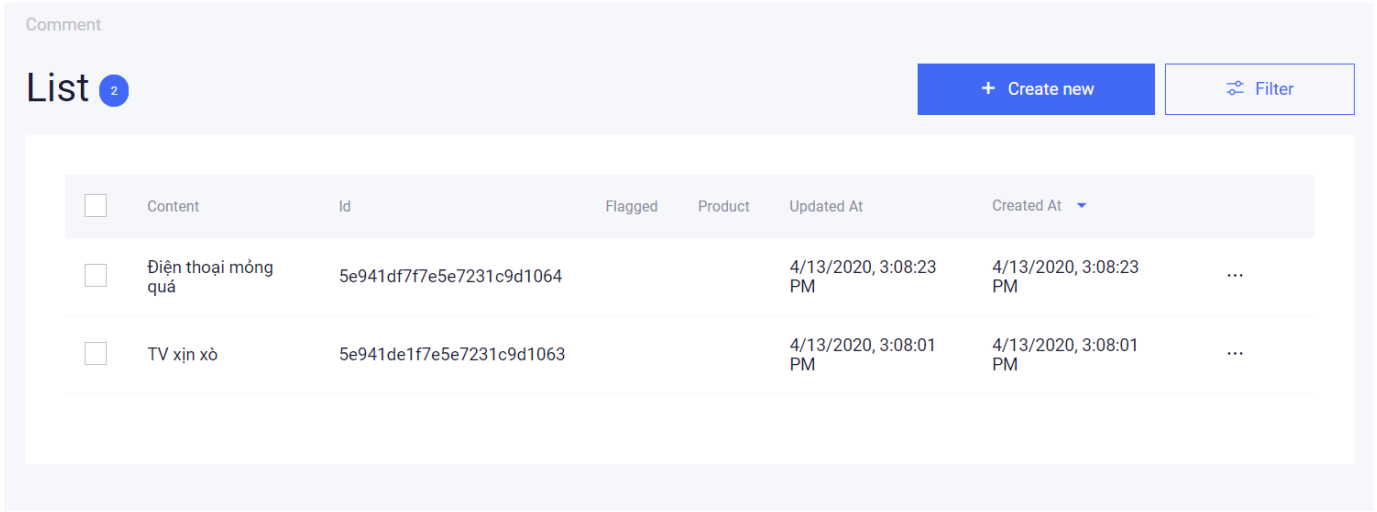
Picture 47. Create a new user account

- **Page:**



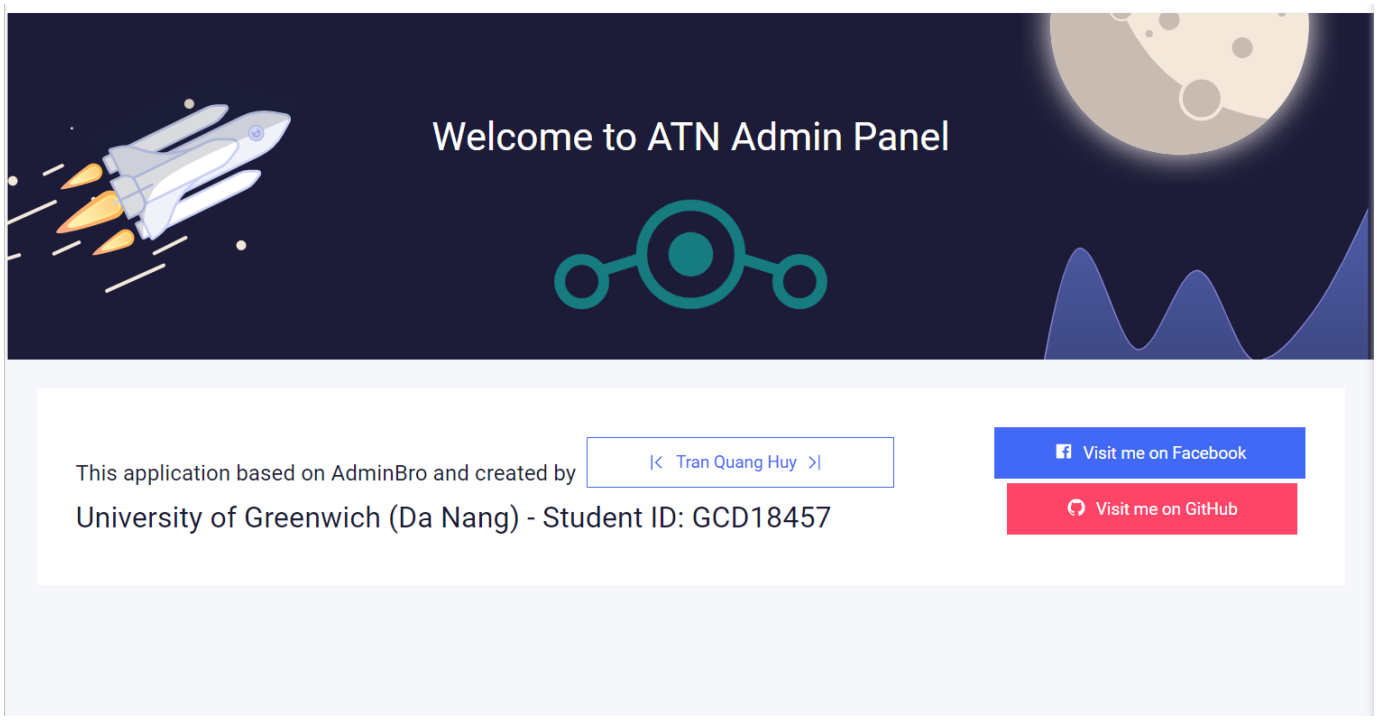
Picture 48. ATN Admin Panel Page

- **Comment:**

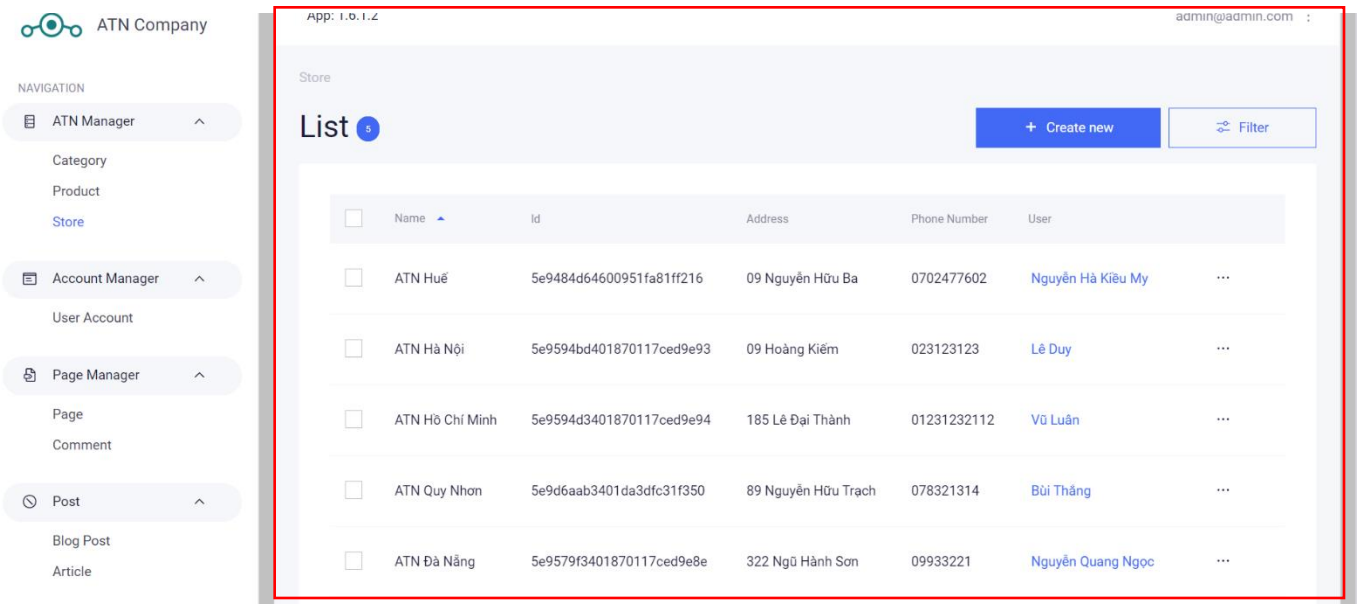


Picture 49. ATN Admin Panel Comment

- **Dashboard**: Show information about the developer and ATN Panel.



Picture 50. ATN Admin Panel Dashboard

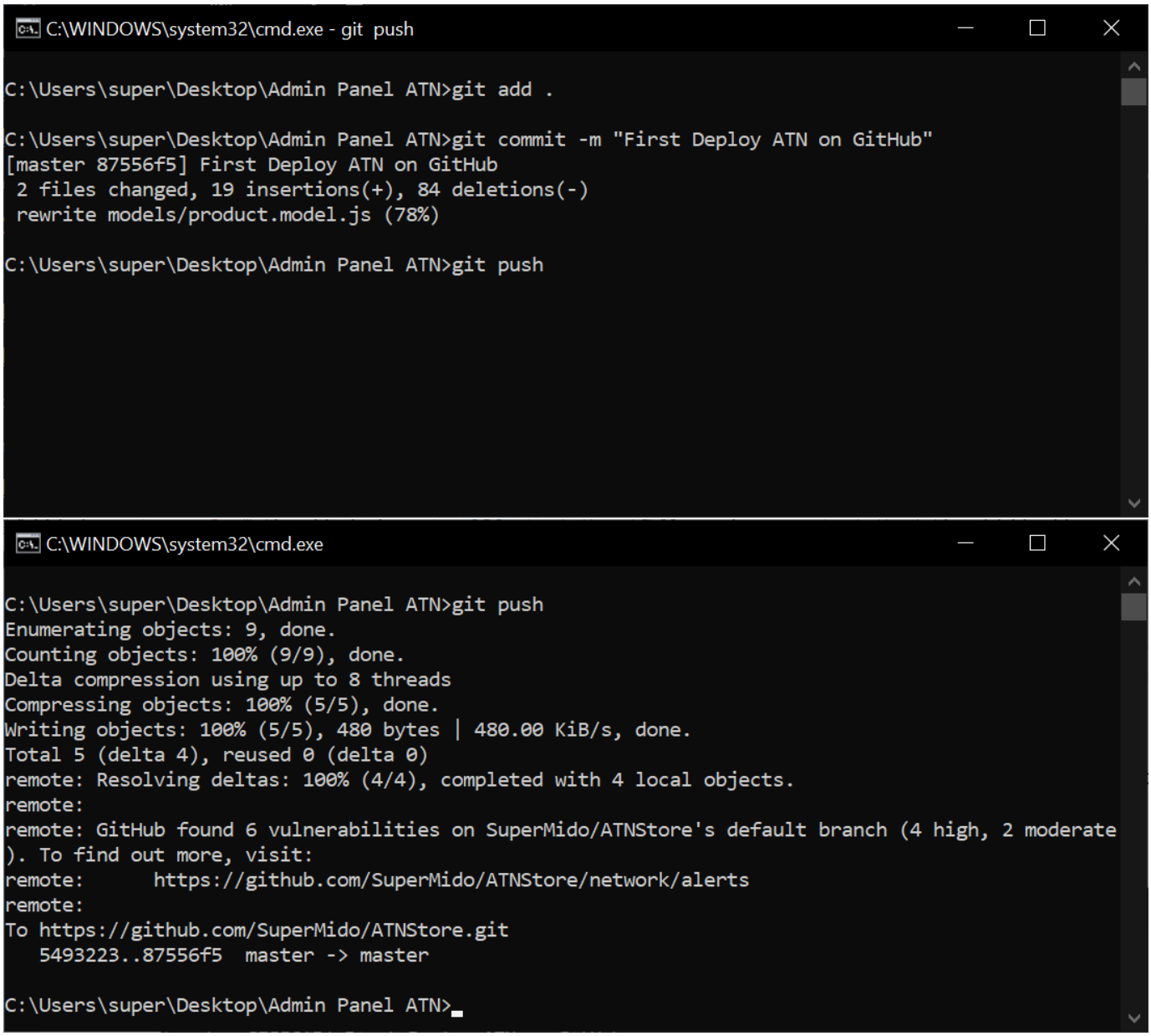


Picture 51. Dashboard in Navigation

2.3. Web application deploy on Heroku

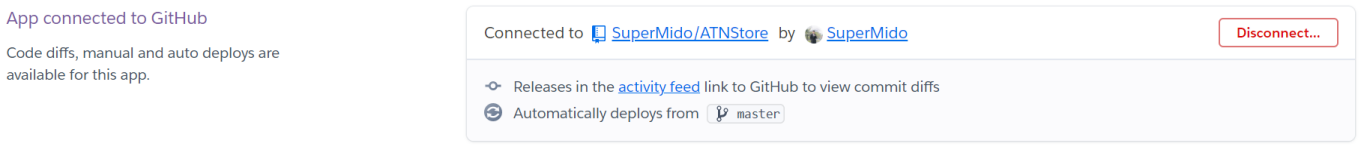
After the completed the development processes of the web front-end, back-end, and database infrastructure, it is time to push all the web components to Heroku.

