

Steps to follow

Hey! Thanks for purchasing the source code. There are a few things you need to do first. If you skip these steps, you won't be able to see the site in action. Let's get started!

Node Version Used: **node v23.10.0 (npm v11.2.0)**

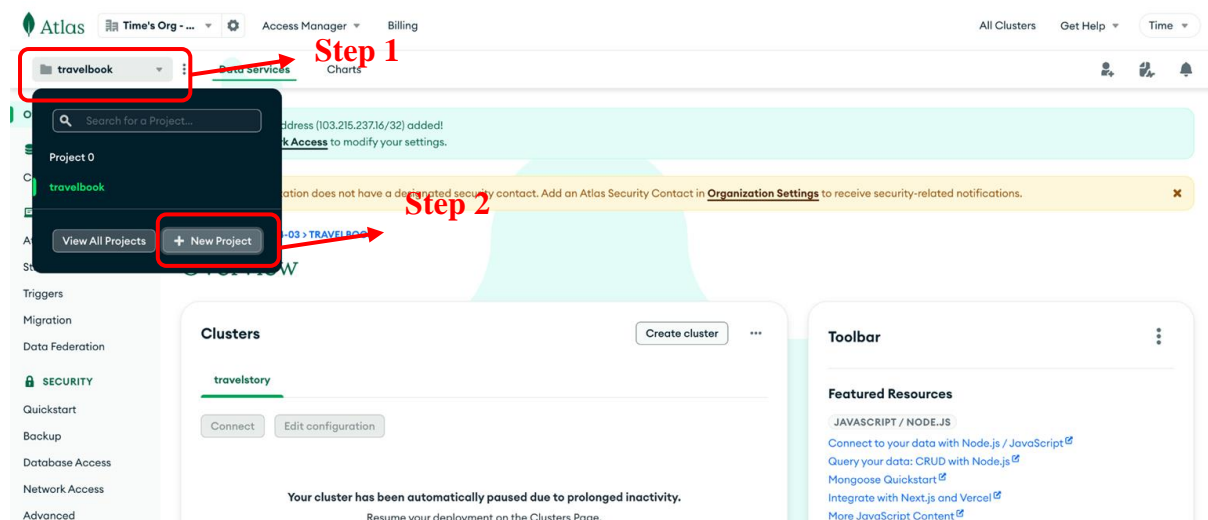
Running the Resume Builder App Project

Backend (Express.js)

1. Navigate to the `backend` folder in your terminal.
2. Run the following command to install the required dependencies:

```
npm install
```

3. Once the dependencies are installed, Let's connect MongoDB
4. Go to <https://www.mongodb.com/>
5. Login or Create an Account
6. Now let's create a project by clicking on the "New Project" button



7. Now, Enter the project name and Click "Next"

Atlas Time's Org - ... Access Manager Billing All Clusters Get Help Time

ORGANIZATION

Projects

Alerts

Activity Feed

Settings

Integrations

Access Manager

Billing

Support

Live Migration

TIME'S ORG - 2024-04-03 > PROJECTS

Create a Project

Name Your Project Add Members

Project names have to be unique within the organization (and other restrictions).

Polling App

Add Tags (Optional)

Use tags to efficiently label and categorize your projects. A project can have a maximum of 50 tags. You can modify tags for the project later. [Learn more](#)

Key	Value	Actions
Select a key or enter your own	: Select a value or enter your own	
+ Add tag		
		0 TAGS

Cancel Next

8. Add Members if needed. Then click on “Create Project”

Atlas Time's Org - ... Access Manager Billing All Clusters Get Help Time

ORGANIZATION

Projects

Alerts

Activity Feed

Settings

Integrations

Access Manager

Billing

Support

Live Migration

TIME'S ORG - 2024-04-03 > PROJECTS

Create a Project

✓ Name Your Project Add Members

Add Members and Set Permissions

Invite new or existing users via email address...

Give your members access permissions below.

