

Deployment

Relevel
by Unacademy



Educator Introduction

Give a brief intro about yourself and talk about the features we have implemented so far in the MBA app.

Features we implemented so far:

- CRUD operation APIs on Theatre and Movie models and APIs relating movies and theatres.
- Authentication and Authorization, using JWT for token validation.
- Several middleware, from model data verification to user access level verification.
- Model for booking and transaction, like create booking, cancel booking and set timeout for a booking.

List of Topics Covered

- Hosting.
- Intro of GitHub.
- Repository Creation.
- Get Database for production.
- Intro to Heroku.
- Understand the deployment process of our movie booking application.

What is Hosting?

- Now that our project is working fine in the local system. It would be useful only when used by others, i.e., it is available to the world. Our purpose can be achieved when we make our application available to the users via the internet and that is what hosting means.
- Many cloud service providers help us rent their server where our application code runs 24/7. These services are provided based on a free trial and price-based model for individuals and organizations—for example, Amazon AWS, Heroku, Microsoft Azure, Google Cloud, etc.
- We will deploy our application using the services of Heroku.

Run a project locally

- Before deploying the project, ensure it's running as expected by running the code and accessing the app locally.
- Execute below command in cmd terminals of root directory:
- **npm start**
- When we run npm start from terminal it will run above script which we need to add in package.json file which will be responsible to start up the application.
- As this is the standard script which many applications hosting service understand for a node application as a run command.

Intro to GitHub

- Now, you can serve your local setup online and make it live via the internet, but it's not easy to maintain and make your server available 24/7. That's why we need some remote server to deploy it and forget, so the cloud/server provider will maintain our application for us.
- We need to take care of one thing before deploying the app, i.e. maintenance and continuous changes to the application.
- We will likely add more features or make some critical changes to our code in the future or maintain different versions of our code.
- Also, things work differently in the industry. There will be many contributors for a single project; frontend developers, UI designers, backend developers, and software testers all working on the same project but locally on their respective machines.
- Finally, all the code from these machines will be merged as a whole, maintaining the flow and integrity of the code and won't break the code's functionality.
- So, here GitHub comes into the picture. As we can integrate Github with the remote server and the local setup to reflect the changes, we can easily change it without going to the server.

Here we will see how Github works and all the basic functionality/commands of GitHub:

1. Creating Repositories:

- a. **git init:** to initialise the repo.

2. Creating branches

- a. **git checkout -b <branch_name> :** to create new branch
- b. **git checkout <existing_branch_name> :** to get back to the original branch.

3. Adding files:

- a. **git add . :** to add all items
- b. **git add <file_name> :** to add specific elements

4. Commits

- a. **git commit -m <commit message> :** to commit the changes locally

5. Pull

- a. **git pull origin <branch_name> :** to get the latest code from the repo

6. Push

- a. **git push origin <branch_name>:** to push local code to the remote repo.

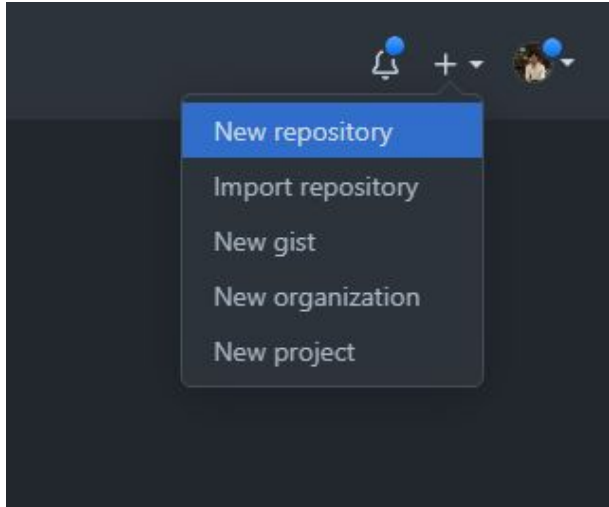
We have 3 ways to do all these steps:

1. Web Interface: Very easy to use but not for complicated steps like resets, rebase, etc.
2. Command Line: Not much user-friendly, but we can perform every operation using this one.
3. Github Desktop: Very much similar to web interface in terms of UI,

Creating Repositories

Using Web Interface:

1. Go to **Repositories** click on “**New**” or on the “+” icon in the top-right corner.
2. Click on **New repository**




3. Name the repo, create it.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

Repository name *

 kumarmohitt19

 /

mba_backend ✓

Great repository names are short, lowercase, and hyphenated. [mba_backend](#) is available. [inspiration?](#) How about [fuzzy-tribble?](#)

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

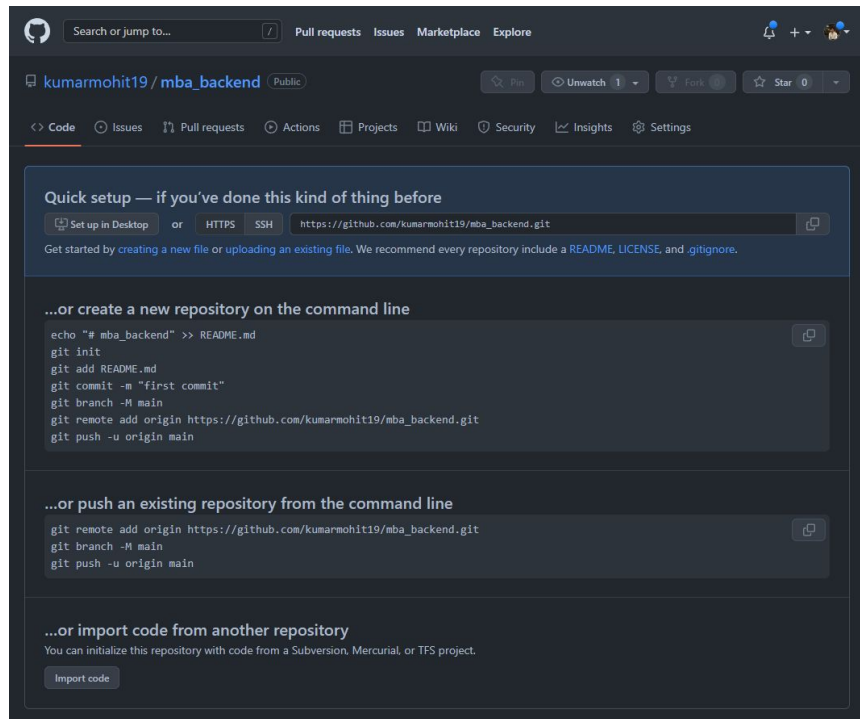
ⓘ You are creating a public repository in your personal account.