First up is push the application into GitHub:

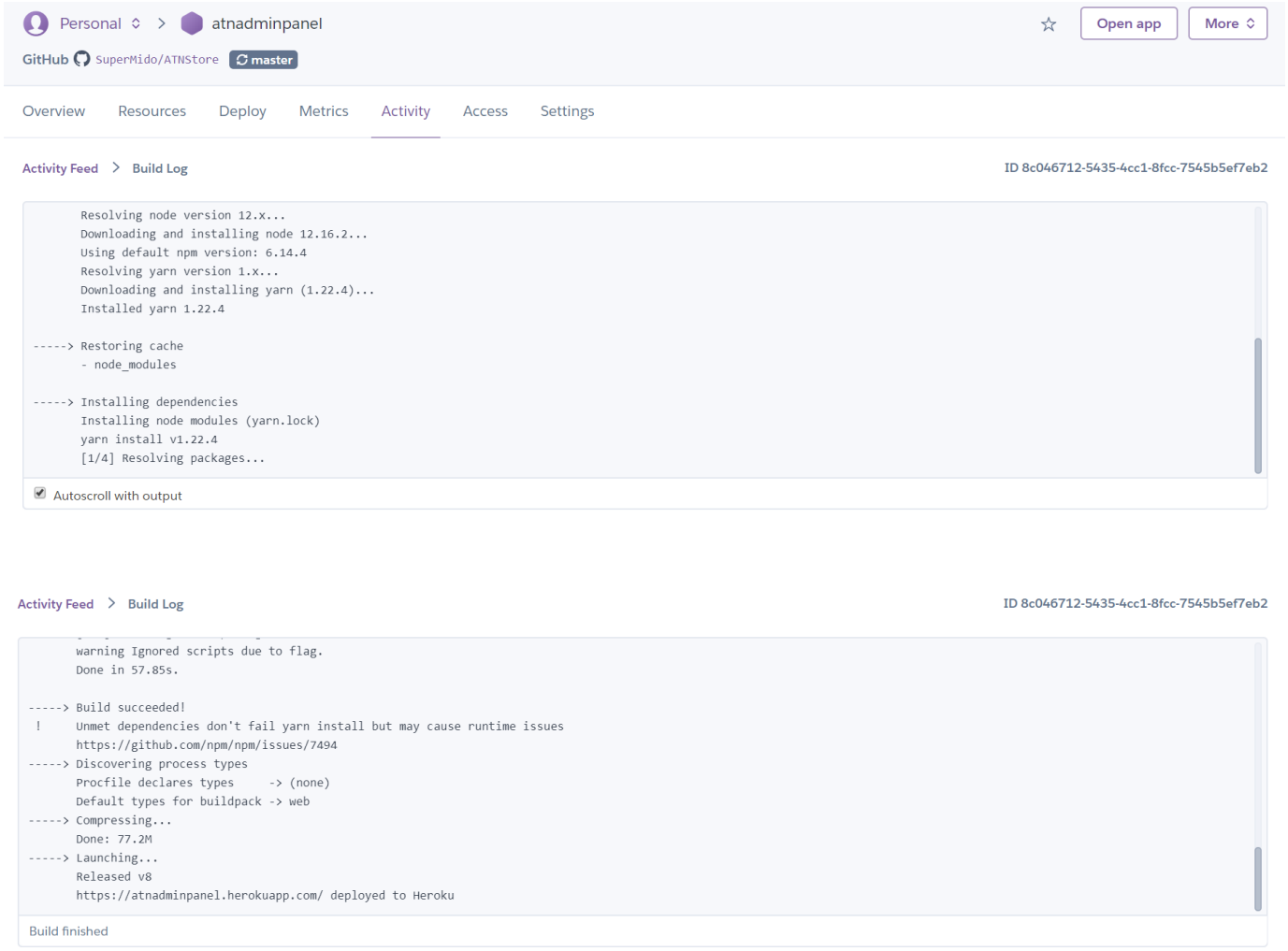


Picture 52. Push data into GitHub

In the Heroku configure, the repository of GitHub is connected to Heroku, so Heroku will automatically deploy from the **master** of GitHub

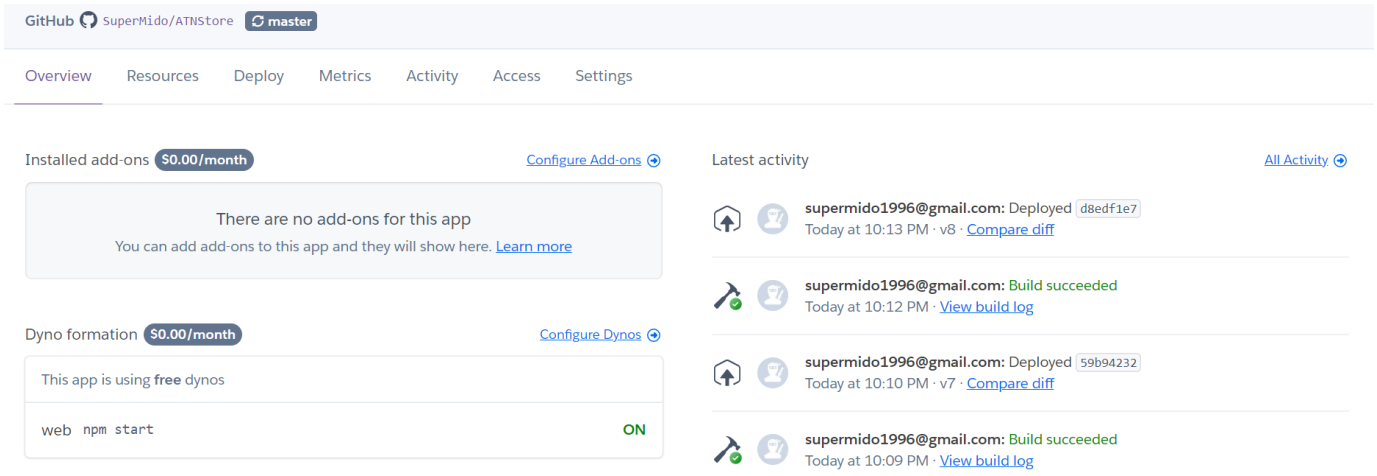


Picture 53. Configuration Heroku connected to GitHub



Picture 54. Heroku build an application with GitHub

The **Heroku** application log now got updated with the information that the web application got deployed successfully:

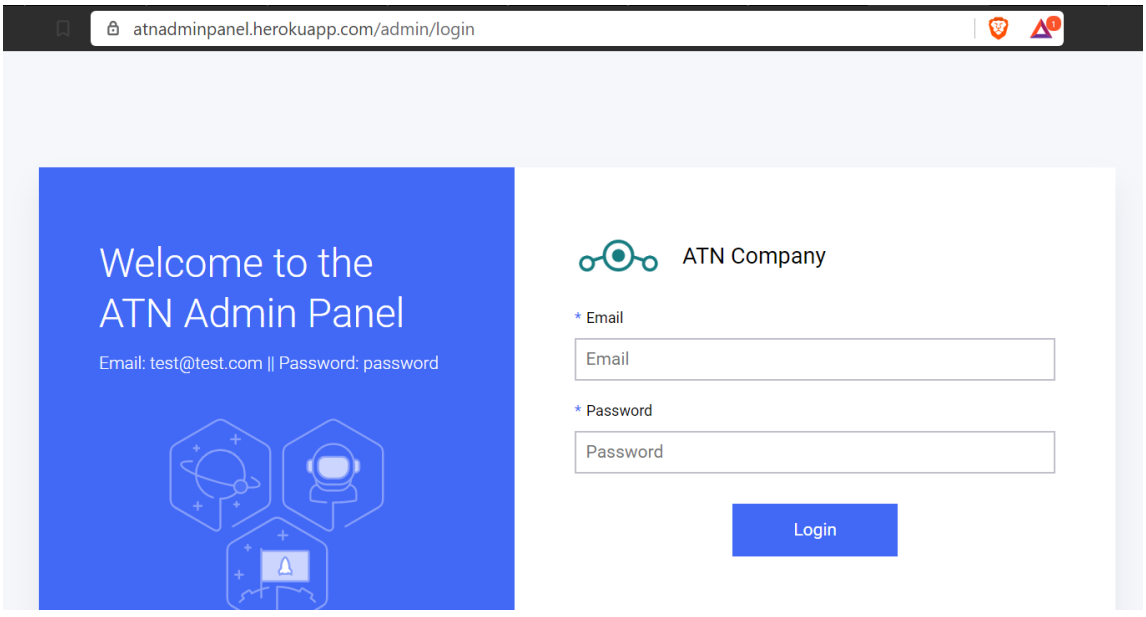


The app is now available at the address: https://atnadminpanel.herokuapp.com/

In order to proceed to internal pages, the login form will require a user account with the following details:

- Email: test@test.com
- Password: password

The source code is available at this GitHub repository: https://github.com/SuperMido/ATNStore



Picture 55. ATN Admin Panel on Heroku

3. The issues and constraints one can face during the development process and how to overcome. (M3-D2)

3.1. Back-End issues constraint

- **Connection to ClusterATN**:

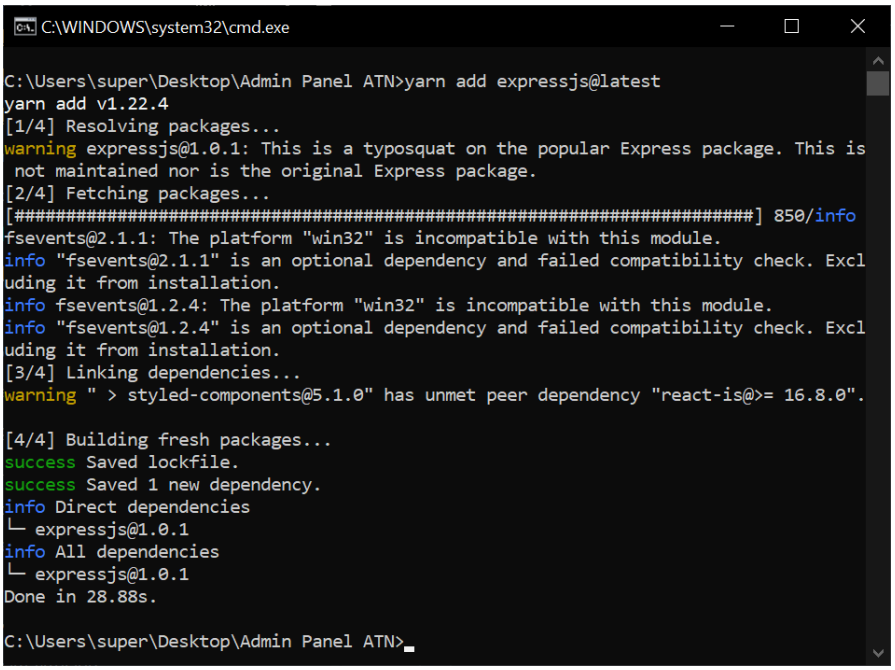
Base on the Node.Js version that the server-side will connect to MongoDB Atlas. In the first time implement application, the Node.Js is lower than 3.0, so MongoDB Atlas did not work.

In this case for ATN Admin Panel, the developer must select the right version of NodeJs so the database will work well.

Picture 56. Select version for Node.js

Solution for overcome connection fails with MongoDB Atlas:

Update latest Node.js version:



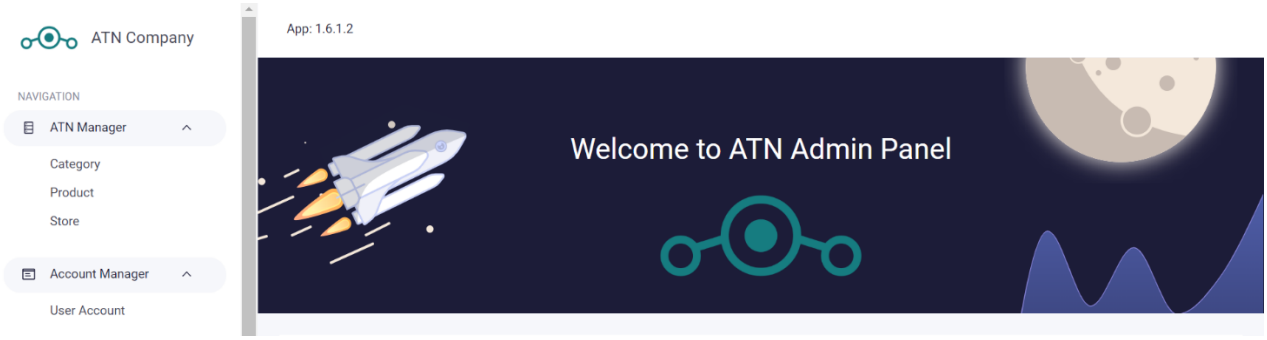
Picture 57. Update expressjs

Picture 58. Current version Node.js after update

- **Bypass login:**

As usual, users will have to log in to use the ATN Admin Panel system when opening the browser with the link https://atnadminpanel.herokuapp.com/.