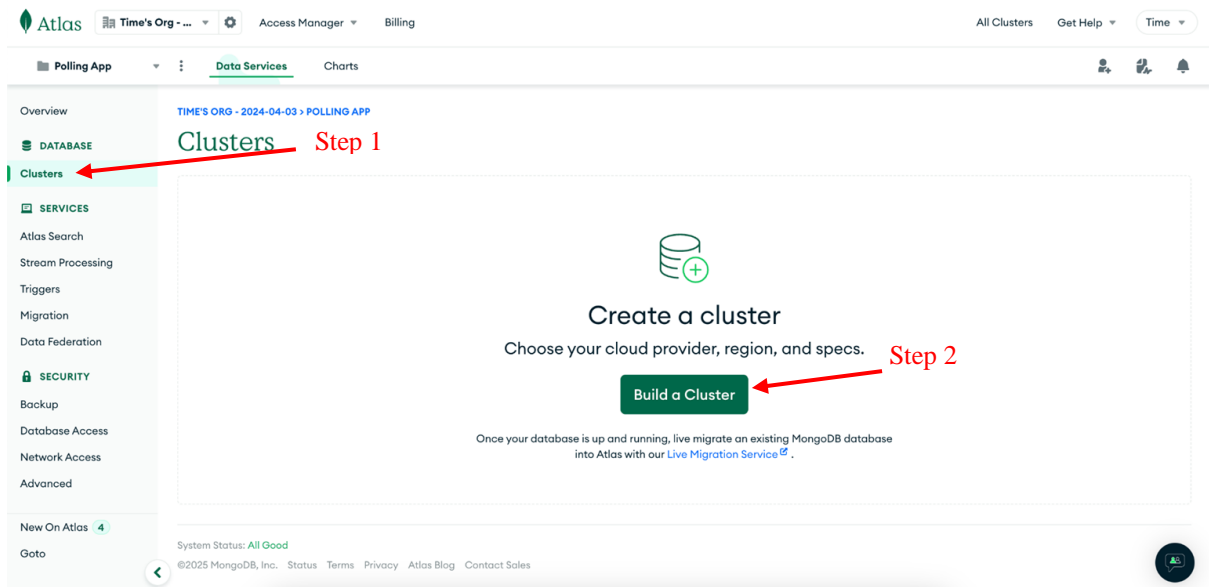
ii.com (you)	Project Owner
--------------	---------------

Back Cancel Create Project

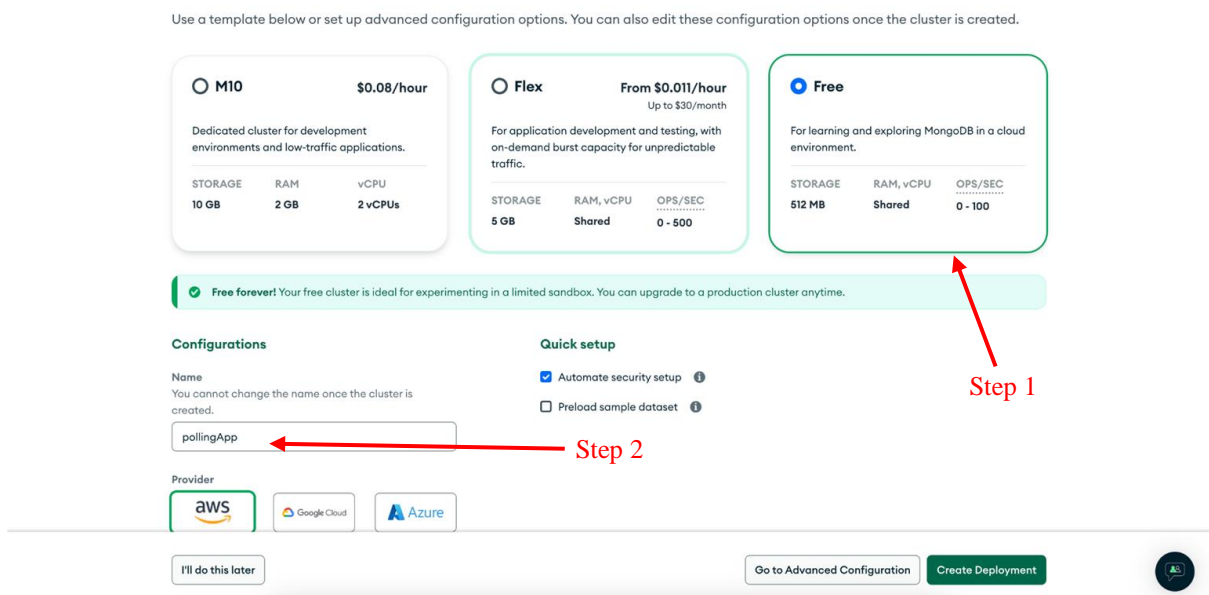
Project Member Permissions

- Project Owner**
Has full administration access
- Project Cluster Manager**
Can update clusters
- Project Data Access Admin**
Can access and modify a cluster's data and indexes, and kill operations
- Project Data Access Read/Write**
Can access a cluster's data and indexes, and modify data
- Project Data Access Read Only**
Can access a cluster's data and indexes
- Project Search Index Editor**
Can view and manage a cluster's search indexes
- Project Read Only**

