Backend Configuration[Both dashtar and kachabaza]

Installation : To run react project you will need couple thing to setup first, Install node version v16.5.0, if you already not installed.

NPM Packages:

- After downloading the template, unzip it
- Open the Terminal.
- Go to the folder (backend): (cd path/to/template)
- Run: npm install, it will install all used packages for this backend.
- Make sure that there is no error.

After the npm packages are installed, make sure that you have a package.json file and check that you have the below lines under the scripts.

```
"scripts": {
  "dev": "nodemon api/index.js",
  "start": "node api/index.js",
  "production": "NODE_ENV=production nodemon api/index.js",
  "test": "echo \"Error: no test specified\" && exit 1",
  "data:import": "node script/seed.js",
  "product": "node script/product.js"
},
```

1. Rename .env.example to .env (use .env not .env.local it's only for backend).

2. Configure your MongoDB database, watch this video MongoDB, after configuring you will find a mongo URI just put that on your .env file MONGO_URI variable.

3. The JWT_SECRET is just a random value for creating a secret token, you can use whatever you want but make sure it is secret.

4. You need an email and password for using email verification and forget the password option. Use an email that you want to send messages to others when they register or request to forget the password. We used Nodemailer and the default email server for this. watch this video to create an app password for email app-password. After that put your email in the .env file EMAIL_USER and app password in the EMAIL_PASS variable.

Also, you will need to **Allow less secure apps to be ON, and access captcha for using this in production environment, see this doc**

4. Use your local server URL in STORE_URL and ADMIN_URL variable, when you run on the local server your URL will http://localhost:3000, and http://localhost:4000 .

Finally, your .env file will look like this:

```
PORT=5055
MONGO_URI=your mongodb uri
JWT_SECRET=alamsfdfsdfsdfsdrafdar!@#$0dlfgjgsdfdsfdsfds
JWT_SECRET_FOR_VERIFY=lfjfjasjfr09ri09wrilfdjdj

SERVICE=gmail
EMAIL_USER=your email //change with your sender email
EMAIL_PASS=you email app password //change with your email app password
HOST=smtp.gmail.com
EMAIL_PORT=465
```

//use this when in dev/local server but when you will run on production/ live server then use that live URL in here and put that live URL on environment variable when hosting this backend

```
STORE_URL= http://localhost:3000
ADMIN_URL= http://localhost:4000
```

Once you successfully connect with MongoDB and configure .env then run "npm run data:import", it will run seed.js file and will import all demo data on the database. (You will find all demo data in the utils folder, change that data according to your need, also use staff email with real email for use of the forgetting password option) If everything is okay, then the backend configuration is done. Now you will find all demo data in your MongoDB database.

Now run *npm run dev*, it will run your backend on local server on PORT 5055 or your input PORT.