In another way user change application path is https://atnadminpanel.herokuapp.com/admin, with this path, the user does not need to login but still can use the ATN Admin Panel system



Picture 59. Bypass login

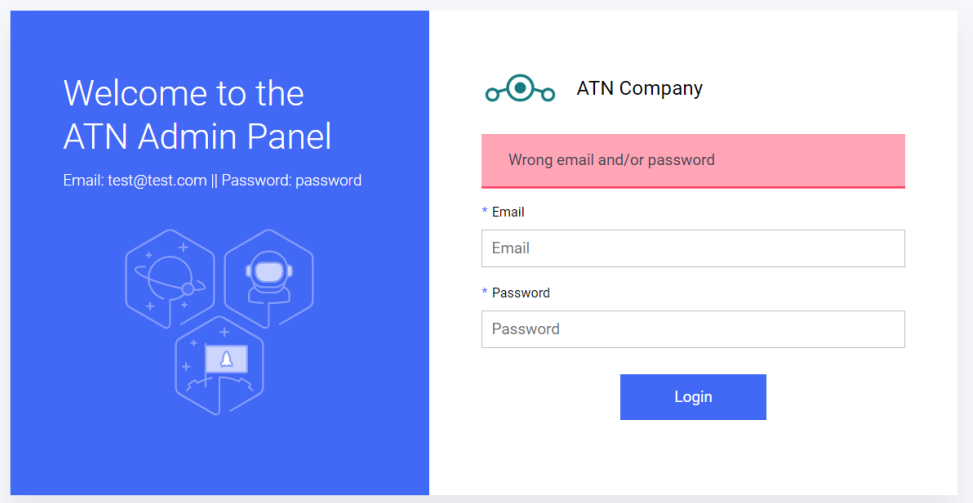
With this bypass, the system will not recognize which user is using the application and make a serious problem to the system.

Solution for overcome Bypass login:

Setup router for server-side:

```
const router = AdminBroExpress.buildAuthenticatedRouter(adminBro, {
  authenticate: async (email, password) => {
    const user = await UserAccountModel.findOne({ email })
    if (user){
      if(password === user.auth.password)
        return user
    }
    return null
  },
  cookieName: 'adminATN',
  cookiePassword: 'LongPassword',
})
```

The application will confirm that email and password are correct then login will be successful. If the login fails, the ATN Admin Panel will return to the login page and require the user to log in again.



Picture 60. Login fail

With configuring the rootPath the Admin Panel will redirect to the login page:

```
module.exports = {
  ...
  rootPath: '/admin',
}
```

```
app.use(adminBro.options.rootPath, router)
app.get('/', router)
```

By using https://atnadminpanel.herokuapp.com/admin, now the ATN Admin Panel does not allow the user to use the system without login and redirect to the login page

Picture 61. Redirect to login page

- **Require login every time**

Whenever users using ATN Admin Panel, the application will require the user to login. But after login, any action of the user such as refresh browser, close the application or new tab on a browser will meet function login again.

Solution for overcome require login every time:

Using packet express-session:

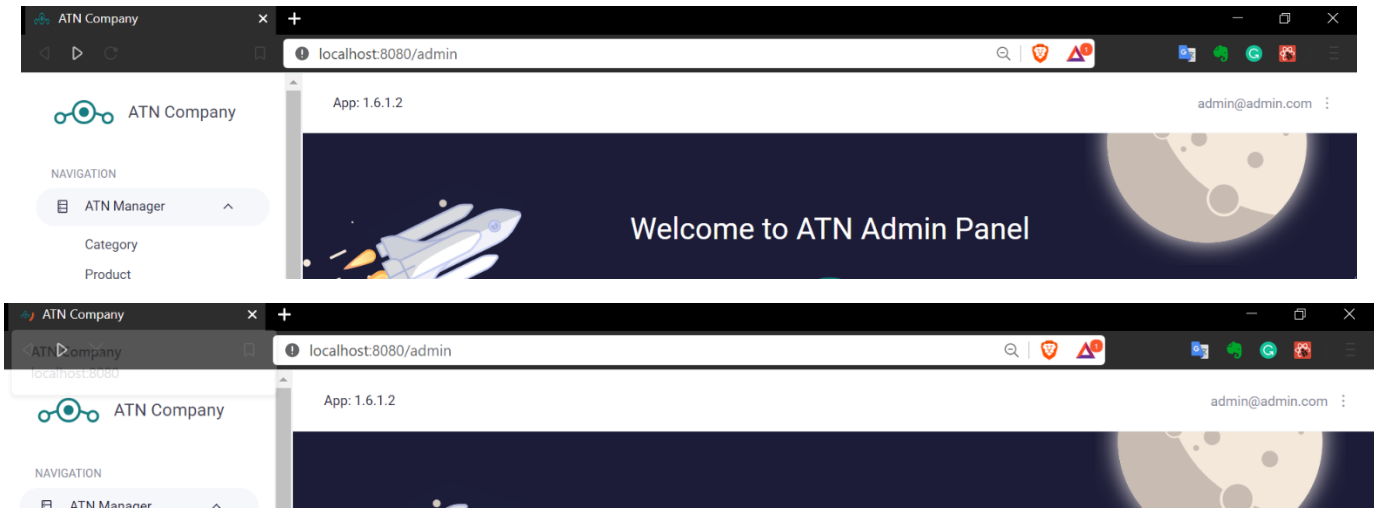
By default Express requests are sequential and no request can be linked to each other. There is no way to know if this request comes from a client that already performed a request previously.

Users cannot be identified unless using some kind of mechanism that makes it possible.

When implemented, every user of API or website will be assigned a unique session, and this allows the application to store the user state.

It can store session data in:

- **Memory**
- **A database:** MongoDB
- **A memory cache:** Memcached



Picture 62. Refresh page still in Panel

```
const buildAuthenticatedRouter = (
  admin,  auth,  predefinedRouter,  sessionOptions = {},  formidableOptions = {},
) => {
  const router = predefinedRouter || express.Router()
  router.use(session({
    ...sessionOptions,
    secret: auth.cookiePassword,
    name: auth.cookieName,
  }))
  router.use(formidableMiddleware(formidableOptions))
  const { rootPath } = admin.options
  let { loginPath, logoutPath } = admin.options
  loginPath = loginPath.replace(rootPath, '')
  logoutPath = logoutPath.replace(rootPath, '')

  router.get(loginPath, async (req, res) => {
    const login = await admin.renderLogin({
      action: admin.options.loginPath,
      errorMessage: null,
    })
    res.send(login)
  })
  router.post(loginPath, async (req, res, next) => {
    const { email, password } = req.fields
    const adminUser = await auth.authenticate(email, password)
    if (adminUser) {
      req.session.adminUser = adminUser
      req.session.save((err) => {
        if (err) {
          next(err)
        }
        res.redirect(rootPath)
      })
    } else {
      const login = await admin.renderLogin({
        action: admin.options.loginPath,
        errorMessage: 'invalidCredentials',
      })
      res.send(login)
    }
  })
  router.use((req, res, next) => {
    if (AdminBro.Router.assets.find(asset => req.originalUrl.match(asset.path))) {
      next()
    } else if (req.session.adminUser) {
      next()
    } else {
      res.redirect(admin.options.loginPath)
    }
  })
  router.get(logoutPath, async (req, res) => {
    req.session.destroy(() => {
      res.redirect(admin.options.loginPath)
    })
  })

  return buildRouter(admin, router, formidableOptions)
}
```

3.2. Front-End issues constraint

- **Full permission with all role user:**

There are 3 main roles in the ATN Admin Panel such as Admin, Store Manager, and sales. Each role has different permission to use Admin Panel, but when implementing this Front-End there are some critical issues:

  - Every role can create an account for the user such as Admin, Store Manager and sales
  - Every role can monitor another store
  - Every role can create a product, category or store

Solution for overcome full permission with all role user:

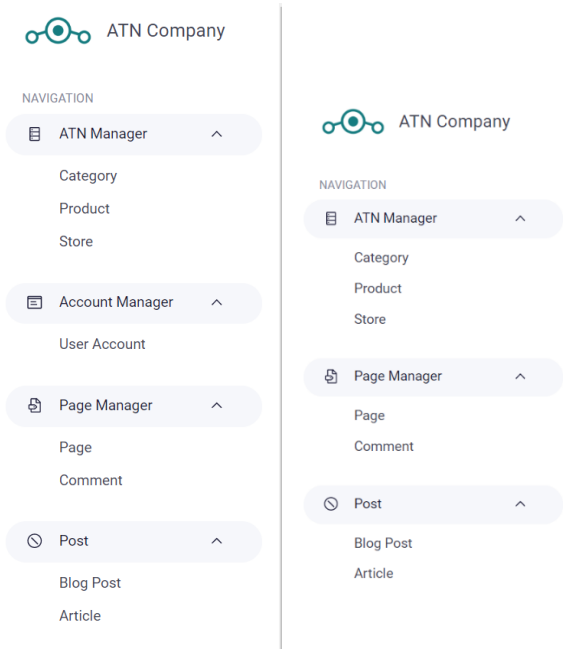
Configure permission to create a user account:

```
const AdminBro = require('admin-bro')
const { sort, timestamps } = require('./sort')

module.exports = {
...
  list: { isAccessible: ({ currentAdmin }) => currentAdmin && currentAdmin.role === 'Admin', showInDrawer: true, },
  show: { isAccessible: ({ currentAdmin }) => currentAdmin && currentAdmin.role === 'Admin', showInDrawer: true, },
  edit: { isAccessible: ({ currentAdmin }) => currentAdmin && currentAdmin.role === 'Admin', showInDrawer: true, },
  delete: { isAccessible: ({ currentAdmin }) => currentAdmin && currentAdmin.role === 'Admin', showInDrawer: true, },
  new: { isAccessible: ({ currentAdmin }) => currentAdmin && currentAdmin.role === 'Admin', showInDrawer: true, },
 }
}
```

With this config, only Admin role can view, create or modify user account

View of Admin account and another account:

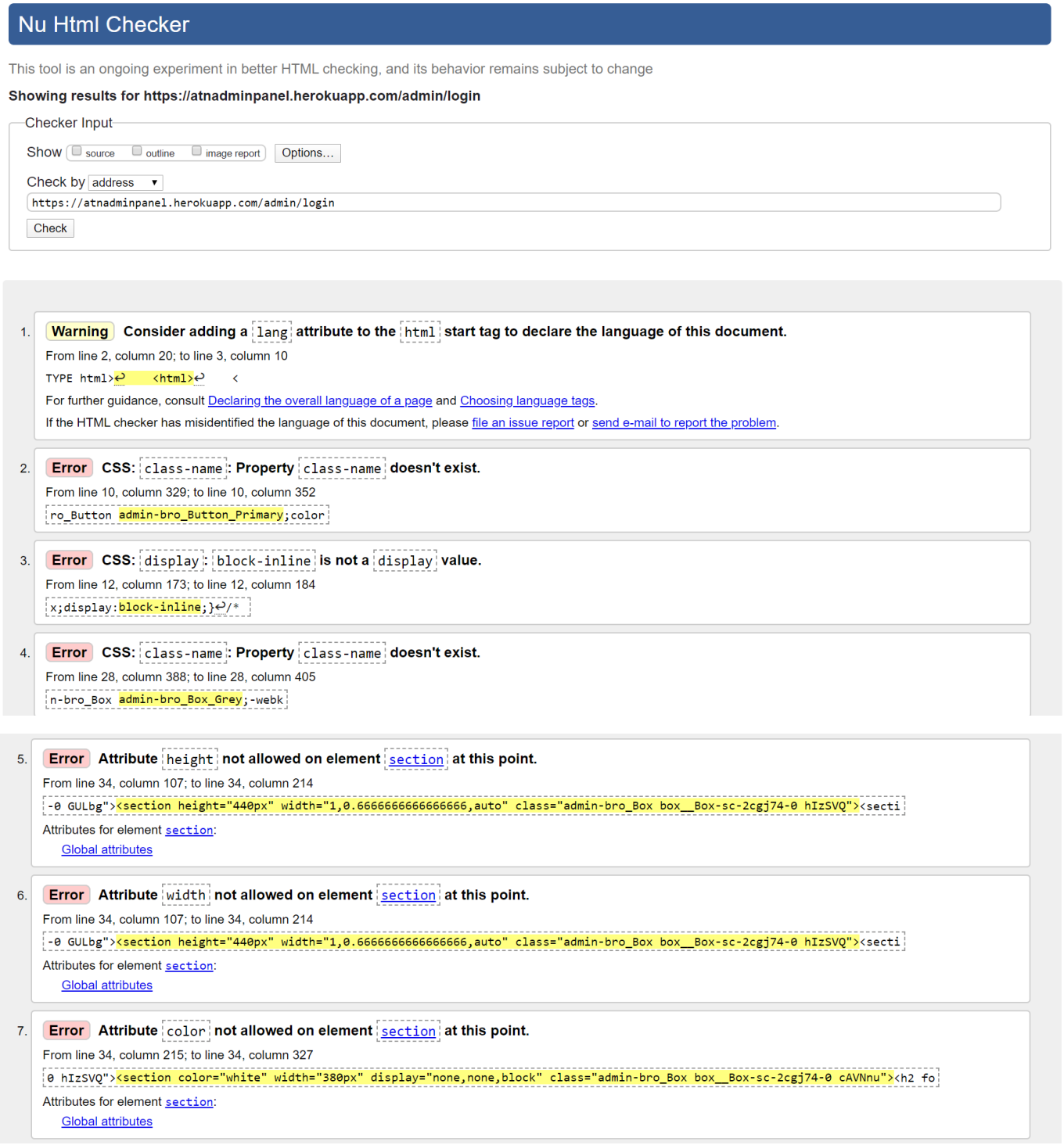


Picture 63. View of Admin account and another account

- **HTML Problem:**

In ATN Admin Panel, there is a lot of error attribute, element, that will cause the system heavy and not stable