9. Now click on “Clusters” option in the side menu and click on “Build a Cluster” button



10. Now, select free tier and give a cluster name



11. Select a server provider, select a region that's near you, and click on "Create Deployment"

Free forever! Your free cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Configurations

Name
You cannot change the name once the cluster is created.

Quick setup

☒ Automate security setup ⓘ

☐ Preload sample dataset ⓘ

Provider

aws

Google Cloud

Azure

Region

India Mumbai (ap-south-1) ★

★ Recommended ⓘ

★ Low carbon emissions ⓘ

Tag (optional)
Create your first tag to categorize and label your resources; more tags can be added later. [Learn more.](#)

I'll do this later

Go to Advanced Configuration

Create Deployment

12. Now we will be directed to the connection steps page.

13. Here, we need to add an IP address for the connection. I usually select the `Allow Access from Where` option and create a database user. And click on the "Choose a connection method" button

Atlas

Ravi's Org

Notes App

Data Service

Overview

DEPLOYMENT

Database

Data Lake

SERVICES

Device Sync

Triggers

Data API

Data Federation

Atlas Search

Stream Processing

Migration

SECURITY

Quickstart

Backup

Database Access

Network Access

Connect to notesdb

1

2

3

Set up connection security

Choose a connection method

Connect

You need to secure your MongoDB Atlas cluster. You can use IP addresses or IP ranges to restrict access to your cluster. You can use the IP address of your computer or a virtual private network (VPN) to access your cluster. You can also use the IP address of a cloud provider to access your cluster.

Step 1

1. Add a connection IP address

Add Your Current IP Address

Add a Different IP Address

Allow Access from Anywhere

Step 2

2. Create a database user

This first user will have [atlasAdmin](#) permissions for this project. You'll need your database user's credentials in the next step.

