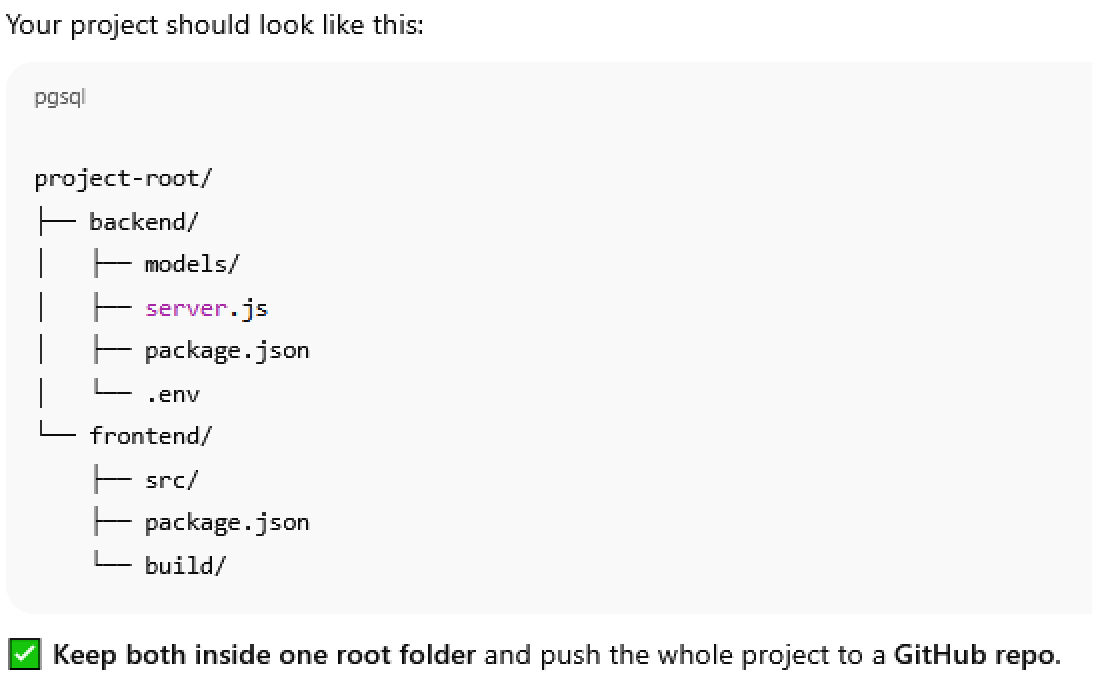
Application Deployment using RENDER Platform:

- **React** → frontend
- **Express.js** → backend middleware
- **MongoDB Atlas** → cloud database

Below is a **complete project structure and working code** you can copy-paste to run.

Project code Github URL: https://github.com/lakshminarayana-kodavali/MERN-KLN-PROJECT.git



Your project should look like this:

```pgsql
project-root/
├── backend/
│   ├── models/
│   ├── server.js
│   ├── package.json
│   └── .env
└── frontend/
    ├── src/
    ├── package.json
    └── build/
```

✅ **Keep both inside one root folder** and push the whole project to a **GitHub repo.**

NOTE: Build folder in frontend project will create automatically during deployment.

Modify Your Backend for Deployment

backend/server.js

Replace your existing code with this modified version
(You only need to update this one file.)

==

```
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
const dotenv = require("dotenv");
const bcrypt = require("bcryptjs");
const path = require("path");
const User = require("./models/User");

dotenv.config();
```

```
const app = express();

app.use(cors());
app.use(express.json());

// ✅ Connect MongoDB
mongoose
  .connect(process.env.MONGO_URI)
  .then(() => console.log("✅ MongoDB Connected"))
  .catch((err) => console.log(err));

// ✅ Signup Route
app.post("/signup", async (req, res) => {
  const { name, email, password } = req.body;
  const userExists = await User.findOne({ email });
  if (userExists) return res.status(400).json({ msg: "User already exists" });

  const hashedPassword = await bcrypt.hash(password, 10);
  const user = new User({ name, email, password: hashedPassword });
  await user.save();
  res.json({ msg: "User registered successfully" });
});

// ✅ Login Route
app.post("/login", async (req, res) => {
  const { email, password } = req.body;
  const user = await User.findOne({ email });
  if (!user) return res.status(400).json({ msg: "User not found" });

  const isMatch = await bcrypt.compare(password, user.password);
  if (!isMatch) return res.status(400).json({ msg: "Invalid credentials" });

  res.json({ msg: "Login successful", user });
});

// ✅ Serve frontend build files after building React app
app.use(express.static(path.join(__dirname, "../frontend/build")));

app.get("*", (req, res) => {
  res.sendFile(path.join(__dirname, "../frontend/build", "index.html"));
});

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`🚀 Server running on port ${PORT}`));

==================================================================
```