From Nu Checker, it shows errors and warnings for https://atnadminpanel.herokuapp.com/



Picture 64. HTML Checker

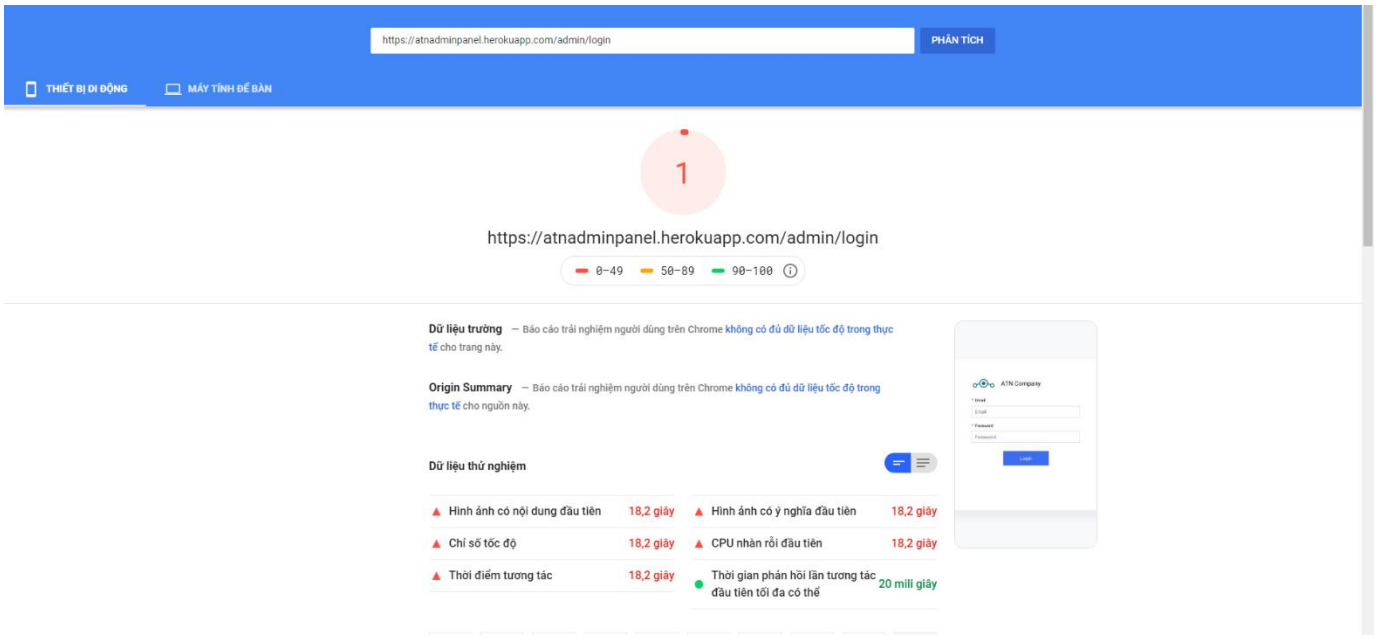
Solution for overcome HTML problem:

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change.

By the way, check errors and warnings form:
https://validator.w3.org/nu/?doc=https://atnadminpanel.herokuapp.com/

The developer can modify or delete the unused attribute, property or element that unnecessary.

3.3. Server loading

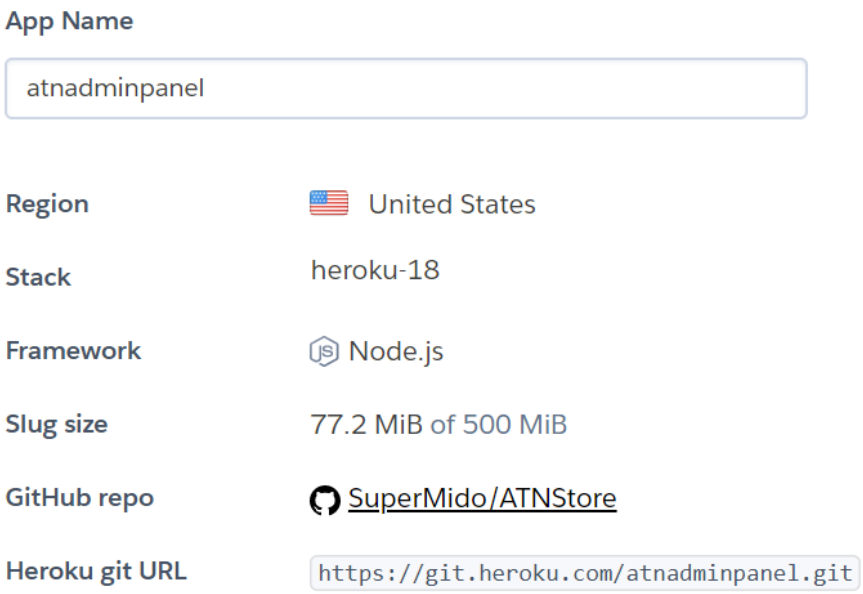


Picture 65. Checking Server Loading

The application loading so slow, it's cost too much time to load everything from the server. Then the first open application will slow, but the next time will be faster.

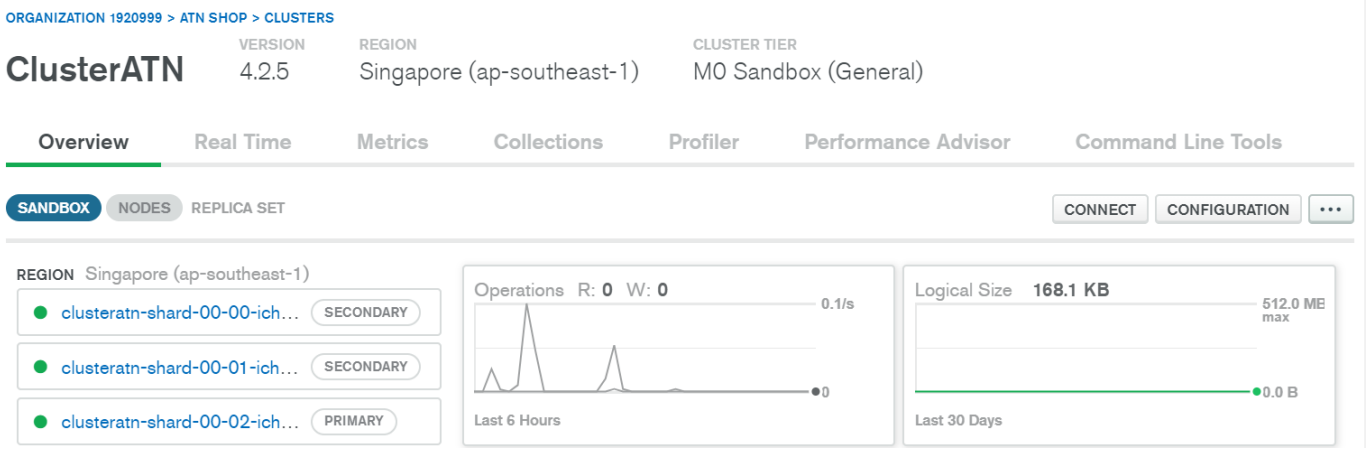
Solution for overcome server loading problem:

The application deploys on Heroku with region United States should be a move to another nearly Viet Nam



Picture 66. Region Heroku configuration

By the way, ATN Company will be using AWS (Amazon Web Service) and relocated to Singapore. When relocated to the server in Singapore, it is the same server with MongoDB Atlas Database.



Picture 67. Region MongoDB Atlas configuration

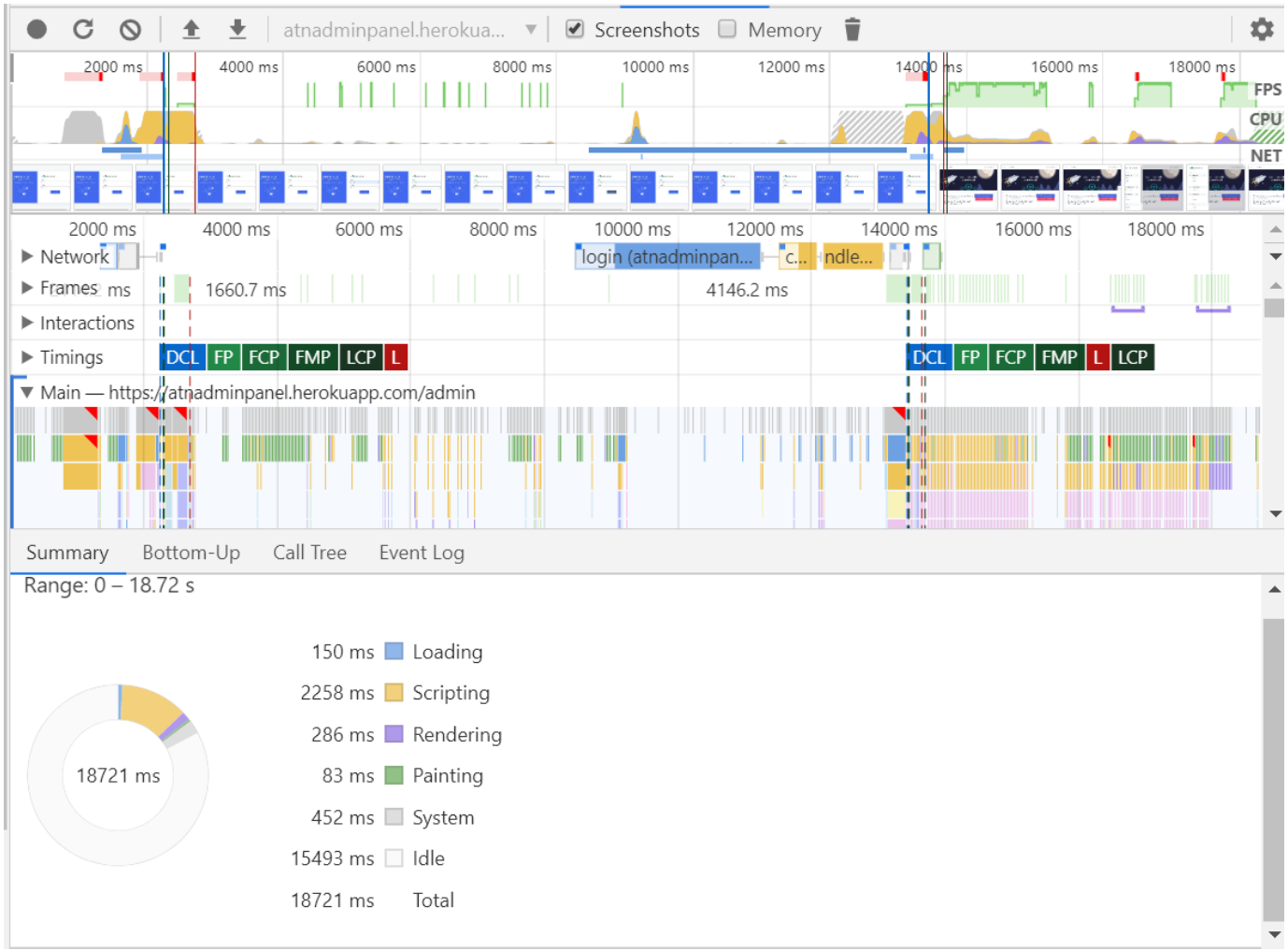
After that, the application will be loading faster from the server.

3.4. Performance

Web performance refers to the speed in which web pages are downloaded and displayed on the user's web browser. Web performance optimization (WPO), or website optimization is the field of knowledge about increasing web performance.

Faster website download speeds have been shown to increase visitor retention and loyalty[1][2] and user satisfaction, especially for users with slow internet connections and those on mobile devices.[3] Web performance also leads to fewer data traveling across the web, which in turn lowers a website's power consumption and environmental impact. (wikipedia, n.d.)

Performance of ATN Admin Panel is not good enough, there is too much time application Idle.



Picture 68. Performance from login to dashboard Admin Panel

Solution for overcome Performance problem:

Async / Await is a feature of JavaScript that helps us to work with asynchronous functions in a more interesting and easier to understand way. It is built on Promises and is compatible with all API-based Promises. Inside:

- **Async** - declares an asynchronous function
  - Automatically convert a regular function into a Promise.
  - When it calls the async function it will handle everything and return the result in its function.
  - Async allows using Await.
- **Await** - pause the execution of async functions.
  - When placed in front of a Promise, it will wait until the Promise ends and return the result.
  - Await only works with Promises, it does not work with callbacks.
  - Await can only be used inside async functions.