Username

testuser

Password

testuser123

HIDE

Copy

Create Database User

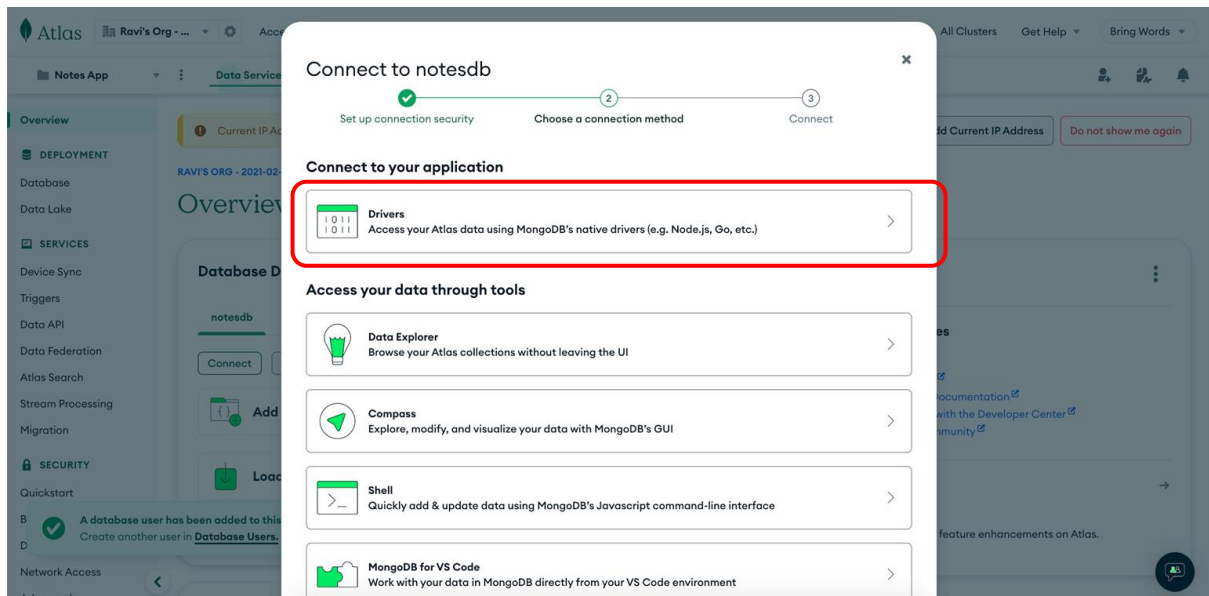
Step 3

Step 4

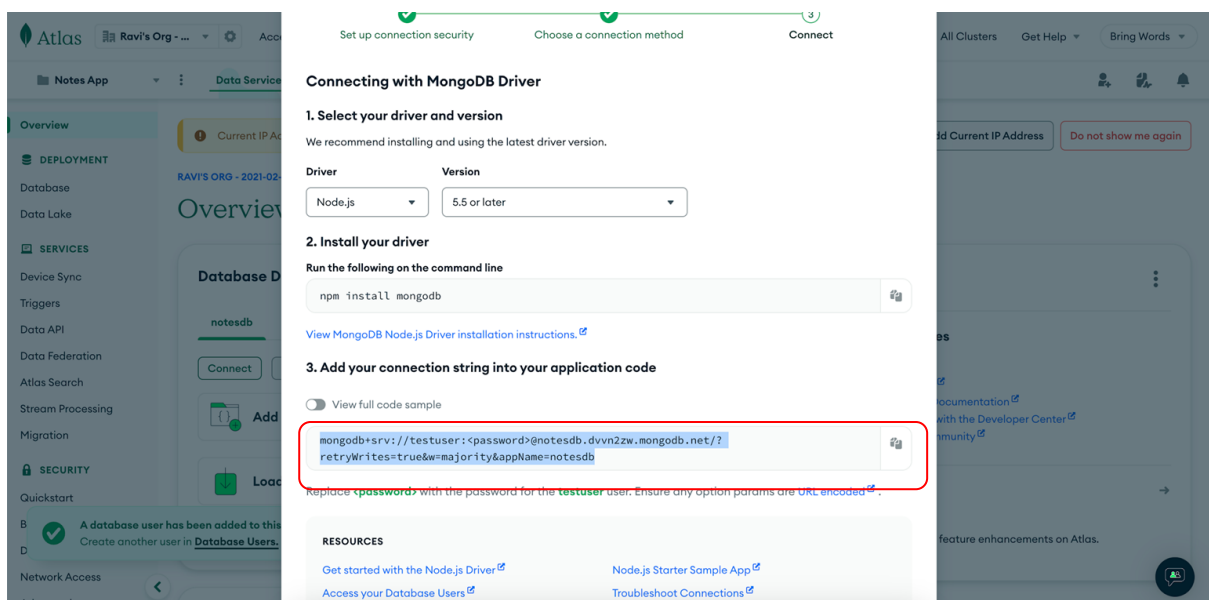
Cancel

Choose a connection method

14. In the Next step, Select the `Drivers` option to access the atlas database using our Node.js project

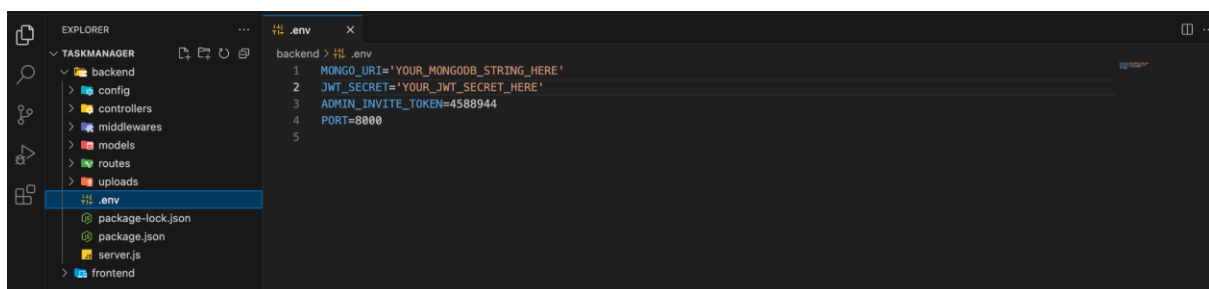


15. Now, copy the connection string

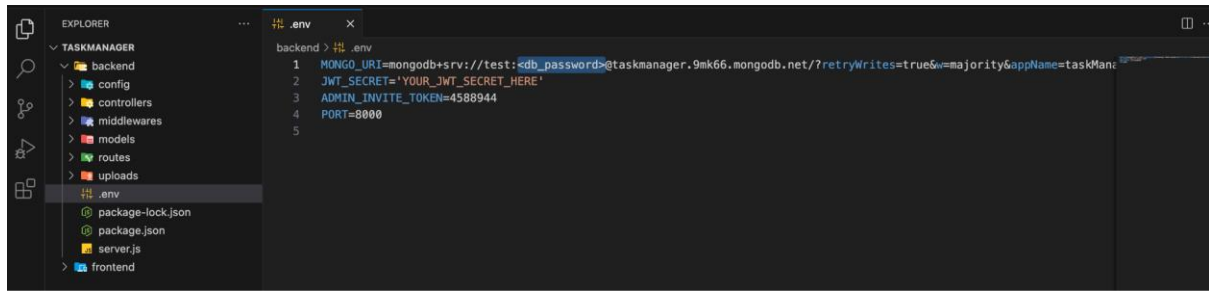


16. Paste the connection string inside the `.env` file:

Before:



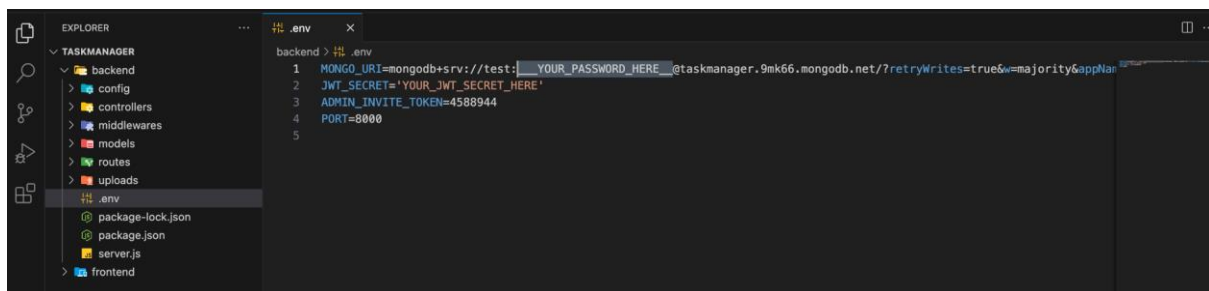
After:



The image shows a VS Code window with the Explorer sidebar on the left displaying the project structure. The 'TASKMANAGER' folder is expanded, showing subfolders like 'backend', 'config', 'controllers', 'middlewares', 'models', 'routes', 'uploads', and files like '.env', 'package-lock.json', 'package.json', 'server.js', and 'frontend'. The '.env' file is selected and its content is shown in the editor:

```
1 MONGO_URI=mongodb+srv://test:<db_password>@taskmanager.9mk66.mongodb.net/?retryWrites=true&w=majority&appName=taskMan
2 JWT_SECRET='YOUR_JWT_SECRET_HERE'
3 ADMIN_INVITE_TOKEN=4588944
4 PORT=8000
5
```