Modify Frontend API Base URL

In your `frontend/src/api.js` file, replace entire code like this:

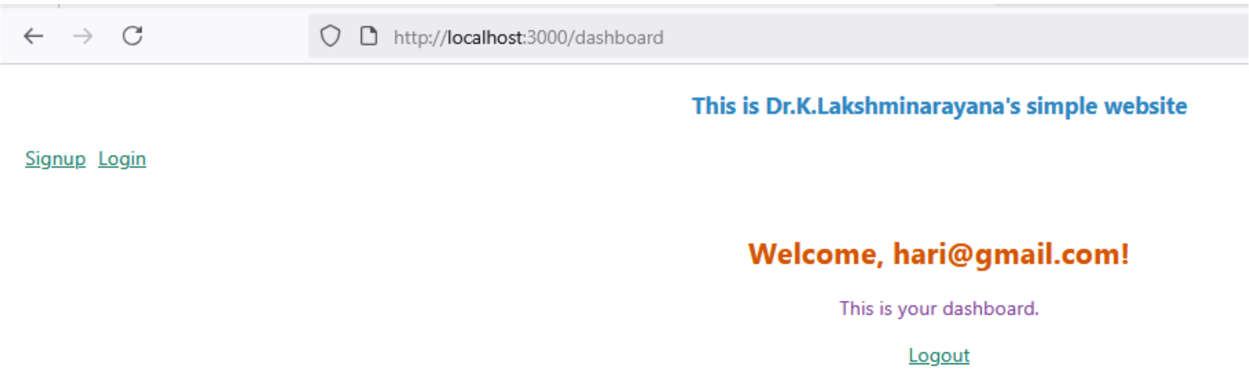
```
import axios from 'axios';

const api = axios.create({
  baseURL: "/", // ✅ relative URL — works for both local & Render
});

export default api;
```

Add Build Script in Backend

In your **backend/package.json**, update this section, with build command:

```
"scripts": {
  "start": "node server.js",
  "dev": "nodemon server.js",
  "build": "cd ../frontend && npm install && npm run build"
}
```

Local output:



This is Dr.K.Lakshminarayana's simple website

Signup Login

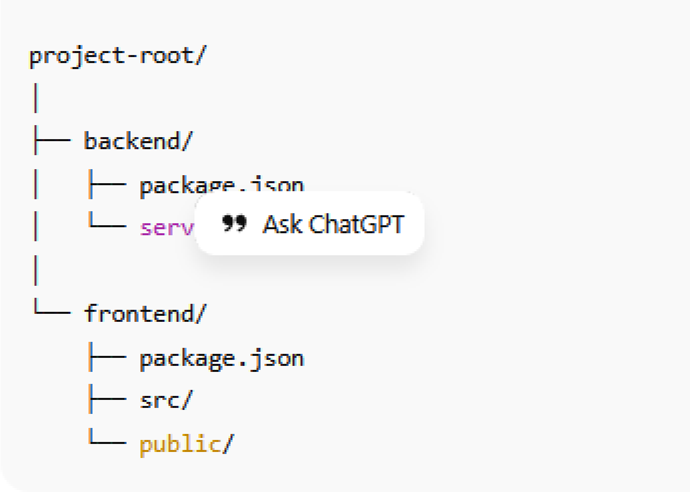
Welcome, hari@gmail.com!

This is your dashboard.