Using async/await to connect database:

```
const run = async () => {
  await mongoose.connect(process.env.MONGO_URL, {
    useNewUrlParser: true,
    useUnifiedTopology: true
  })
  app.listen(process.env.PORT, () => console.log(` Admin Panel is under localhost:${process.env.PORT}` ))
}
```

Using async/await to login into system:

```
const router = AdminBroExpress.buildAuthenticatedRouter(adminBro, {
  authenticate: async (email, password) => {
    const user = await UserAccountModel.findOne({ email })
    if (user){
      if(password === user.auth.password)
        return user
    }
    return null
  },
  cookieName: 'adminATN',
  cookiePassword: 'LongPassword',
})
```

Analyzing the technical challenges for cloud applications and assess their risks

1. The most common problems which arise in a Cloud Computing platform and discuss appropriate solutions to the problems (P7)

- **Data Security concern**

When talking about the security concern of cloud technology, then a lot of questions remain unanswered. Multiple serious threats like virus attack and hacking of the client's site are the biggest cloud computing data security issues. Entrepreneurs have to think about these issues before adopting cloud computing technology for their business. Since the user is transferring the company's important details to a third party so it is important to ensure ATN company about the manageability and security system of the cloud.

A lot of questions remain unanswered when describing the security concern of cloud technology. Multiple threats like virus attack and hacking of the client's site are the biggest cloud computing data security issues

- **Selecting the perfect cloud set-up**

Choosing the appropriate cloud mechanism as per the needs of the business is very necessary. There are three types of cloud configuration such as public, private, and hybrid. The main secret behind the successful implementation of the cloud is picking up the right cloud. If ATN company are not selecting the right cloud then maybe ATN company usually use public clouds. A few companies like to go for a balanced approach with hybrid clouds. Choose a cloud computing consulting service that is aware and disclose the terms and conditions regarding cloud implementation and data security. (educba, n.d.)

- **Real-time monitoring requirements**

It is required to monitor its system in real-time. It is a compulsory term for their business that they continuously monitor and maintain their inventory system. Banks and some government agencies need to update their system in real-time but cloud service providers are unable to match this requirement. This is a big challenge for cloud services providers.

- **Dependency on service providers**

For uninterrupted services and proper working, ATN company must acquire a vendor service with proper infrastructural and technical expertise. An authorized vendor who can meet the security standards set by the company's internal policies and government agencies. While selecting the service provider ATN must carefully read the service level agreement and understand their policies and terms and provision of compensation in case of an outage or lock-in clauses.

- **Downtime**

Downtime is a significant shortcoming of cloud technology. No seller can promise a platform that is free of possible downtime. Cloud technology makes small companies reliant on their connectivity, so companies with an untrustworthy internet connection probably want to think twice before adopting cloud computing.

Downtime is often cited as one of the biggest disadvantages of cloud computing. Since cloud computing systems are internet-based, service outages are always an unfortunate possibility and can occur for any reason. (cloudacademy, n.d.)

Solution:

  - Consider implementing dedicated connectivity such as AWS Direct Connect, Azure ExpressRoute, or Google Cloud's Dedicated Interconnect or Partner Interconnect. These services provide a dedicated network connection between the ATN company and the cloud service point of presence. This can reduce exposure to the risk of business interruption from the public internet.
  - Define and implement a disaster recovery plan in line with business objectives that provide the lowest possible recovery time and recovery point objectives.
  - Design services with high availability and disaster recovery in mind. Leverage the multi-availability zones provided by cloud vendors in infrastructure.

- **Password Security**

Industrious password supervision plays a vital role in cloud security. However, the more people have to access the cloud account, the less secure it is. Anybody aware of passwords will be able to access the information ATN company store there.

Businesses should employ multi-factor authentication and make sure that passwords are protected and altered regularly, particularly when staff members leave. Access rights related to passwords and usernames should only be allocated to those who require them.

- **Data security**

Due to huge infrastructure, cost organizations are slowly switching to cloud technology. Data are stored in the CSP's infrastructure. As data do not reside in organization territory, many complex challenges arise. Some of the complex data security challenges in the cloud include the following:

  - The need to protect the confidential business, government, or regulatory data
  - Cloud service models with multiple tenants sharing the same infrastructure
  - Data mobility and legal issues relative to such government rules
  - Lack of standards about how CSPs securely recycle disk space and erase existing data
  - Auditing, reporting, and compliance concerns
  - Loss of visibility to key security and operational intelligence that no longer is available to feed enterprise IT security intelligence and risk management
  - A new type of insider who does not even work for a company but may have control and visibility into data

- **Limited control and flexibility**

Since the cloud infrastructure is entirely owned, managed, and monitored by the service provider, it transfers minimal control over to the customer.

To varying degrees (depending on the particular service), cloud users may find they have less control over the function and execution of services within a cloud-hosted infrastructure. A cloud provider's end-user license agreement (EULA) and management policies might impose limits on what customers can do with their deployments. Customers retain control of their applications, data, and services, but may not have the same level of control over their backend infrastructure.

Solution:

  - Consider using a cloud provider partner to help with implementing, running, and supporting cloud services.
  - Understand the responsibilities and the responsibilities of the cloud vendor in the shared responsibility model to reduce the chance of omission or error.
  - Make time to understand cloud service provider's basic level of support. Will this service level meet support requirements? Most cloud providers offer additional support tiers over and above the basic support for an additional cost.
  - Make sure the developer understand the SLA concerning the infrastructure and services are going to use and how that will impact agreements with customers.

- **Cost concerns**

Adopting cloud solutions on a small scale and for short-term projects can be perceived as being expensive. However, the most significant cloud computing benefit is in terms of IT cost savings. Pay-as-you-go cloud services can provide more flexibility and lower hardware costs, but the overall price tag could end up being higher than the ATN company expected. Until ATN company is sure of what will work best for the company, it's a good idea to experiment with a variety of offerings. ATN might also make use of the cost calculators made available by providers like Amazon Web Services and Google Cloud Platform.

Solution:

  - Try not to over-provision services, but rather look into using auto-scaling services.
  - Ensure the ATN company has the option to scale DOWN as well as UP.
  - Pre-pay and take advantage of reserved instances if ATN company have a known minimum usage.
  - Automate the process to start/stop instances to save money when they are not being used.
  - Create alerts to track cloud spending.

- **Cost barrier**

For efficient working of cloud computing, ATN company have to bear the high charges of the bandwidth. Businesses can cut down the cost of hardware but they have to spend a huge amount on the bandwidth. For smaller application cost is not a big issue but for large and complex applications it is a major concern. For transferring complex and intensive data over the network ATN company must have sufficient bandwidth. This is a major obstacle in front of small organizations, which restricts them from implementing cloud technology in their business.

- **Lack of knowledge and expertise**

ATN Company does not have sufficient knowledge about the implementation of cloud solutions. They have not expertise staff and tools for the proper use of cloud technology. Delivering the information and selecting the right cloud is quite difficult without the right direction. Teaching ATN's staff about the process and tools of cloud computing is a very big challenge in itself. Requiring an organization to shift its business to cloud-based technology without having any proper knowledge is like asking for disaster. They would never use this technology for their business functions.

- **Recovery of lost data**

Cloud services face an issue of data loss. A proper backup policy for the recovery of data must be placed to deal with the loss. Vendors must set proper infrastructures to efficiently handle server breakdown and outages. All the cloud computing service providers must set up their servers at economically stable locations where ATN should have proper arrangements for the backup of all the data in at least two different locations. Ideally, ATN should manage a hot backup and a cold backup site.

- **Cloud management**

Managing a cloud is not an easy task. It consists of a lot of technical challenges. A lot of dramatic predictions are famous about the impact of cloud computing. People think that the traditional IT department will be outdated and research supports the conclusions that cloud impacts are likely to be more gradual and less linear. Cloud services can easily change and update by business users. It does not involve any direct involvement of the IT department. It is a service provider’s responsibility to manage the information and spread it across the organization. So it is difficult to manage all the complex functionality of cloud computing

- **Vendor lock-in**

Vendor lock-in is another perceived disadvantage of cloud computing. Easy switching between cloud services is a service that hasn't yet completely evolved, and organizations may find it difficult to migrate their services from one vendor to another. Differences between vendor platforms may create difficulties in migrating from one cloud platform to another, which could equate to additional costs and configuration complexities. Gaps or compromises made during migration could also expose data to additional security and privacy vulnerabilities.

Solution:

- Design with cloud architecture best practices in mind. All cloud services provide the opportunity to improve availability and performance, decouple layers, and reduce performance bottlenecks. If ATN company have built services using cloud architecture best practices, ATN is less likely to have issues porting from one cloud platform to another.
- Properly understand what vendors are selling to help avoid lock-in challenges.
- Employ a multi-cloud strategy to avoid vendor lock-in. While this may add both development and operational complexity to deployments, it doesn't have to be a deal-breaker. Training can help prepare teams to architect and select best-fit services and technologies.
- Build-in flexibility as a matter of strategy when designing applications to ensure portability now and in the future.
- Build applications with services that offer cloud-first advantages, such as modularity and portability of microservices and code. Think containers and Kubernetes.

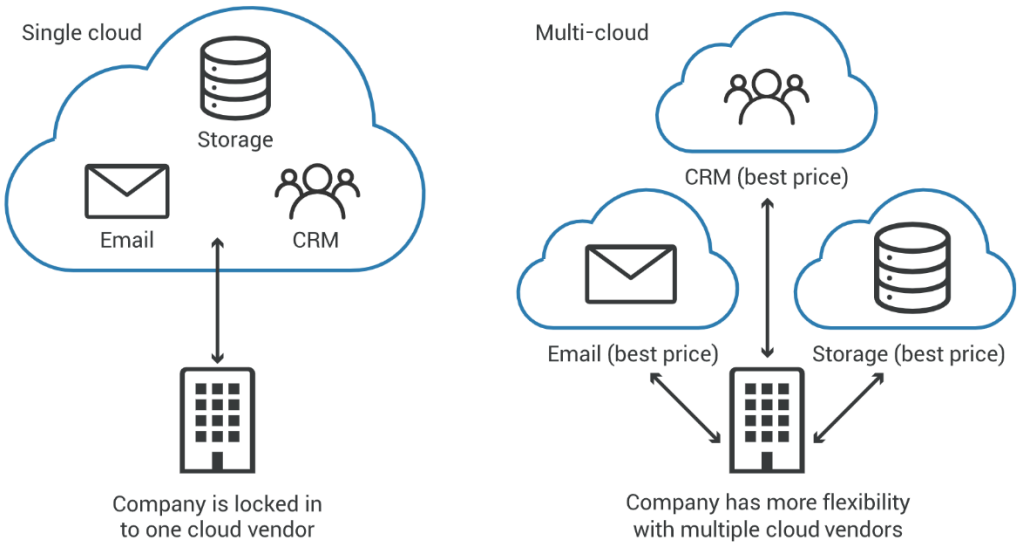


Figure 4. Vendor lock-in in cloud computing

2. The most common security issues in cloud environments and how to overcome security issues when building a secure cloud platform (P8-M4)

2.1. Database URL

The connection to the database should be secure, in the source code should not has information about databases such as user and password.

This is a bad way for ATN Admin Panel to connect to MongoDB server:

```
const run = async () => {
 await mongoose.connect('mongodb+srv://mido:123Chicken@clusteratn-ich09.mongodb.net/ATN?retryWrites=true&w=majority', {
  useNewUrlParser: true,
  useUnifiedTopology: true
 })
 app.listen(process.env.PORT, () => console.log(` Admin Panel is under localhost:${process.env.PORT}`))
}
```

Solution for overcome database URL:

By using the environment variable, the ATN system will be secured. The information about the connection to the database will not show up by replacing the **MongoDB server URL** with **process.env.MONGO_URL**

An environment variable is a variable whose value is set outside the program, typically through a functionality built into the operating system or microservice. An environment variable is made up of a name/value pair, and any number may be created and available for reference at a point in time.

Picture 69. Create .env file

```
const run = async () => {
 await mongoose.connect('process.env.MONGO_URL', {
  useNewUrlParser: true,
  useUnifiedTopology: true
 })
 app.listen(process.env.PORT, () => console.log(` Admin Panel is under localhost:${process.env.PORT}`))
}
```

2.2. Sniffer attacks

Any network packet having information in plain text can be intercepted and read by the attackers. This information can be usernames, passwords, secret codes, banking details, or any information which is of value to the attacker. This attack is just the technical equivalent of a physical spy. (greycampus, n.d.)

These types of attacks are launched by applications that can capture packets flowing in a network and if the data that is being transferred through these packets are not encrypted, it can be read and there are chances that vital information flowing across the network can be traced or captured. A sniffer program can use the NIC (Network Interface Card) ensures that the data/traffic linked to other systems on the network also gets recorded.