Create repository

4. Clone it to local, open cmd in the desired directory and write command

git clone <urlname>

5. If you have existing code in the local machine that you want to push to Github it can be done by pasting the folders/files in this cloned directory and we can commit and push the files to the remote repo, using command line.



2. Using the Command line:

- **Step1:** Create a directory in local or open cmd for the root directory of an existing project.
- **Step 2:** Type git init will make our current directory a local git repository.
- Note: Explain to students the difference between a local repo and remote repo,
- Local repo is the app code present in the git repo in local machine
- Remote repo is where collaborators push/merge their code

```
PS D:\Mohit\Relevel\MBA\session8\mba_backend> git init
Initialized empty Git repository in D:/Mohit/Relevel/MBA/session8/mba_backend/.git/
PS D:\Mohit\Relevel\MBA\session8\mba_backend> git add .
```

Step 3: We can add and commit the existing project files to the remote repository.

Step 4: Type `git add`. (to add all the files) or just `git add <file_name>` you want to push.

Step 5: Type `git commit -m <commit message>` where commit message is a string that informs why this commit happens and its purpose.

```
PS D:\Mohit\Relevel\MBA\session8\mba_backend> git commit -m "mba initial commit"
[master (root-commit) b2392ef] mba initial commit
34 files changed, 5793 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 configs/auth.config.js
create mode 100644 configs/db.config.js
create mode 100644 configs/server.config.js
create mode 100644 controllers/auth.controller.js
create mode 100644 controllers/booking.controller.js
create mode 100644 controllers/movie.controller.js
create mode 100644 controllers/payment.controller.js
create mode 100644 controllers/theatre.controller.js
create mode 100644 controllers/user.controller.js
create mode 100644 middlewares/authjwt.js
create mode 100644 middlewares/index.js
create mode 100644 middlewares/verifyBookingReqBody.js
create mode 100644 middlewares/verifyMovieReqBody.js
create mode 100644 middlewares/verifyPaymentReqBody.js
create mode 100644 middlewares/verifyTheatreReqBody.js
create mode 100644 middlewares/verifyUserReqBody.js
create mode 100644 models/booking.model.js
create mode 100644 models/movie.model.js
create mode 100644 models/payment.model.js
create mode 100644 models/theatre.model.js
create mode 100644 models/user.model.js
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 routes/auth.routes.js
create mode 100644 routes/booking.routes.js
create mode 100644 routes/movie.routes.js
create mode 100644 routes/payment.routes.js
create mode 100644 routes/theatre.routes.js
create mode 100644 routes/user.routes.js
create mode 100644 server.js
create mode 100644 utils/NotificationClient.js
create mode 100644 utils/constants.js
PS D:\Mohit\Relevel\MBA\session8\mba_backend> []
```

Step 6: Now we have to connect it to GitHub:

git remote add origin

[git@github.com:username/<repo_name>](https://github.com/username/<repo_name>)

```
PS D:\Mohit\Relevel\MBA\session8\mba_backend> git remote add origin https://github.com/kumarmohit19/mba_backend.git
PS D:\Mohit\Relevel\MBA\session8\mba_backend> 
```

Step 7: Finally, push using: git push origin master.

```
PS D:\Mohit\Relevel\MBA\session8\mba_backend> git push -u origin master
Enumerating objects: 42, done.
Counting objects: 100% (42/42), done.
Delta compression using up to 4 threads
Compressing objects: 100% (41/41), done.
Writing objects: 100% (42/42), 56.22 KiB | 1.76 MiB/s, done.
Total 42 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/kumarmohit19/mba_backend.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS D:\Mohit\Relevel\MBA\session8\mba_backend> 
```

Code: https://github.com/kumarmohit19/mba_backend

Get Hosted MongoDB database:

Why hosted DB

When we will be going to host our application on Heroku or any other platform it will be needing a database which will be live 24x7.

Now while developing the application, we used MongoDB in our local machine which was running in background and serving as database, which will not be working for application hosted on cloud service app, so we will be opting for a MongoDB cloud database hosting service here.

Steps to follow to get cloud MongoDB

First create an account on [MongoDB atlas](#) cloud database and then follow the steps.

Step 1: Go to the home page and then navigate to Projects menu on left hand side bar. Here, you will see all your project which may be empty unlike the below image. Now, click on “New Project” on right hand top.

The screenshot shows the MongoDB Cloud interface. The left sidebar contains the 'ORGANIZATION' menu with 'Projects' highlighted. The main content area is titled 'MOHIT'S ORG - 2021-03-02' and 'Projects'. A 'New Project' button is in the top right. Below a search bar, a table lists four projects: MERN, Project 0, test-project, and todo-td. Each project row shows '1 Database', '1 User', '0 Teams', and '0 Alerts', followed by action icons (three dots and a trash can).

Project Name	Databases	Users	Teams	Alerts	Actions
MERN	1 Database	1 User	0 Teams	0 Alerts	...
Project 0	1 Database	1 User	0 Teams	0 Alerts	...
test-project	1 Database	1 User	0 Teams	0 Alerts	...
todo-td	1 Database	1 User	0 Teams	0 Alerts	...

Step 2: Give a name to your project and click Next.

The screenshot shows the 'Create a Project' page in the Relevel dashboard. The top navigation bar includes a logo, a dropdown menu for 'Mohit's Org - 2021-0...', and links for 'Access Manager' and 'Billing'. On the right, there are links for 'All Clusters', 'Get Help', and a user profile 'Mohit'. The left sidebar lists navigation options: 'ORGANIZATION', 'Projects' (highlighted), 'Alerts', 'Activity Feed', 'Settings', 'Integrations', 'Access Manager', 'Billing', 'Support', and 'Live Migration'. The main content area is titled 'Create a Project' and features two tabs: 'Name Your Project' (active) and 'Add Members'. Below the tabs, the 'Name Your Project' section includes a note: 'Project names have to be unique within the organization (and other restrictions).' and a text input field containing 'mba_backend'. At the bottom right of the input field are 'Cancel' and 'Next' buttons.

Mohit's Org - 2021-0... Access Manager Billing All Clusters Get Help Mohit

ORGANIZATION

Projects

Alerts

Activity Feed

Settings

Integrations

Access Manager

Billing

Support

Live Migration

MOHIT'S ORG - 2021-03-02 > PROJECTS

Create a Project

Name Your Project Add Members

Name Your Project

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

mba_backend

Cancel Next

Step 3: Here, you do not need to change anything unless you want to provide access to this project to others. Click on Create Project.