Logout

Push to GitHub

Make sure your folder structure is like this:

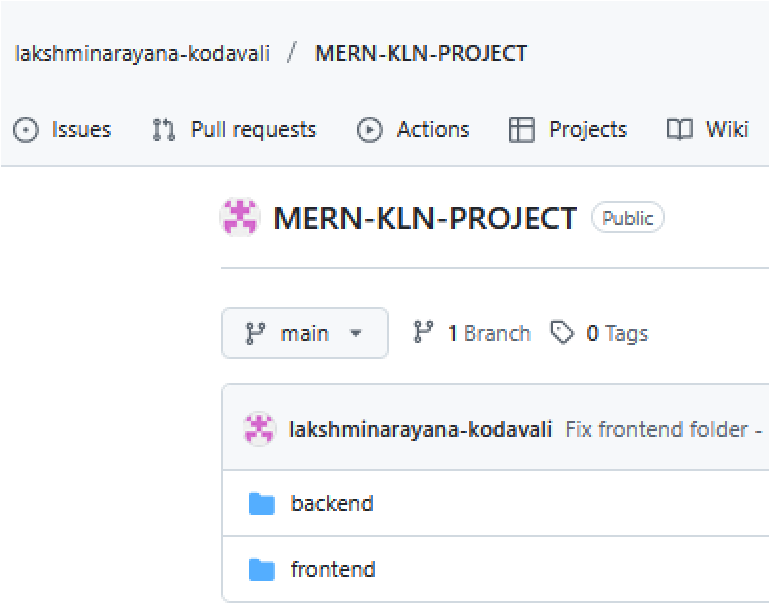


```
project-root/
│
├── backend/
│   ├── package.json
│   └── serv
│
└── frontend/
    ├── package.json
    ├── src/
    └── public/
```

Then push everything to a new GitHub repository.

Push Commands

```
git init
git add .
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/lakshminarayana-kodavali/MERN-KLN-
PROJECT.git
git push -u origin main
```



lakshminarayana-kodavali / MERN-KLN-PROJECT

MERN-KLN-PROJECT Public

main · 1 Branch · 0 Tags

lakshminarayana-kodavali Fix frontend folder -

backend

frontend

Deploy Process on Render

1. Go to https://render.com
2. • Log in with GitHub.
3. • Click **"New Web Service"** → **Connect your repo**.
4. • In setup page, set these options:
5. **Setting Value**

Root Directory	backend

Build Command	`npm install && npm run build`
Start Command	`node server.js`
Environment Variables	(add below)

Add environment variables:

```
MONGO_URI = your-mongodb-atlas-uri
JWT_SECRET = anysecretkey
PORT = 5000
```

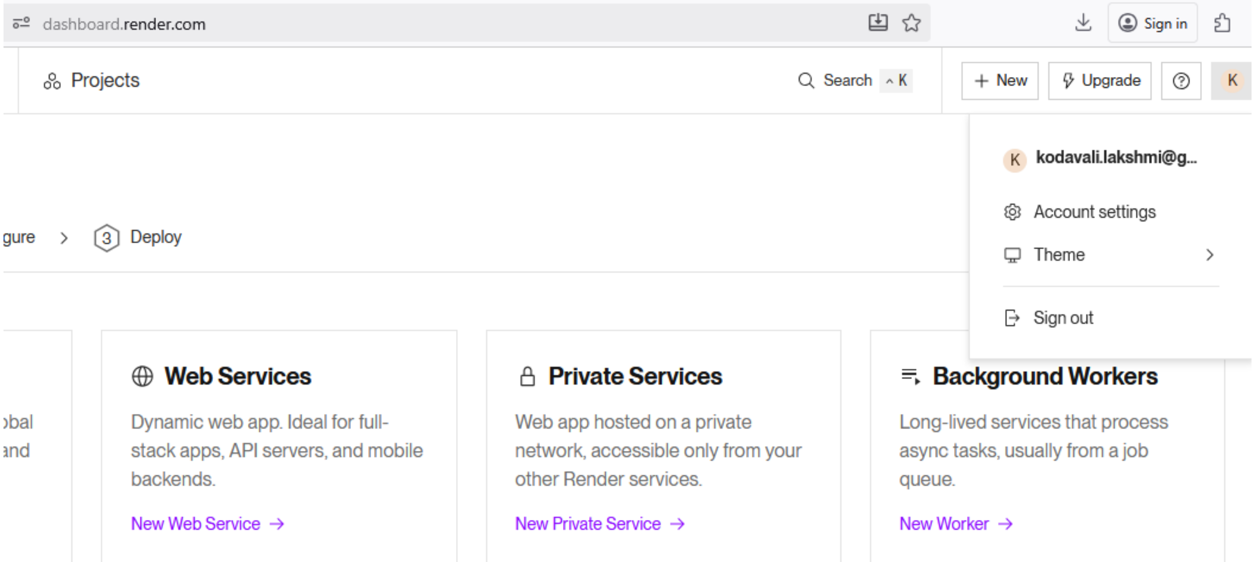
Click **Deploy Web Service**.