Sniffing motives:

- Getting username a passwords
- Stealing bank-related/transaction-related information
- Spying on email and chat messages
- Identity theft

Browsers can use content or MIME sniffing to adapt to different data types coming from a response. They override the Content-Type headers to guess and process the data. While this can be convenient in some scenarios, it can also lead to some dangerous attacks. This middleware sets the X-Content-Type-Options header to nosniff. This instructs the browser to not bypass the provided Content-Type.

Solution for overcome sniffer attacks:

Helmet packet can help for sniffer attacks

```
C:\Users\super\Desktop\Admin Panel ATN>yarn add helmet
yarn add v1.22.4
[1/4] Resolving packages...
[2/4] Fetching packages...
[-------------------------------------------------------------------------------------------------
```

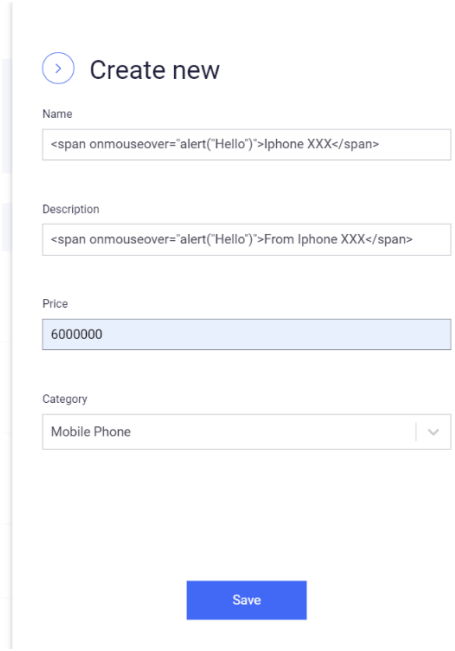
Picture 70. Install helmet packet

```
const helmet = require('helmet')
...
app.use(helmet.noSniff());
```

2.3. Cross-Site Scripting (XSS) attacks

Cross-site scripting (XSS) is a frequent type of attack where malicious scripts are injected into vulnerable pages, to steal sensitive data like session cookies or passwords.

The basic rule to lowering the risk of an XSS attack is simple: "**Never trust user's input**". A developer should always sanitize all the input coming from the outside. This includes data coming from forms, GET query URLs, and even from POST bodies. Sanitizing means that ATN company should find and encode the characters that may be dangerous e.g. <, >.



Picture 71. XSS Attacks

Modern browsers can help to mitigate the risk by adopting better software strategies. Often these are configurable via HTTP headers.

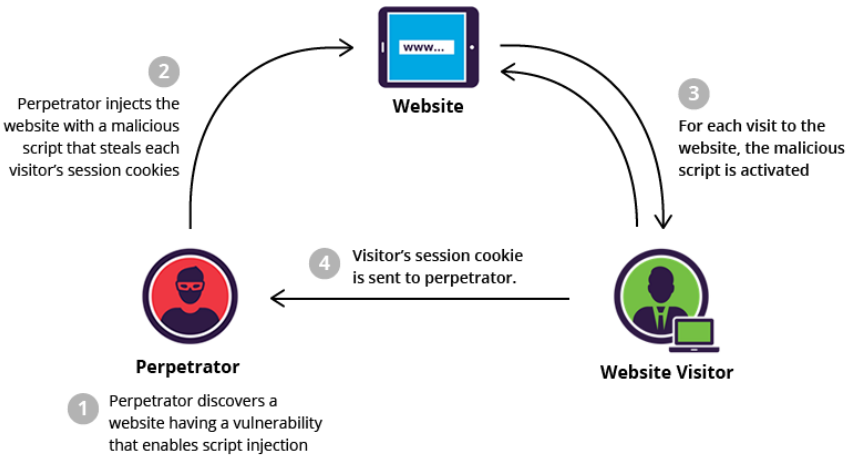


Figure 5. Cross-Site Scripting (XSS) attacks

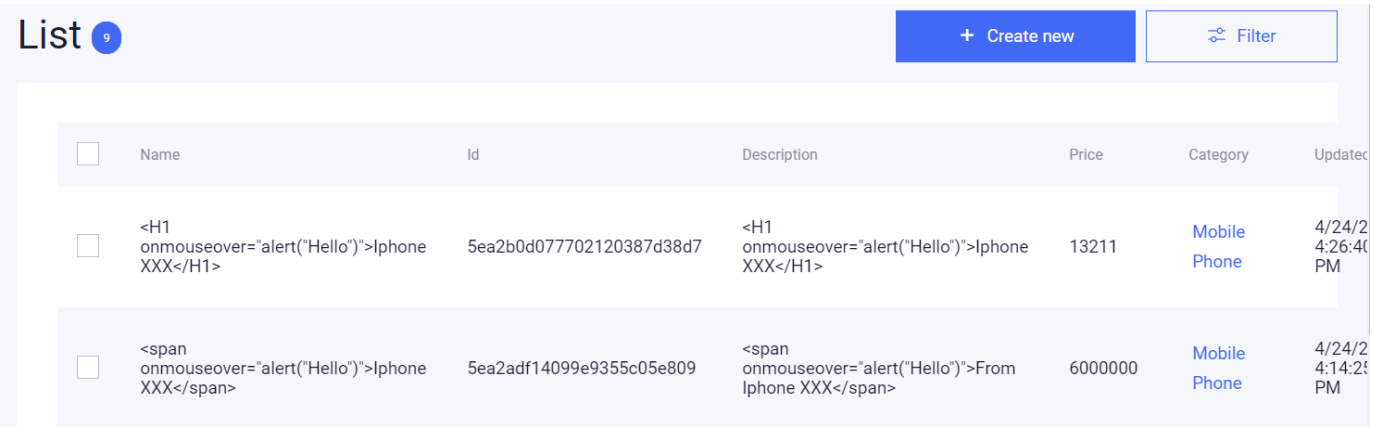
The solution to overcome:

The helmet helps secure Express apps by setting various HTTP headers.

The X-XSS-Protection HTTP header is basic protection. The browser detects a potential injected script using a heuristic filter. If the header is enabled, the browser changes the script code, neutralizing it.

```
const helmet = require('helmet')

...
app.use(helmet.xssFilter({}));
```



Picture 72. XSS Attack does not work on ATN Admin Panel

2.4.DoS & DDoS attacks

A DoS attack is a denial of service attack where a computer (or computers) is used to flood a server with TCP and UDP packets. A DDoS attack is where multiple systems target a single system with a DoS attack. The targeted network is then bombarded with packets from multiple locations.

A DoS attack is a denial of service attack where a computer (or computers) is used to flood a server with TCP and UDP packets. During this type of attack, the service is put out of action as the packets sent overload the server's capabilities and make the server unavailable to other devices and users throughout the network. DoS attacks are used to shut down individual machines and networks so that they can't be used by other users. (comparitech, n.d.)

There are several different ways that DoS attacks can be used:

- **Buffer overflow attacks:** This type of attack is the most common DOS attack experienced. Under this attack, the attacker overloads a network address with traffic so that it is put out of use.
- **Ping of Death or ICMP flood:** An ICMP flood attack is used to take unconfigured or misconfigured network devices and uses them to send spoof packets to ping every computer within the network. This is also known as a ping of death (POD) attack.
- **SYN flood:** SYN flood attacks send requests to connect to a server but don't complete the handshake. The result is that the network becomes inundated with connection requests that prevent anyone from connecting to the network.
- **Teardrop Attack:** During a teardrop DOS attack an attacker sends IP data packet fragments to a network. The network then attempts to recompile these fragments into their original packets. The process of compiling these fragments exhausts the system and it ends up crashing. It crashes because the fields are designed to confuse the system so that it can not put them back together.

A DDoS attack is one of the most common types of DoS attacks in use today. During a DoS attack, multiple systems target a single system with a DoS attack. The targeted network is then bombarded with packets from multiple locations. By using multiple locations to attack the system the attacker can put the system offline more easily. The reason for this is that there is a larger number of machines at the attackers' disposal and it becomes difficult for the victim to pinpoint the origin of the attack.

DDoS attacks can come in various forms including:

- **UDP Floods**: A UDP flood is a DDoS attack that floods the victim network with User Datagram Protocol (UDP) packets. The attack works by flooding ports on a remote host so that the host keeps looking for an application listening at the port. When the host discovers that there is no application it replies with a packet that says the destination wasn't reachable. This consumes network resources and means that other devices can't connect properly.
- **Ping Flood:** Much like a UDP flood attack, a ping flood attack uses ICMP Echo Request or ping packets to derail a network's service. The attacker sends these packets rapidly without waiting for a reply in an attempt to make the network unreachable through brute force. These attacks are particularly concerning because bandwidth is consumed both ways with attacked servers trying to reply with their ICMP Echo Reply packets. The result is a decline in speed across the entire network.

- **SYN Flood**: SYN Flood attacks are another type of DoS attack where the attacker uses the TCP connection sequence to make the victim's network unavailable. The attacker sends SYN requests to the victim's network which then responds with an SYN-ACK response. The sender is then supposed to respond with an ACK response but instead, the attacker doesn't respond (or uses a spoofed IP address to send SYN requests instead). Every request that goes unanswered takes up network resources until no devices can make a connection.
- **HTTP Flood**: In an HTTP Flood attack the attacker users HTTP GET or POST requests to launch an assault on an individual web server or application. HTTP floods are a Layer 7 attack and don't use malformed or spoofed packets. Attackers use this type of attack because they require less bandwidth than other attacks to take the victim's network out of operation.

The solution to overcome:

Load balancers are ideally suited for inclusion within a layered security model. The primary function of a load balancer is to spread workloads across multiple servers to prevent overloading servers, optimize productivity, and maximize uptime. Load balancers also add resiliency by rerouting live traffic from one server to another if a server falls prey to DDoS attacks or otherwise becomes unavailable. In this way, load balancers help to eliminate single points of failure, reduce the attack surface, and make it harder to exhaust resources and saturate links.

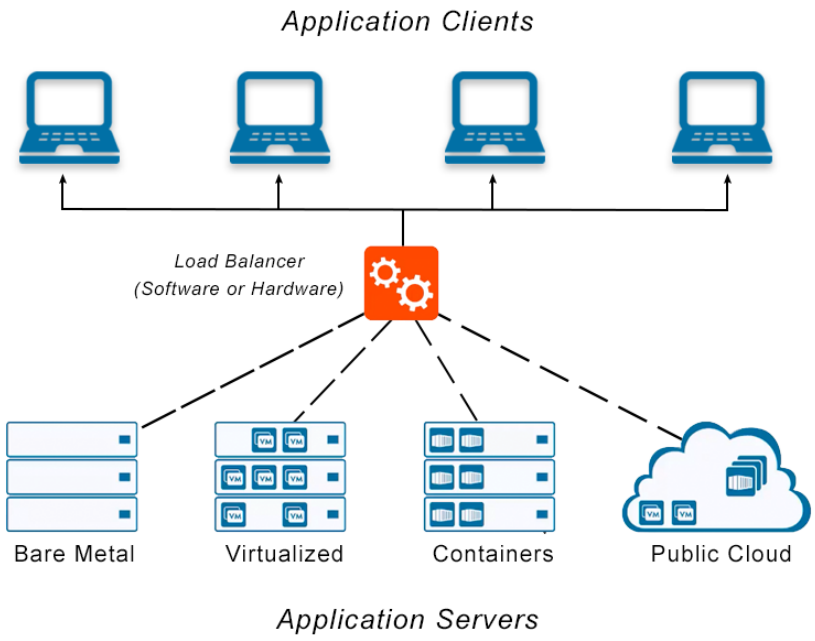


Figure 6. Using Load Balance

For this strategy to be truly effective, it's necessary to ensure that the data centers are connected to different networks and that there are no obvious network bottlenecks or single points of failure on these networks.

2.5. Cookie poisoning

Cookie poisoning is the act of manipulating or forging a cookie (a small piece of data created and stored in a user's browser that keeps track of important information regarding his or her session information for a particular site) to bypass security measures or send false information to a server. An attacker using cookie poisoning can gain unauthorized access to a user's account on the particular site the cookie was created for, or potentially tricking a server into accepting a new version of the original intercepted cookie with modified values.