ORGANIZATION

- Projects
- Alerts
- Activity Feed
- Settings
- Integrations
- Access Manager
- Billing
- Support
- Live Migration

MOHIT'S ORG - 2021-03-02 > 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.

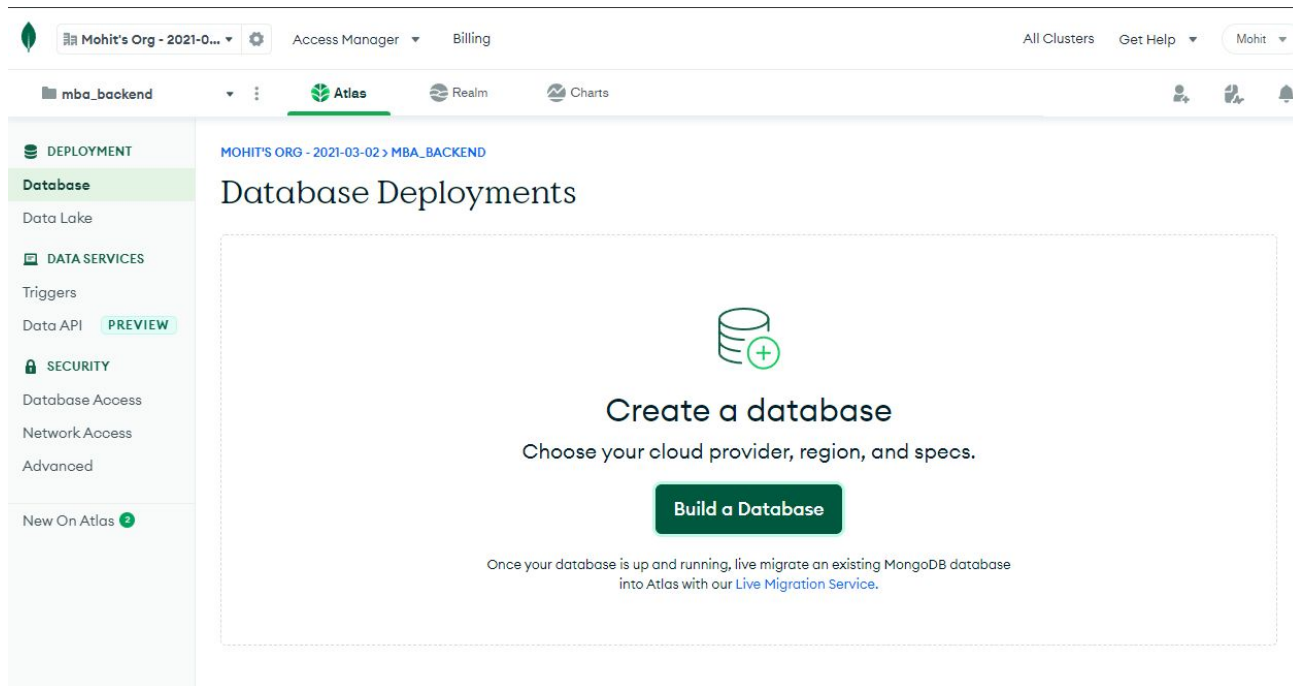
mohitnevatru@gmail.com (you) Project Owner

Cancel Go Back 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**
May only modify personal preferences

Step 4: Now, we will land on the database page where we need to create a new database, let's get started and click on Build a Database.



Step 5: Here, we will go with shared plan in which we will get the database with zero cost.

₹ INR

Deploy a cloud database

Experience the best of MongoDB on AWS, Azure, and Google Cloud. Choose a deployment option to get started.

PREVIEW

Serverless

For serverless applications that aren't critical with variable traffic. Minimal configuration required.

- ✓ Pay only for the operations you run
- ✓ Resources scale seamlessly to meet your workload
- ✓ Always-on security and backups

Create

Starting at
-₹23.13/1M reads

[I'll do this later](#)

ADVANCED

Dedicated

For production applications with sophisticated workload requirements. Advanced configuration controls.

- ✓ Network isolation and fine-grained access controls
- ✓ On-demand performance advice
- ✓ Multi-region and multi-cloud options available

Create

Starting at
-₹6.17/hr*

*estimated cost -₹4390.64/month

FREE

Shared

For learning and exploring MongoDB in a cloud environment. Basic configuration options.

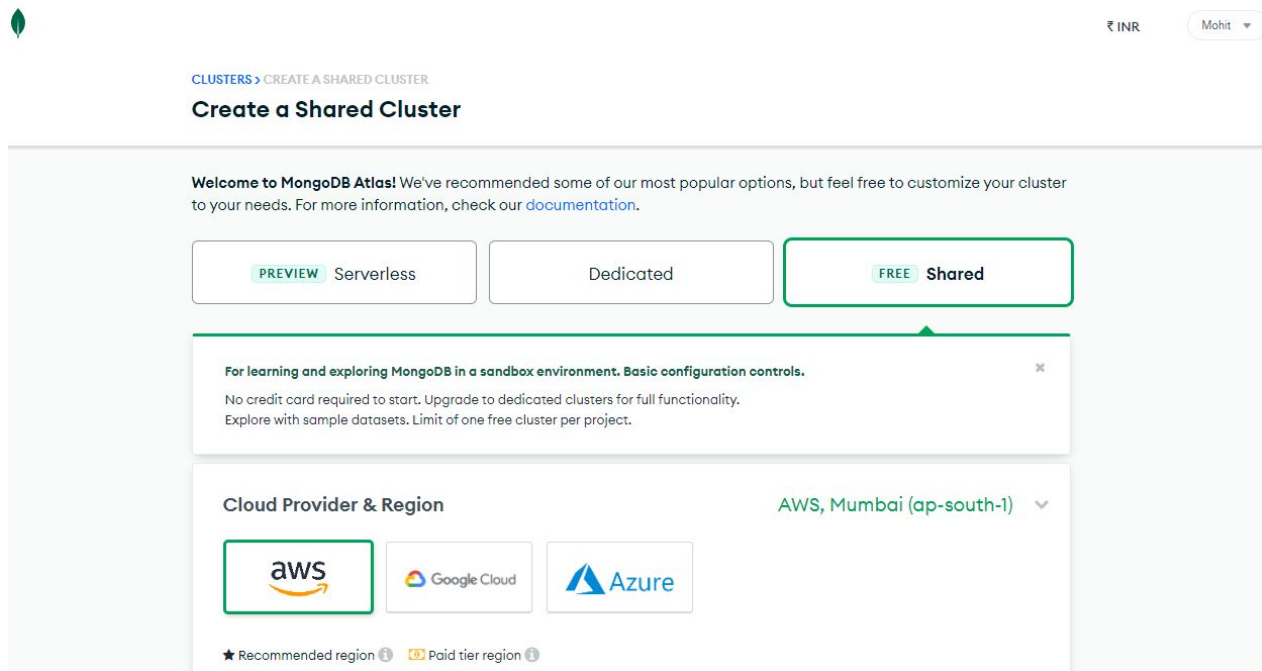
- ✓ No credit card required to start
- ✓ Explore with sample datasets
- ✓ Upgrade to dedicated clusters for full functionality