Render will:

- install dependencies,
- build your React app,
- start your Express server,
- serve React from `/frontend/build`.

Update Later (Auto Deployment)

Login into Render for deployment:

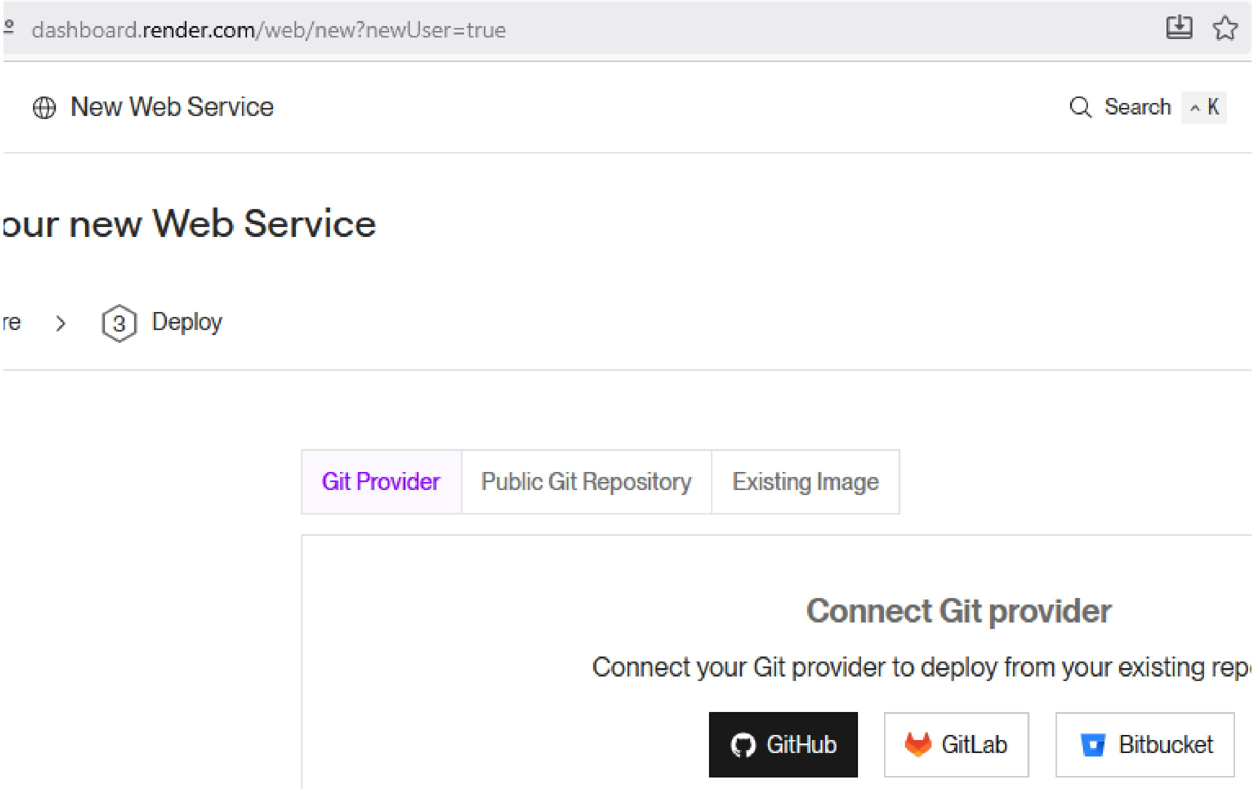


Web Services: Dynamic web app. Ideal for full-stack apps, API servers, and mobile backends. New Web Service →