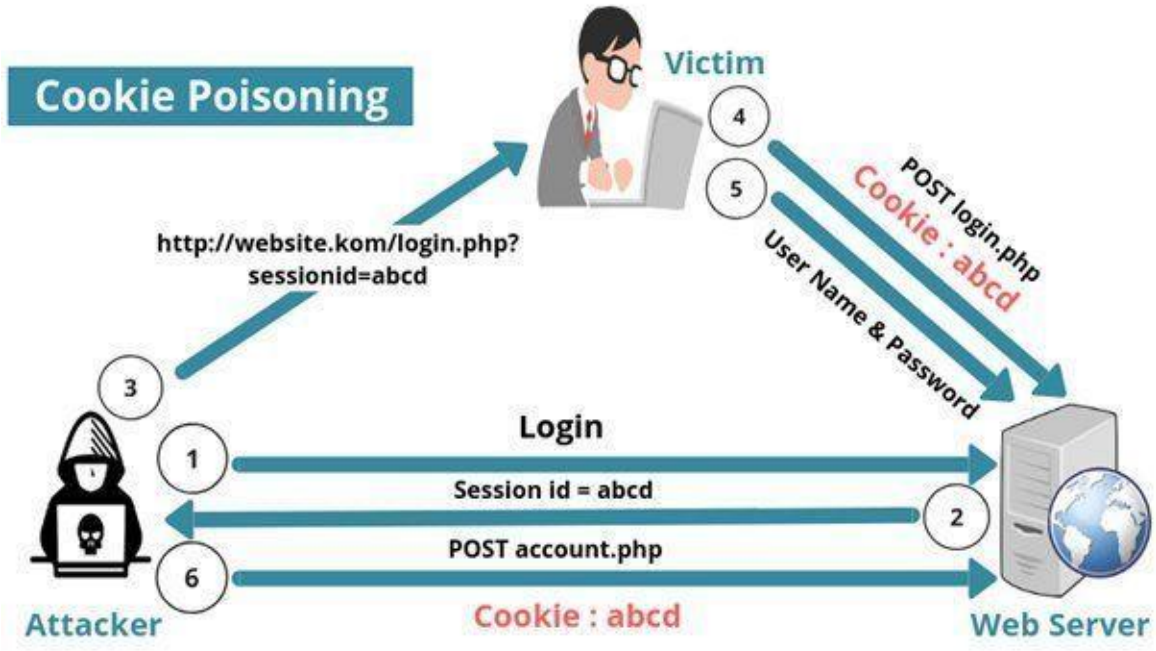


Figure 7. Cookie poisoning. (twitter, n.d.)

The solution to overcome:

- Require the user to use https (443) the entire session. This will prevent any man-in-the-middle attacks from sniffing the cookie
- Only allow one session to be active at a time. Once the second session shows up, the first session is invalidated.
- Require the user to provide his old password when changing the password (edit: or email address or anything else that could allow the account to be stolen once the attacker is logged in); this will prevent someone from hijacking the account and easily changing the password.
- Have a very limited life for the session cookie - a few hours.

3. How an organization should protect their data when they migrate to a cloud solution (D3)

Nowadays most organizations migrate data to the cloud. Unsurprisingly, the large cloud vendors make it as easy as possible to migrate data to the cloud and provide a raft of tools to help with its migration.

Nonetheless, the ultimate responsibility for migrating data and maintaining its security belongs to the ATN company, and ATN company must be aware of the pitfalls as well as the upsides of cloud migration.

3.1. Know the data

In the past, enterprises would often treat their data storage like an attic or a basement in one's home just store it there, and it doesn't matter what 'it' is.

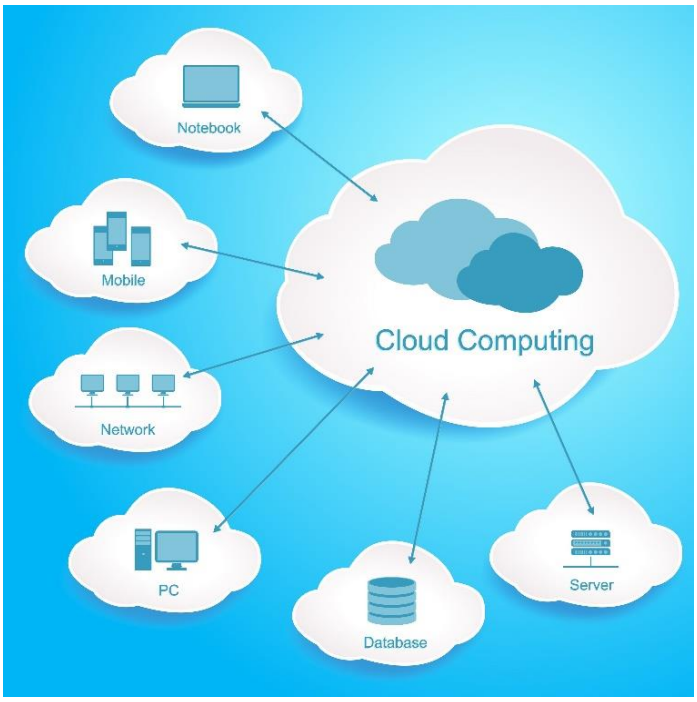


Figure 8. Know data in the cloud. (ebuyer, n.d.)

With data so valuable now, a potential migration is an excellent reason to take stock of exactly what the data contains. Is it in a form that will be usable now and later? How accurately is the data conforming to storage and retention policies within the data governance framework? And what can ATN company safely dispose of?

There are many data management tools available to assist the process of understanding exactly what data ATN company has in the domain. Most cloud vendors will offer their solutions to help the company understand data, and there's a range of paid services available on the market.

Make good use of these tools. And avoid the temptation to hoard. Too often companies hold on to data they no longer need for. Unnecessary data retention (unless legally required) increases a company's liability in the event of a security breach.

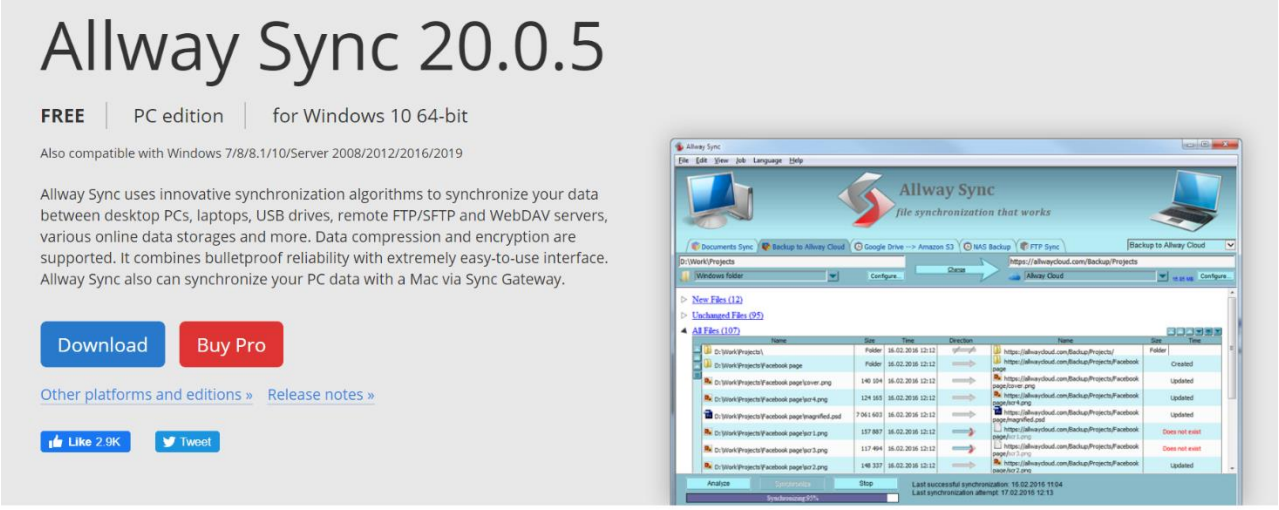
Data migration is a good opportunity to clean each ATN shop. (horizontechnology, n.d.)

3.2. Back-Up Data Locally

When migrating to cloud ATN company has comes to managing data to always have a backup for data. Generally speaking, it is good practice to create electronic copies for any of the ATN company's data so that ATN will still be able to access them even when the original is lost or has been corrupted. There are many cloud storage services available in the market today, which means the ATN company can set up some cloud accounts for backup purposes.

If ATN company has data in the cloud, the ATN company should also manually backup data in an external physical storage drive or device, like a hard disk or a thumb drive. This also allows ATN company to access the information when ATN company have poor or no Internet connection.

Allway Sync synchronizes ATN company's files and directories among PC, external drives, remote FTP servers, and more with an innovative synchronization algorithm. It comes with an easy-to-use graphical interface and can generate a report for any synchronization actions.



Picture 73. Backup data in local with allway Sync

3.3. Setting ATN company Data Retention Policy

Data retention policies are formulated to not just store but also to organize the information so that it can be easily searched for and accessed at any point in the future. The other side of this objective is to get rid of data that has outlived its usefulness. Hence, it becomes important for businesses to understand the policies to avoid getting caught unaware and dealing with the complexities of losing sensitive and important data.

A data retention policy, which should be clearly documented and centrally owned, is generally designed in response to a combination of operational needs and regulatory requirements.

Core components of a well-designed data retention policy include:

- An overview of the different types of information the business handles customer data, financial reports, legal agreements, billing information, email, strategy documents, meeting notes, marketing collateral, and so on.
- Rules for how long different types of data should be retained and guidance on how the information should be stored. What's the policy on encryption, not only for data in flight and data at rest but data in use?
- A permissions-based framework specifying authorizations to access retained data. What is a protocol for setting access privileges, and how are ATN company enforcing that? What about managing vendor and contractor access to data?
- An accurate and accessible description of the regulatory landscape and an explanation of how data retention policy ensures compliance.
- Clear protocol for the standards around the digital sanitization of the data, as well as the physical destruction of paper and (where necessary) electronic equipment.

While data loss on the Cloud happens due to a variety of reasons (like hacking, intentional deletion, a software malfunction, etc.), deletion due to human error tops the chart. Most businesses wrongly assume that data on the Cloud has a permanent life and even if it gets wiped off from an application, intentionally or otherwise, it can somehow be retrieved.

Cloud applications delete the data permanently after retaining it for a specific time.

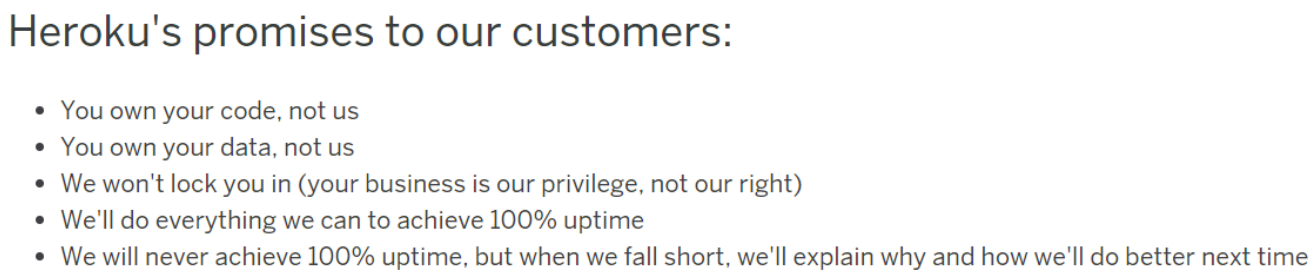
3.4. Read the Small Print of the Cloud Service Provider

Besides storing ATN's data, some cloud services allow ATN company to share photos and files with others. This sounds appealing, but sometimes these services come with a catch. There might be some fine print that they don't advertise but will stuff in their **Terms of Service** to make it legitimate.

For Example: Back in 2011, Twitpic wrote in their TOS that sharing pictures on their service gives them the right to 'use or distribute' the pictures. They later apologized but further clarified that they can distribute the securing-cloud-data on Twitpic and affiliated partners, although the final copyright still belongs to the owner of the photographs.

While not exactly a dedicated cloud storage service, Twitpic puts forward a good case for why ATN company should be cognizant of what to expect from the cloud provider, especially concerning their security and privacy policies.

In this case of ATN company, the acceptable policy(https://www.heroku.com/policy/aup) is acceptable.



Picture 74. Heroku's promises to customers

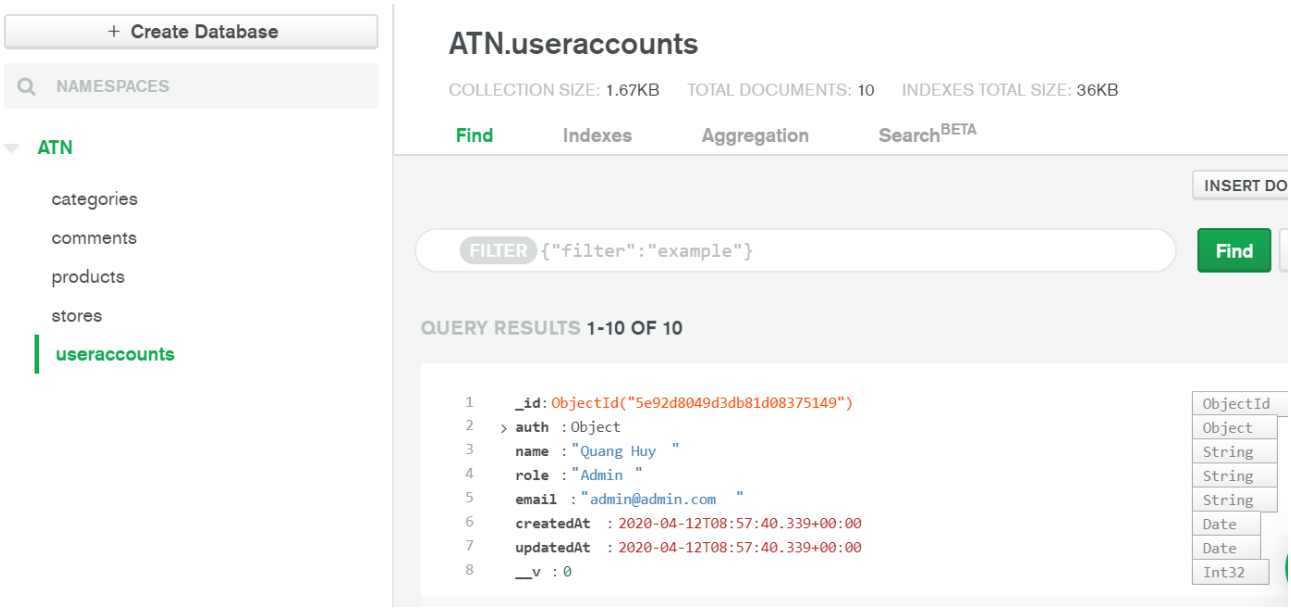
3.5. Avoid storing sensitive information in the cloud

Avoid storing sensitive information on the cloud. In addition to the obvious, such as social security numbers, copies of IDs, or important financial statements even old ones consider what other information someone could get their hands on. Never keep racy pictures or intimate interactions with partners in the cloud and if ATN company is sensitive about items such as diet progress pictures, avoid posting those as well.