Create

Starting at
FREE

[Advanced Configuration Options](#)

Step 6: We will be redirected to next page where most of the things are selected, we just need to verify and move forward.



CLUSTERS > CREATE A SHARED CLUSTER

Create a Shared Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

PREVIEW Serverless Dedicated **FREE** Shared

For learning and exploring MongoDB in a sandbox environment. Basic configuration controls.
No credit card required to start. Upgrade to dedicated clusters for full functionality.
Explore with sample datasets. Limit of one free cluster per project.

Cloud Provider & Region AWS, Mumbai (ap-south-1)

aws Google Cloud Azure

★ Recommended region ⓘ 💰 Paid tier region ⓘ

Step 7: Cloud provider region should be nearest available server. Which is Mumbai here.

Cloud Provider & Region

AWS, Mumbai (ap-south-1) ▼

aws Google Cloud Azure

★ Recommended region ⓘ Paid tier region ⓘ

NORTH AMERICA	EUROPE	AUSTRALIA
N. Virginia (us-east-1) ★	Paris (eu-west-3) ★	Sydney (ap-southeast-2) ★
Oregon (us-west-2) ★	Ireland (eu-west-1) ★	ASIA
Ohio (us-east-2) ★ ⓘ	Frankfurt (eu-central-1) ★	Tokyo (ap-northeast-1)
N. California (us-west-1) ⓘ	Stockholm (eu-north-1) ★	Singapore (ap-southeast-1) ★
Montreal (ca-central-1) ⓘ	London (eu-west-2) ★ ⓘ	Mumbai (ap-south-1)
SOUTH AMERICA	Milan (eu-south-1) ★ ⓘ	Seoul (ap-northeast-2)
Sao Paulo (sa-east-1)	MIDDLE EAST	Hong Kong (ap-east-1) ★
	Bahrain (me-south-1) ★	Jakarta (ap-southeast-3) ★ ⓘ
	AFRICA	Osaka (ap-northeast-3) ★ ⓘ
	Cape Town (af-south-1) ★	

Step 8: Give a proper cluster name here, and then click on Create Cluster, which is basically a database.

Cluster Tier

M0 Sandbox (Shared RAM, 512 MB Storage)
Encrypted

Additional Settings

MongoDB 5.0, No Backup


Cluster Name

movie-booking-app

One time only: once your cluster is created, you won't be able to change its name.

movie-booking-app

Cluster names can only contain ASCII letters, numbers, and hyphens.

 Privacy · Terms


FREE

₹ INR

Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

[Back](#)

Create Cluster



Step 9: Now, we will be redirected to Quickstart page, where we have to create user as show below.

Mohit's Org - 2021-03-02

Access Manager Billing

All Clusters Get Help Mohit

mba_backend Atlas Realm Charts

DEPLOYMENT

Database

Data Lake

DATA SERVICES

Triggers

Data API PREVIEW

SECURITY

Quickstart

Database Access

Network Access

Advanced

New On Atlas

MOHIT'S ORG - 2021-03-02 > MBA_BACKEND

Security Quickstart

To access data stored in Atlas, you'll need to create users and set up network security controls. [Learn more about security setup](#)

1 How would you like to authenticate your connection?

Your first user will have permission to read and write any data in your project.

Username and Password Certificate

Create a database user using a username and password. Users will be given the *read and write to any database privilege* by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password.

Username

admin

Password

Welcome!

Autogenerate Secure Password Copy

Create User

No Cluster Provisioning... This process will take 3-5 minutes.

to connect from?

ADVANCED

Step 10: Next, we need to setup the server access, where we we like to connect from, here since we are deploying our app on Heroku we have to select Cloud Environment and then the Configure button of IP Access List tab.

The screenshot shows the MongoDB Atlas web interface. The top navigation bar includes the MongoDB logo, the organization name 'Mohit's Org - 2021-0...', and links for 'Access Manager', 'Billing', 'All Clusters', 'Get Help', and a user profile 'Mohit'. The left sidebar shows a navigation menu with sections: 'DEPLOYMENT' (Database, Data Lake), 'DATA SERVICES' (Triggers, Data API with a 'PREVIEW' badge), 'SECURITY' (Quickstart, Database Access, Network Access, Advanced), and 'New On Atlas' with a plus icon. The main content area is titled '2 Where would you like to connect from?' and includes the instruction 'Enable access for any network(s) that need to read and write data to your cluster.' Below this are two main options: 'My Local Environment' (with a 'Configure' button) and 'Cloud Environment' (marked as 'ADVANCED' with a 'Configure in New Tab' button). A section titled 'Set your network security with any of the following options' contains three cards: 'IP Access List' (with a 'Configure' button), 'VPC Peering' (with a 'Configure in New Tab' button), and 'Private Endpoint' (with a 'Configure in New Tab' button'). A 'Finish and Close' button is at the bottom right.

Step 11: We want to access it from anywhere for now, so we will be providing IP address 0.0.0.0/0 and the description as same.

The screenshot shows the MongoDB Atlas web interface. The top navigation bar includes the user's organization 'Mohit's Org - 2021-0...', 'Access Manager', and 'Billing'. The left sidebar lists navigation options: 'DEPLOYMENT' (Database, Data Lake), 'DATA SERVICES' (Triggers, Data API), 'SECURITY' (Quickstart, Database Access, Network Access, Advanced), and 'New On Atlas'. The main content area is titled '2 Where would you like to connect from?' and includes a sub-header 'Enable access for any network(s) that need to read and write data to your cluster.' There are two main options: 'My Local Environment' and 'Cloud Environment' (marked as 'ADVANCED'). Below these, a section titled 'Set your network security with any of the following options' contains a table for adding IP addresses and two boxes for 'VPC Peering' and 'Private Endpoint' configurations.