Private Services: Web app hosted on a private network, accessible only from your other Render services. New Private Service →

Background Workers: Long-lived services that process async tasks, usually from a job queue. New Worker →

Click on “New Web Service”



dashboard.render.com/web/new?newUser=true

New Web Service

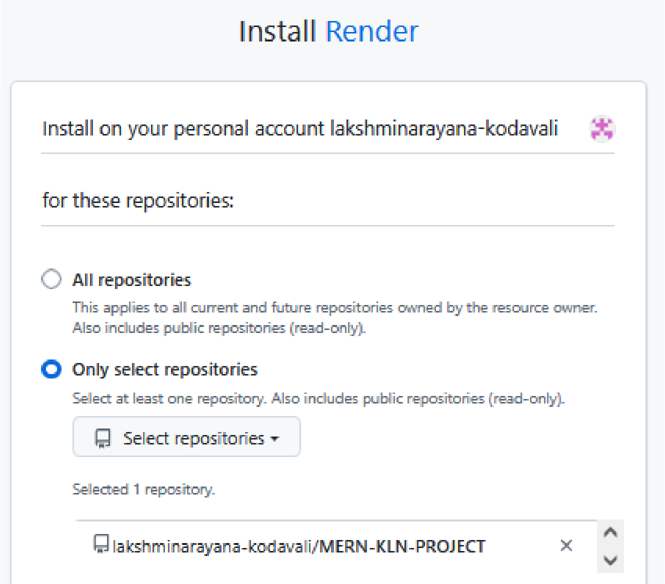
our new Web Service

re › 3 Deploy

Connect Git provider

Connect your Git provider to deploy from your existing rep

Connect with HitHub and Select the project repository



Install Render

Install on your personal account lakshminarayana-kodavali

for these repositories:

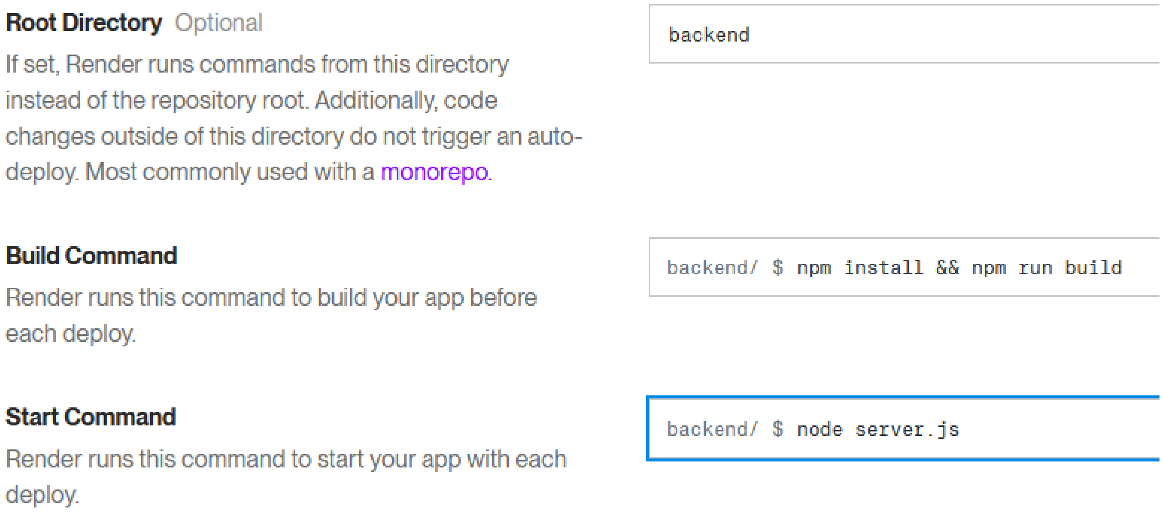
○ All repositories
This applies to all current and future repositories owned by the resource owner. Also includes public repositories (read-only).