17. Now replace ``<password>`` in the connection string with the user's password that we have created in Step 13

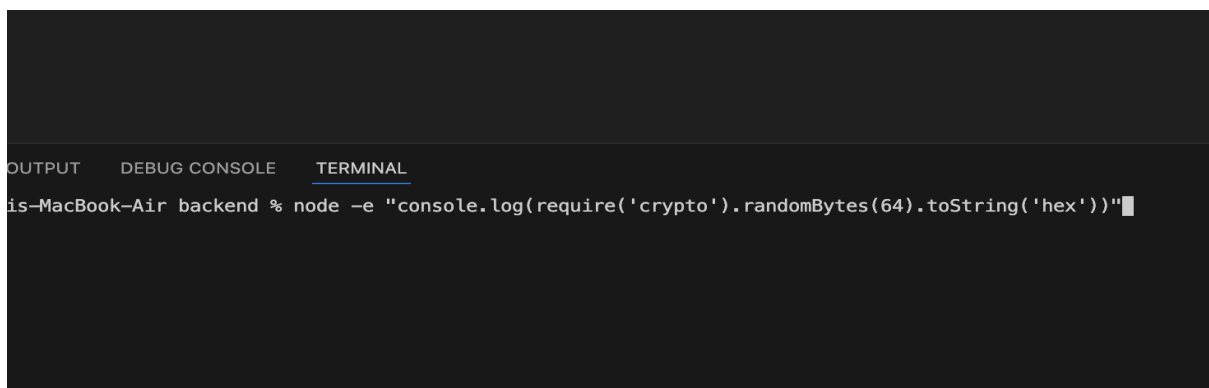


The image shows the same VS Code window as before, but the '.env' file content has been updated to replace the password placeholder with a text prompt:

```
1 MONGO_URI=mongodb+srv://test:YOUR_PASSWORD_HERE@taskmanager.9mk66.mongodb.net/?retryWrites=true&w=majority&appName=taskMan
2 JWT_SECRET='YOUR_JWT_SECRET_HERE'
3 ADMIN_INVITE_TOKEN=4588944
4 PORT=8000
5
```

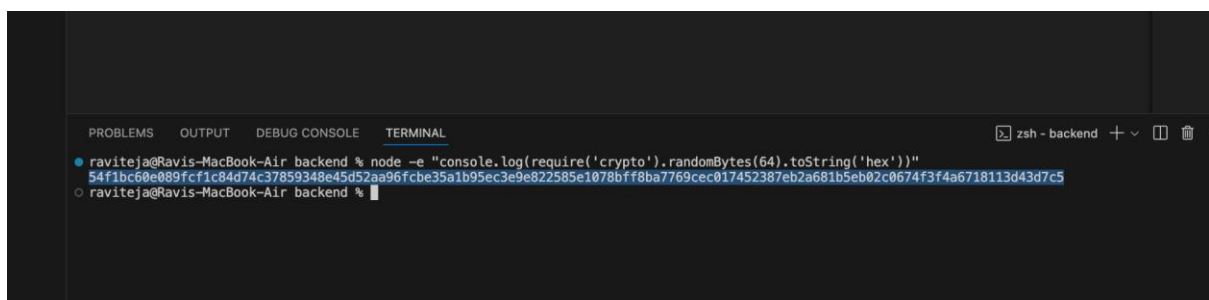
18. After updating the connection string, let's generate our JWT_SECRET. To do so, you can run the following command in the terminal.

`node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"`



The image shows a terminal window with the command prompt 'ravis-MacBook-Air backend %'. The command `node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"` has been entered and executed. The output is a long hexadecimal string:

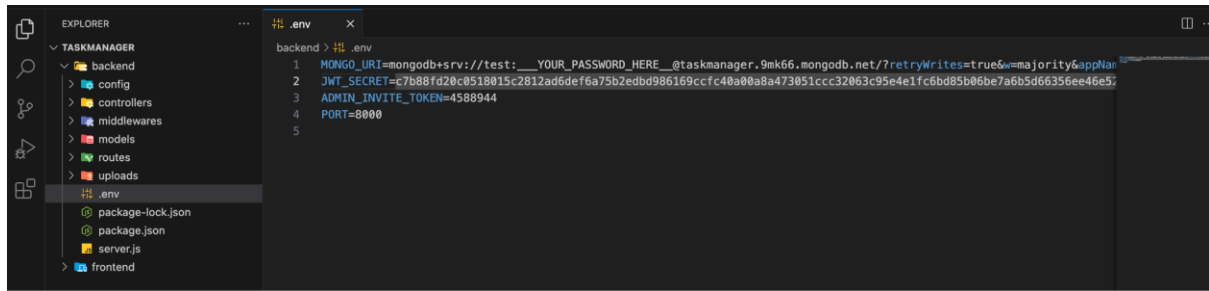
```
54f1bc68e889fcf1c84d74c37859348e45d52aa96fcb35a1b95ec3e9e822585e1078bff8ba7769cec017452387eb2a681b5eb02c0674f3f4a6718113d43d7c5
```



The image shows a terminal window with the command prompt 'ravis-MacBook-Air backend %'. The command `node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"` has been entered and executed. The output is a long hexadecimal string:

```
54f1bc68e889fcf1c84d74c37859348e45d52aa96fcb35a1b95ec3e9e822585e1078bff8ba7769cec017452387eb2a681b5eb02c0674f3f4a6718113d43d7c5
```

19. Now lets update the JWT_SECRET



That's it, we are done.

20. Now, start the server by running:

```
npm run dev
```

Frontend

1. Navigate to the `frontend/resume-builder` folder.
2. Run the following command to install the required dependencies:

```
npm install
```

3. After the installation is complete, start the React development server by running:

```
npm run dev
```

This will start the frontend server and open the app in your default web browser.