IP Address	Description	Action
0.0.0.0/0	allows access from anyw	Add Entry

Buttons: 'Add My Current IP Address', 'Configure In New Tab' (for VPC Peering and Private Endpoint).

Step 12: Click on Finish and Close

The screenshot shows the MongoDB Atlas 'Security' configuration page. The left sidebar contains navigation links for 'DEPLOYMENT', 'DATA SERVICES', and 'SECURITY'. Under 'SECURITY', 'Quickstart' is selected. The main content area is titled 'Set your network security with any of the following options'. It includes a section for 'IP Access Lists' with a table showing an entry for '0.0.0.0/0' with the description 'allows access from anywhere'. Below this are two cards for 'VPC Peering' and 'Private Endpoint'. At the bottom right, there is a green 'Finish and Close' button.

DEPLOYMENT

- Database
- Data Lake

DATA SERVICES

- Triggers
- Data API **PREVIEW**

SECURITY

- Quickstart**
- Database Access
- Network Access
- Advanced

New On Atlas

Set your network security with any of the following options

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the [Network Access Page](#).

IP Address	Description	
<input type="text" value="Enter IP Address"/>	<input type="text" value="Enter description"/>	<button>Add Entry</button> <button>Add My Current IP Address</button>

IP Access List	Description	
0.0.0.0/0	allows access from anywhere	<button>REMOVE</button>

VPC Peering

Peer your VPC with your Atlas cluster's VPC to ensure that traffic does not traverse the public internet. Requires an M10 cluster or higher.

Configure In New Tab

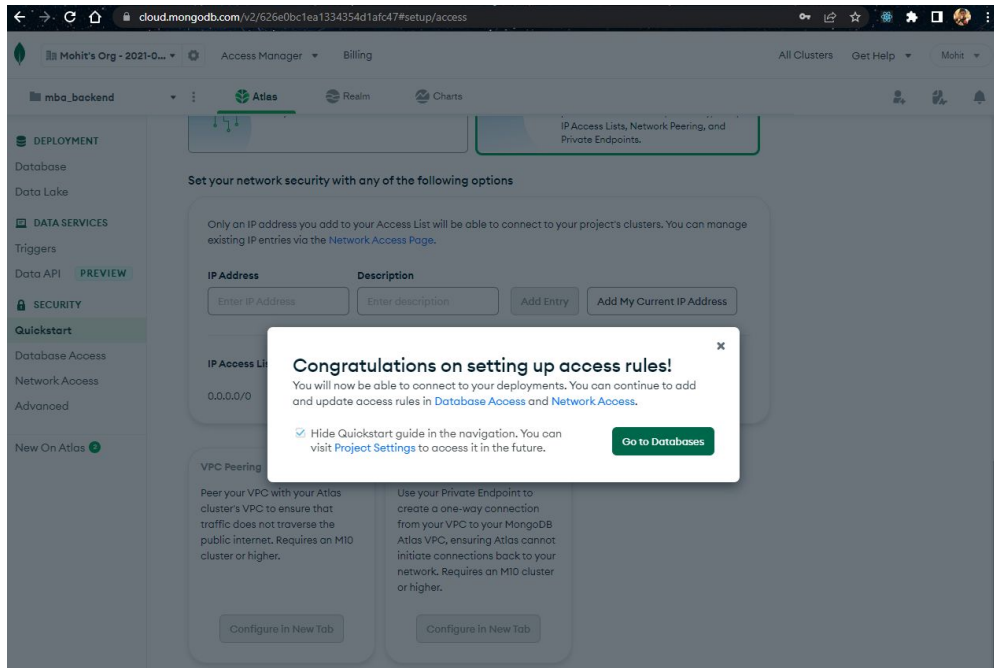
Private Endpoint

Use your Private Endpoint to create a one-way connection from your VPC to your MongoDB Atlas VPC, ensuring Atlas cannot initiate connections back to your network. Requires an M10 cluster or higher.

Configure In New Tab

Finish and Close

Step 13: Once step 12 is successful we will see the below prompt. Here, click on Go to Database button.

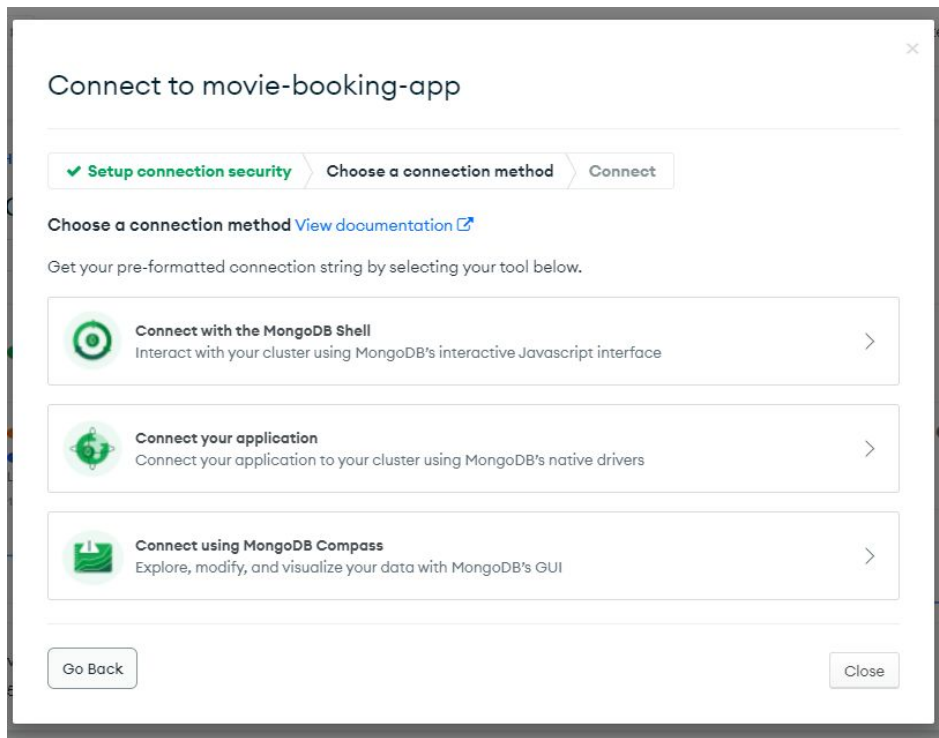


Step 14: This is how the database page will look, now let's use this database and connect to it with our express app. Click on Connect button.