● Only select repositories
Select at least one repository. Also includes public repositories (read-only).

Selected 1 repository.

lakshminarayana-kodavali/MERN-KLN-PROJECT ×

Setting Values



Root Directory Optional

If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a monorepo.

```
backend
```

Build Command

Render runs this command to build your app before each deploy.

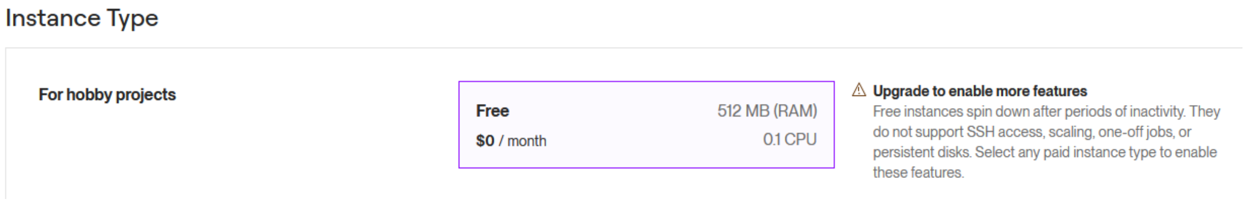
```
backend/ $ npm install && npm run build
```

Start Command

Render runs this command to start your app with each deploy.

```
backend/ $ node server.js
```

Select Instance type: FREE



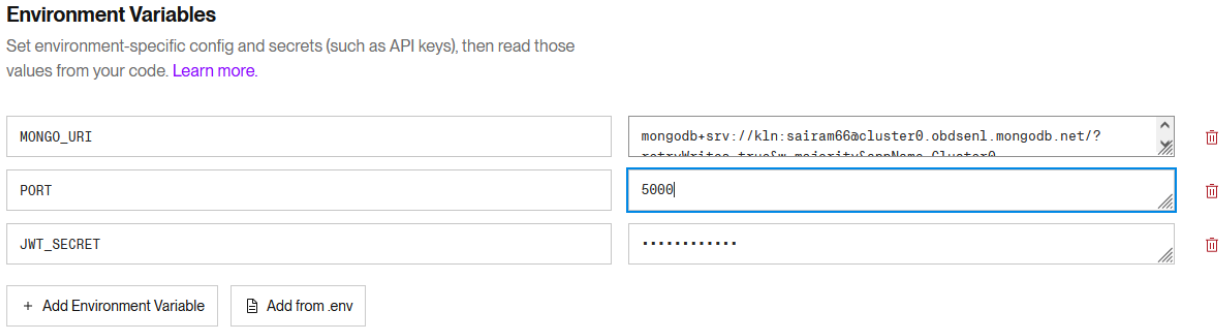
Instance Type

For hobby projects

Free 512 MB (RAM)
$0 / month 0.1 CPU

⚠ **Upgrade to enable more features**
Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.

Add environment variables:

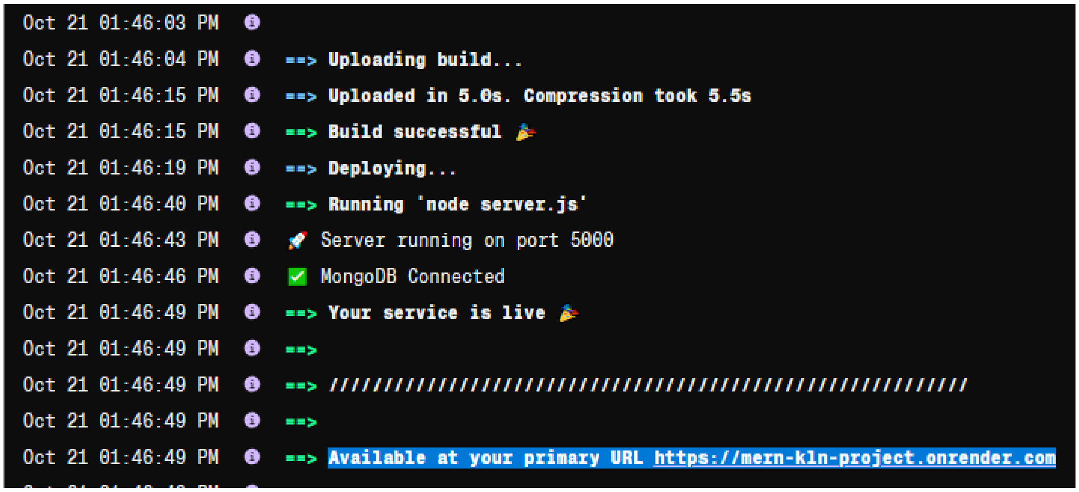


Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. Learn more.

MONGO_URI	mongodb+srv://kln:sairam66@cluster0.obdsenl.mongodb.net/? retryWrites=true&w=majority&appName=Cluster0
PORT	5000
JWT_SECRET	•••••••••••••

Click on “Deploy Web Service”

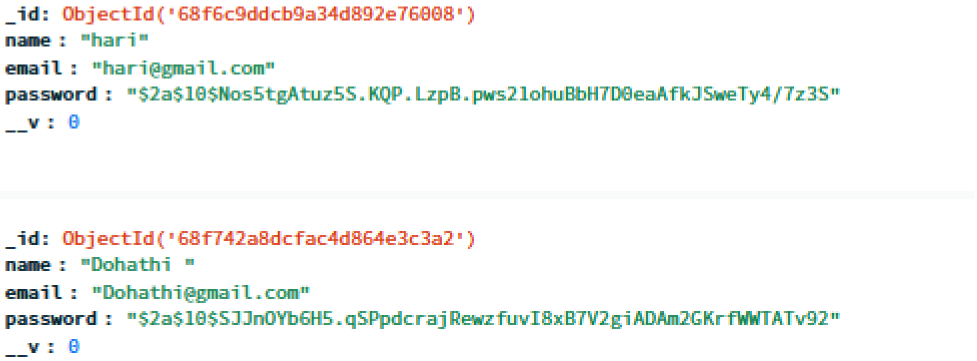


```
Oct 21 01:46:03 PM  ℹ
Oct 21 01:46:04 PM  ℹ  ==> Uploading build...
Oct 21 01:46:15 PM  ℹ  ==> Uploaded in 5.0s. Compression took 5.5s
Oct 21 01:46:15 PM  ℹ  ==> Build successful 🎉
Oct 21 01:46:19 PM  ℹ  ==> Deploying...
Oct 21 01:46:40 PM  ℹ  ==> Running 'node server.js'
Oct 21 01:46:43 PM  ℹ  🚀 Server running on port 5000
Oct 21 01:46:46 PM  ℹ  ✅ MongoDB Connected
Oct 21 01:46:49 PM  ℹ  ==> Your service is live 🎉
Oct 21 01:46:49 PM  ℹ  ==>
Oct 21 01:46:49 PM  ℹ  ==> ///////////////////////////////////////////////////////////
Oct 21 01:46:49 PM  ℹ  ==>
Oct 21 01:46:49 PM  ℹ  ==> Available at your primary URL https://mern-kln-project.onrender.com
```

Deployment link is Available at your primary URL: https://mern-kln-project.onrender.com

I accessed the above project through my mobile and signup and signin with one record (Dohathi) successfully.

Login into mongodb atlas Cloud to check my database update https://www.mongodb.com/cloud/atlas



```
_id: ObjectId('68f6c9ddcb9a34d892e76008')
name : "hari"
email : "hari@gmail.com"
password : "$2a$10$Nos5tgAtuz5S.KQP.LzpB.pws2lohuBbH7D0eaAfkJSweTy4/7z3S"
__v : 0
```

```
_id: ObjectId('68f742a8dcfac4d864e3c3a2')
name : "Dohathi "
email : "Dohathi@gmail.com"
password : "$2a$10$SJJnOYb6H5.qSPpdcrajRewzfuvI8xB7V2giADAm2GKrfWWTATv92"
__v : 0
```

Database successfully updated with Dohathi record.