ATN company needs to keep only those files which the company need to access frequently and avoid putting up documents containing passwords for various online accounts or personally identifiable information (PII) such as credit card numbers, national identification number, home address, etc.

If the ATN company must include this information in files, make sure to encrypt them before upload it.

For example in ATN company is the password of the user account will be encrypted before uploading to the cloud.



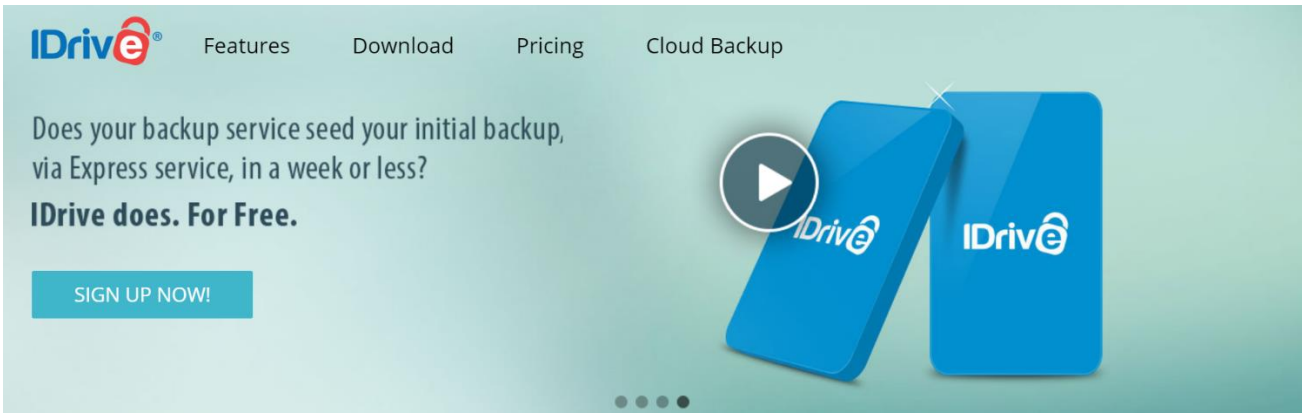
Picture 75. Password will encrypt before uploading into cloud

3.6. Use Cloud Services That Encrypt ATN company Data

One of the easiest ways to safeguard privacy when using cloud storage services is to look for one that offers local encryption for data. This provides an additional layer of security since decryption will be required before the ATN company can be granted access to the data.

While keeping data encrypted in the cloud may be good enough, it would be even better if the cloud service also ensures encryption during the uploading and downloading phases. This can be done using military-grade **Advanced Encryption Standard** (AES) (256 bits), which services like DrivePop adopt.

IDrive transfers and stores data using 256-bit AES encryption, which makes data highly secured. The IDrive client is available for Windows, Mac OS, Android, and iOS. It's easy to store and backup files and folders.



Picture 76. Backup service IDrive

ATN company can back up files from multiple devices in one account and manage multiple accounts through its client. They offer 5GB free storage.

3.7. Protect ATN company's system with Anti-Virus & Anti-Spy

ATN company may be using a secure cloud service provider which the company trust, but sometimes the weakest link happens to be the computer system that ATN company logging in from. Without proper protection for the system, ATN company will expose to bugs and viruses that provide penetration points for hackers to access accounts.

Take for instance the presence of a Keylogger Trojan which attempts to track all keystrokes. By embedding this malicious software to seemingly legitimate files, hackers will be able to get hold of user ID and password if ATN's system isn't well protected enough to detect it, and if the login isn't secured and encrypted.

3.8. Encrypt the Data Before Putting it on The Cloud

If the ATN company chooses not to use a cloud service that will help the company encrypt the data, ATN can use a third-party tool to perform the encryption. All ATN got to do is download a cloud-protection app which will allow ATN company to apply passwords and generate secret key sequences to files before the ATN company upload them to the cloud.

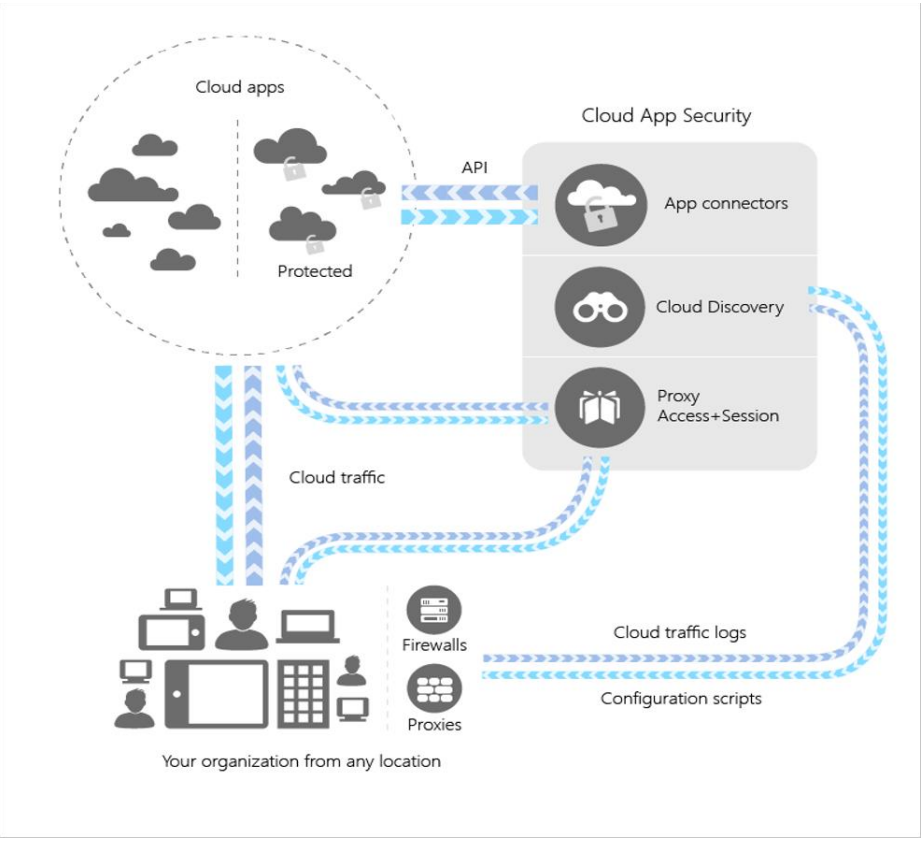


Figure 9. Cloud app security

Even if ATN company is already opting for an encrypted cloud service, it wouldn't hurt to go through a preliminary round of encryption for files to get a little extra assurance.

3.9. Use a Strong Password / Use Two-Factor Authentication

While practices such as the principle of least privileges can protect data, ATN company should also take the further step of enforcing two-factor authentication to reduce the risk of unauthorized access to mission-critical data stored in the cloud. Two-factor authentication requires the use of two separate methods of identification to gain access to restricted cloud storage or applications.

ATN company will require Password with:

- Minimum Password Length: 8
- Must Meet 1 special character: !@#$%^&*()_=+
- Must Meet 1 number: [0-9]
- Must Meet 1 uppercase character

As the first line of defense against malicious hackers out there, ATN company had better be sure that the password can stand a hacking or cracking attempt. There are tons of tips on the Internet on what makes for a good password. Aside from going for a strong and unique password, make sure to change it frequently and not repeat it across all other online accounts ATN has.

Alternatively, ATN company may go for the much more secure two-step verification for login if cloud service offers the option.

In the case of Google Drive, users have to log in to their Google account first to use the cloud storage service. Two-step verification can be turned on for Google accounts that is a verification code sent to the mobile phone gives the much needed added security on top of just a password to be able to access cloud data.

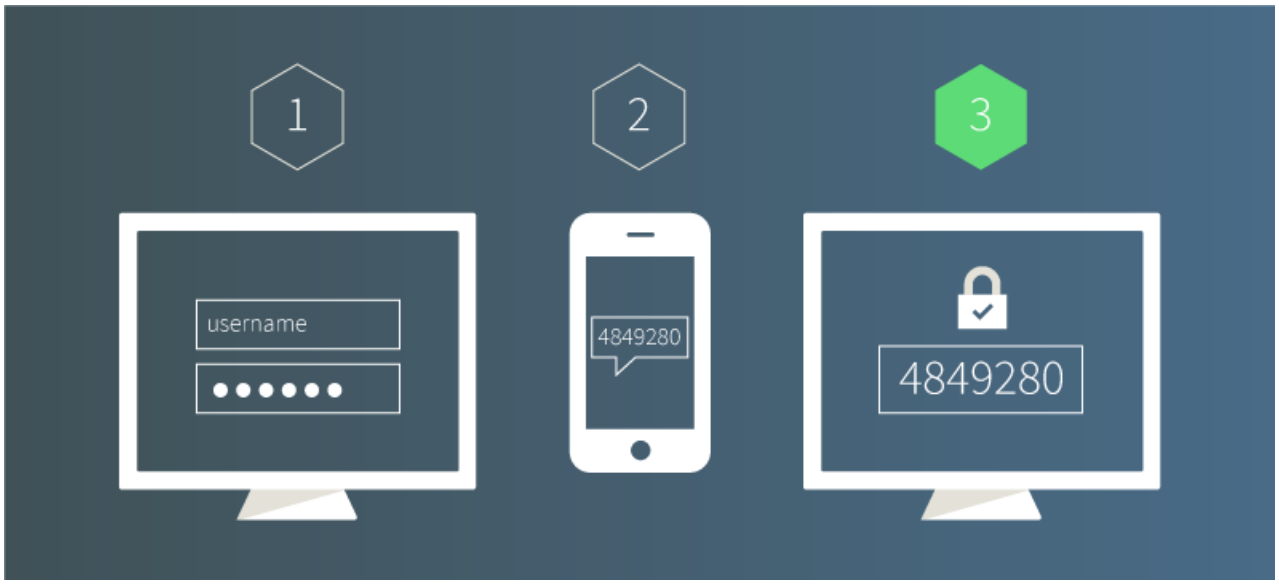


Figure 10. Use Two-Factor Authentication

CONCLUSION

Considering the numerous benefits that cloud computing offers to organizations, a fair case can be made that cloud computing is increasingly becoming the new normal. Cloud computing is helping society to cope with future problems such as managing big data, cyber-security, and quality control. In addition to this, emerging technologies such as Artificial Intelligence, distributed ledger technology, and many other capabilities are becoming available as services through cloud computing.

Consequently, these technologies to be adaptable to various platforms such as mobile devices hence increasing their use. Innovations based on cloud computing such as cloud automation and the Industry cloud are also being developed to integrate cloud computing into more specific industrial activities which will make various operations even more streamlined. The final verdict for cloud computing is that it's a transformational technology that has helped organizations in different jurisdictions to deliver their products and services in a better way than before.

References

algoworks. (n.d.). *algoworks*. Retrieved from algoworks: https://www.algoworks.com/blog/why-use-expressjs-over-nodejs-for-server-side-coding/

cloudacademy. (n.d.). *cloudacademy*. Retrieved from cloudacademy: https://cloudacademy.com/blog/disadvantages-of-cloud-computing/

comparitech. (n.d.). *comparitech*. Retrieved from comparitech: https://www.comparitech.com/net-admin/dos-vs-ddos-attacks-differences-prevention/

ebuyer. (n.d.). *ebuyer*. Retrieved from ebuyer: https://www.ebuyer.com/blog/2015/11/where-is-my-cloud-data-stored/

educba. (n.d.). *educba*. Retrieved from educba: https://www.educba.com/cloud-computing-issues-challenges/

greycampus. (n.d.). *greycampus*. Retrieved from greycampus: https://www.greycampus.com/blog/information-security/what-is-a-sniffing-attack-and-how-can-you-defend-it

horizontechnology. (n.d.). *horizontechnology*. Retrieved from horizontechnology: https://www.horizontechnology.com/news/seven-tips-for-protecting-your-data-during-a-cloud-migration/

twitter. (n.d.). *twitter*. Retrieved from twitter: https://twitter.com/hashtag/cyberwarning?lang=cs

wikipedia. (n.d.). *wikipedia*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/Web_performance