The screenshot shows the Relevel Atlas interface for a database deployment named 'movie-booking-app'. The left sidebar contains navigation links for DEPLOYMENT, Database, Data Lake, DATA SERVICES, Triggers, Data API (PREVIEW), SECURITY, Quickstart, Database Access, Network Access, Advanced, and New On Atlas. The main content area is titled 'Database Deployments' and includes a search bar, a '+ Create' button, and a 'Connect' button. Below these are four charts: Connections (0), In/Out B/s (0.0 B/s), Data Size (0.0 B), and an 'Enhance Your Experience' section with an 'Upgrade' button. At the bottom, a table lists deployment details.

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED REALM APP	ATLAS SEARCH
5.0.0	AWS / Mumbai (ap-south-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Create Index

Step 15: Select Connect your application option as we want to connect to the DB using our node application.



Step 16: Here, you will see the connection string and proper notes to replace the <password> and myFirstDatabase string with actual ones.

×

Connect to movie-booking-app

✓ Setup connection security

✓ Choose a connection method

Connect

1

Select your driver and version

DRIVER

Node.js

VERSION

4.0 or later

2

Add your connection string into your application code

☐ Include full driver code example

mongodb+srv://admin:<password>@movie-booking-app.yjxuy.mongodb.net/myFirstDatabase?retryWrites=true&w=majority

📋

Replace <password> with the password for the **admin** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

Go Back

Close

Step 17: Adding production database URL into our app

```
configs > db.config.js > <unknown> > PRODUCTION_DB_URL
1  module.exports = {
2    DB_NAME: "mba_db",
3    DB_URL: "mongodb://localhost/mba_db",
4    PRODUCTION_DB_URL:
5      "mongodb+srv://admin:Welcome1@movie-booking-app.yjxuy.mongodb.net/movie-booking-app?
6      retryWrites=true&w=majority&ssl=true",
7  };
```

Step 18: Modify server.js to use production database for production environment:

```
18  /**
19   * DB Connection initialization
20   */
21  if (process.env.NODE_ENV === "production") {
22    try {
23      console.log(process.env.NODE_ENV);
24      // Connect to the MongoDB cluster
25      mongoose.connect(
26        dbConfig.PRODUCTION_DB_URL,
27        { useNewUrlParser: true, useUnifiedTopology: true },
28        () => {
29          console.log("Connected to production Mongo DB ");
30          init();
31        }
32      );
33    } catch (err) {
34      console.log("Could not connect to the datababse: " + err.message);
35    }
36  } else {
37    console.log(process.env.NODE_ENV);
38    mongoose.connect(
39      dbConfig.DB_URL,
40      () => {
41        console.log("Connected to Mongo DB ");
42        init();
43      },
44      (err) => {
45        console.log("Error :", err.message);
46      }
47    );
48  }
49
```

Intro to Heroku

Heroku is a cloud platform that helps developers maintain, build, deliver, and scale apps.

Deploy App:

Let us use our application on Heroku, but before that, we have to do some steps:


Create an account on Heroku, (nothing much here just fill the prompted details and verify account for specified email id).

1. Download Heroku CLI:

<https://devcenter.heroku.com/articles/heroku-cli#download-and-install>

- [Git installation](#)
- [First-time Git setup](#)

Install with an Installer

 The Windows installers display a warning titled "Windows protected your PC" to some users. To run the installation when this warning shows, click "More info", verify the publisher as "salesforce.com, inc", then click the "Run anyway" button.

macOS

```
$ brew tap heroku/brew && brew install heroku
```

Windows

Download the appropriate installer for your Windows installation:

[64-bit installer](#)

[32-bit installer](#)

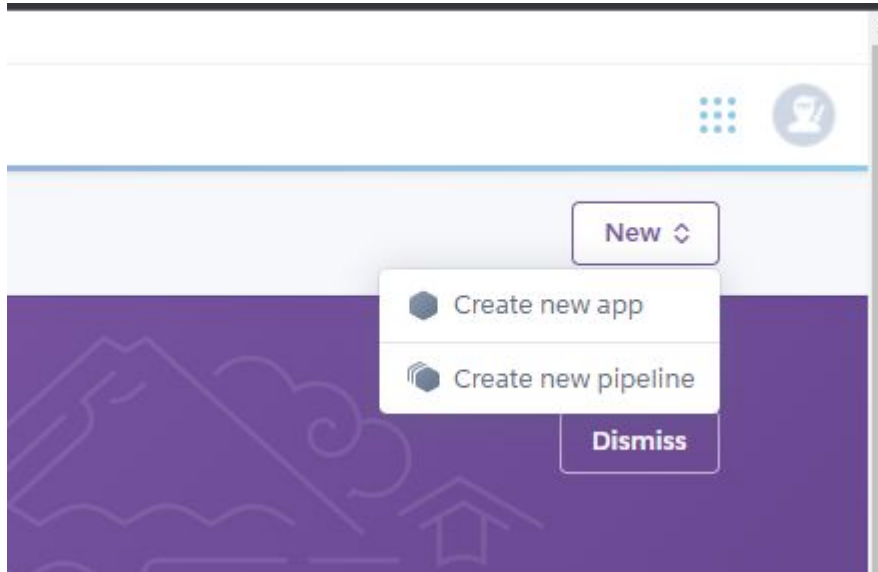
Ubuntu 16+

Run the following from your terminal:

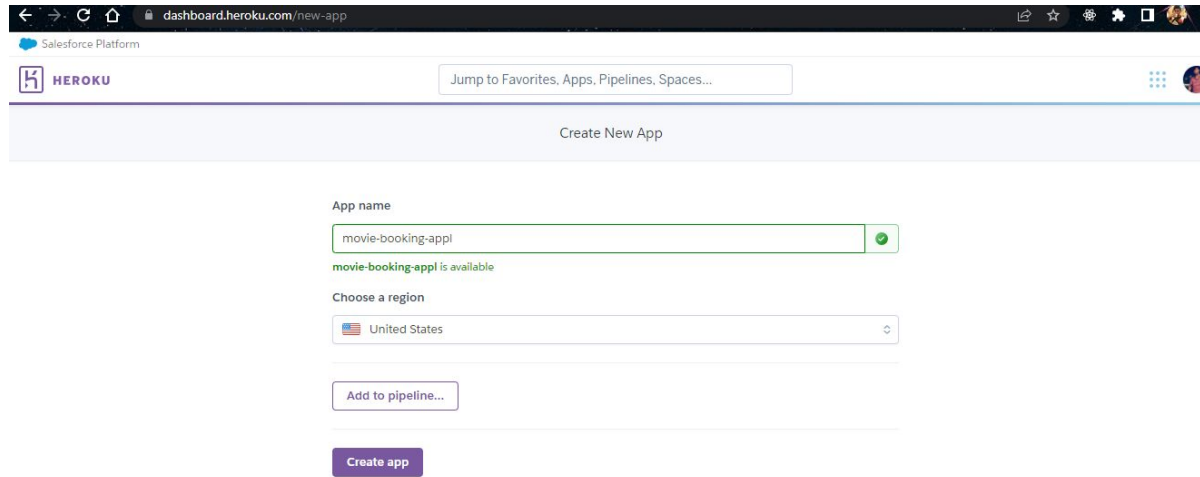
```
$ sudo snap install --classic heroku
```

Snap is available on other Linux OS's as well.

2. Go to the Heroku website <https://dashboard.heroku.com/apps> and log in. After login, click on “New.” and click “Create new app”.



3. The name must be unique, and click on Create app.



The screenshot shows the Heroku dashboard at `dashboard.heroku.com/new-app`. The page has a header with the Heroku logo and a search bar. Below the header is a large button labeled "Create New App". The form below the button includes:

- App name:** A text input field containing "movie-booking-appl" with a green checkmark icon to its right. Below the field, a green message states "movie-booking-appl is available".
- Choose a region:** A dropdown menu showing "United States" with a US flag icon.
- Add to pipeline...** A button.
- Create app** A purple button.

4. Now, we have to install [Heroku CLI](#) to proceed with the next steps:

The screenshot shows the Salesforce Platform Heroku interface. At the top, there's a navigation bar with the Salesforce Platform logo and the Heroku logo. Below the navigation bar, there's a search bar with the text "Jump to Favorites, Apps, Pipelines, Spaces...". The main content area is divided into several sections:

- Add this app to a pipeline:** This section includes a sub-section "Create a new pipeline or choose an existing one and add this app to a stage in it." and a "Choose a pipeline" dropdown menu.
- Add this app to a stage in a pipeline to enable additional features:** This section includes two sub-sections: "Pipelines let you connect multiple apps together and promote code between them." and "Pipelines connected to GitHub can enable review apps, and create apps for new pull requests." Both sub-sections include a "Learn more" link.
- Deployment method:** This section includes three options: "Heroku Git Use Heroku CLI", "GitHub Connect to GitHub", and "Container Registry Use Heroku CLI".
- Deploy using Heroku Git:** This section includes a sub-section "Install the Heroku CLI" with instructions to download and install the Heroku CLI, and a sub-section "Clone the repository" with instructions to use Git to clone the moviebookingapp's source code to your local machine. Both sub-sections include a "Deploy your changes" sub-section with instructions to make some changes to the code you just cloned and deploy them to Heroku using Git.

```
$ heroku login
```

```
$ heroku git:clone -a moviebookingapp1
$ cd moviebookingapp1
```

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

5. After successful installation of Heroku, go to you project root directory and open command prompt and do Heroku login a shown below.

```
PS D:\Mohit\Relevel\MBA\session8\mba_backend> heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/691fe342-78b6-47d3-b6aa-9928a52f8723
?requestor=SFMyNTY.g2gDbQAAAA8xMjIuMTc2LjE5MC4xMTZuBgDY_7h-gAFiAAFRgA.pgsDctnQi-ySJg58upBgI5haAoQbMr
SGNPwrK0s_pUQ
Logging in... done
Logged in as mohitnevtru@gmail.com
PS D:\Mohit\Relevel\MBA\session8\mba_backend>
```

6. After successful login we will assign another remote repo for Heroku using command:

```
heroku git:remote -a
```

```
<application-name-you-set-on-heroku>
```

```
PS D:\Mohit\Relevel\MBA\session8\mba_backend> heroku git:remote -a moviebookingappl
set git remote heroku to https://git.heroku.com/moviebookingappl.git
PS D:\Mohit\Relevel\MBA\session8\mba_backend> git remote -v
heroku https://git.heroku.com/moviebookingappl.git (fetch)
heroku https://git.heroku.com/moviebookingappl.git (push)
origin https://github.com/kumarmohit19/mba_backend.git (fetch)
origin https://github.com/kumarmohit19/mba_backend.git (push)
```

7. Now, run the push command to trigger the deployment.
git push heroku master

```
PS D:\Mohit\Relevel\MBA\session8\mba_backend> git push heroku master
Enumerating objects: 42, done.
Counting objects: 100% (42/42), done.
Delta compression using up to 4 threads
Compressing objects: 100% (41/41), done.
Writing objects: 100% (42/42), 56.22 KiB | 1.94 MiB/s, done.
Total 42 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_VERBOSE=false
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
remote:
remote: -----> Installing binaries
remote:       engines.node (package.json):  unspecified
remote:       engines.npm (package.json):   unspecified (use default)
remote:
remote:       Resolving node version 16.x...
remote:       Downloading and installing node 16.15.0...
remote:       Using default npm version: 8.5.5
remote:
remote: -----> Installing dependencies
remote:       Installing node modules
remote:
remote:       added 216 packages, and audited 217 packages in 5s
remote:
remote:       23 packages are looking for funding
remote:       run `npm fund` for details
remote:
remote:       found 0 vulnerabilities
remote:
```

```
remote: Resolving node version 16.x...
remote: Downloading and installing node 16.15.0...
remote: Using default npm version: 8.5.5
remote:
remote: -----> Installing dependencies
remote: Installing node modules
remote:
remote: added 216 packages, and audited 217 packages in 5s
remote:
remote: 23 packages are looking for funding
remote:   run `npm fund` for details
remote:
remote: found 0 vulnerabilities
remote:
remote: -----> Build
remote:
remote: -----> Caching build
remote:   - node_modules
remote:
remote: -----> Pruning devDependencies
remote:
remote: up to date, audited 102 packages in 623ms
remote:
remote: 7 packages are looking for funding
remote:   run `npm fund` for details
remote:
remote: found 0 vulnerabilities
remote:
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote: Procfile declares types   -> (none)
remote: Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote: Done: 34.9M
remote: -----> Launching...
remote: Released v3
remote: https://moviebookingappl.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/moviebookingappl.git
* [new branch] master -> master
```

Once you see the Build succeeded! Message your deployment will be successful and can access the application on below given URL.

Deployed code link: <https://moviebookingappl.herokuapp.com/>

Testing our hosted App

SignUp: POST /auth/signup route

The screenshot displays a REST client interface for testing the `POST /auth/signup` endpoint. The URL bar shows `https://moviebookingappl.herokuapp.com/mba/api/v1/auth/signup`. The request body is a JSON object with the following fields: `name`, `userId`, `email`, `userType`, and `password`. The response is also in JSON format, showing a successful status of 201 (Created) with additional fields like `userStatus`, `createdAt`, and `updatedAt`.

Request:

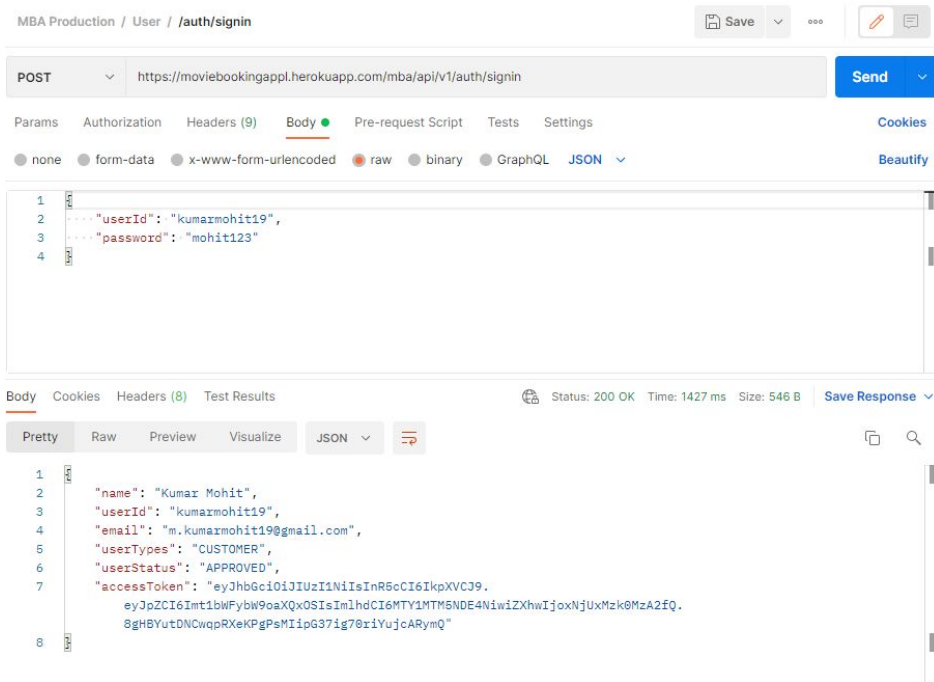
```
1 {
2   "name": "Kumar Mohit",
3   "userId": "kumarmohit19",
4   "email": "m.kumarmohit19@gmail.com",
5   "userType": "CUSTOMER",
6   "password": "mohit123"
7 }
```

Response:

```
1 {
2   "name": "Kumar Mohit",
3   "userId": "kumarmohit19",
4   "email": "m.kumarmohit19@gmail.com",
5   "userTypes": "CUSTOMER",
6   "userStatus": "APPROVED",
7   "createdAt": "2022-05-01T08:28:45.329Z",
8   "updatedAt": "2022-05-01T08:28:45.329Z"
9 }
```

SignIn: POST /auth/signin route

SignIn: POST /auth/signin route



Testing our hosted App

Using access token, GET /movies route

MBA Production / Movies / /movies

Save Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Headers 7 hidden

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZi...			
Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 452 ms Size: 2.03 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "626e44aed45eeb4ebce22c2",
3   "name": "Bachhan Pandey",
4   "description": "Comedy Masala Movie",
5   "casts": [
6     "Akshay Kumaz",
7     "Jacqueline Fernandez"
8   ],
9   "trailerUrl": "http://bachhanpandey/trailers/1",
10  "posterUrl": "http://bachhanpandey/posters/1",
11  "language": "Hindi",
12  "releaseDate": "18-03-2022",
13  "director": "Fazhad Samji",
14  "releaseStatus": "RELEASED",
15  "updatedAt": "2022-05-01T06:28:30.964Z",
16  "__v": 0
17 },
18 {
19   "_id": "626e44afid45eeb4ebce22c4",
20   "name": "7-1-1"
```

Testing our hosted App

And, GET /theatres route

MBA Production / Theatres / /theatres

GET https://moviebookingapp.herokuapp.com/mba/api/v1/theatres

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Headers 7 hidden

KEY	VALUE	DESCRIPTION		Bulk Edit	Presets
<input checked="" type="checkbox"/> x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZiZ...				
Key	Value	Description			

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 481 ms Size: 214 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "626e44b0d45eeb4ebce22cd",
3   "name": "FunCinemas",
4   "description": "Top class theatre",
5   "city": "Bangalore",
6   "pinCode": 560052,
7   "movies": [
8     "626e44aed45eeb4ebce22c2",
9     "626e44af045eeb4ebce22c4",
10    "626e44af045eeb4ebce22c6"
11  ],
12   "ownerId": "626e44aed45eeb4ebce22ba",
13   "createdAt": "2022-05-01T08:28:32.167Z",
14   "updatedAt": "2022-05-01T08:28:32.167Z",
15   "__v": 0
16 },
17 {
18   "_id": "626e44b0d45eeb4ebce22cf",
19   "name": "PVR Cinemas - Kormangala",
20   "description": "PVR Cinemas - Kormangala"
```

MCQ's:

1. What is the meaning of origin in git command?

- A. Origin is the name of the branch.
- B. It's the name of the repo.
- C. It signifies a remote name where the user wants to push or pull the changes.
- D. None of the above.

2. What does a checkout argument do in git command?

- A. Checkout means to go out of the repo.
- B. It means the act of switching between different versions of a target entity or the branches of the repository.
- C. It is used to commit the change and push in one command.
- D. None of the above.

3. What Does a Heroku app URL look like with the app name “alpha”?

- A. www.herokuapp.com/alpha
- B. alpha.heroku.com
- C. alpha.herokuapp.com
- D. www.alpha.herokuapp.com

4. Is it possible to change the domain of the application deployed on Heroku.

- A. No.
- B. Yes

5. Which branch is used for deployment from GitHub to Heroku?

- A. master
- B. We can choose.
- C. origin
- D. None of the above.

Thank You!