Note: Both Store and Admin backend is the same so use Only one for both

Store Configuration (Kachabazar) *[ignore this if you not purchase kachabazar]*

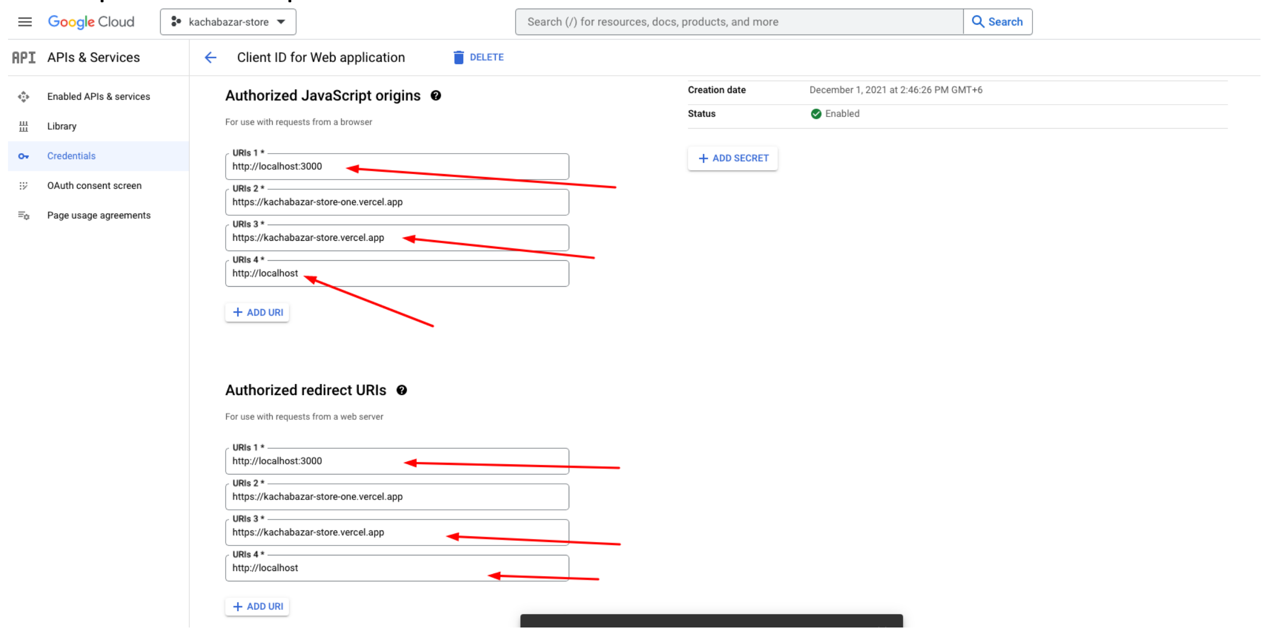
NPM Packages:

- After downloading the template, unzip it
- Open the Terminal /Vs code terminal
- Go to the folder (kachabazar): (cd path/to/template)
- Run: npm install, it will install all used packages for this kachabazar.
- Make sure that there is no error.

1. Rename .env.example to .env.local

2. Please watch this video for Cloudinary configuration Cloudinary configuration, doc (We use Cloudinary for profile image upload).

3. This video for google client id Google Client ID (For google sign in), your google developer console api & services Creadientals will look like this.



Google Cloud | kachabazar-store | Search (/) for resources, docs, products, and more | Search

APIs & Services — Client ID for Web application — DELETE

Enabled APIs & services · Library · Credentials · OAuth consent screen · Page usage agreements

Creation date: December 1, 2021 at 2:46:26 PM GMT+6
Status: Enabled
ADD SECRET

Authorized JavaScript origins
For use with requests from a browser
URIs 1 * http://localhost:3000
URIs 2 * https://kachabazar-store-one.vercel.app
URIs 3 * https://kachabazar-store.vercel.app
URIs 4 * http://localhost
ADD URI

Authorized redirect URIs
For use with requests from a web server
URIs 1 * http://localhost:3000
URIs 2 * https://kachabazar-store-one.vercel.app
URIs 3 * https://kachabazar-store.vercel.app
URIs 4 * http://localhost
ADD URI

4. Also need a stripe API key, for using the stripe payment option. If you want this, then go to react-stripe and create an account, get your test stripe API key, and put that on the .env.local variable. But if you do not add this value on the .env.local file then stripe will not work, others will work fine.

After Configure your .env.local file will look like this:

NEXT_PUBLIC_STRIPE_KEY="your stripe key" //for use stripe, change with your stripe API key

NEXT_PUBLIC_API_BASE_URL=http://localhost:5055/api //base API URL, when run on localhost/dev server.

NEXT_PUBLIC_CLOUDINARY_URL=https://api.cloudinary.com/v1_1/**your-cloudinary-user-name**/image/upload

NEXT_PUBLIC_CLOUDINARY_UPLOAD_PRESET=fg1vfge //your cloudinary upload preset

NEXT_PUBLIC_GOOGLE_CLIENT_ID=72898gfgdf0628gf79-jvugigp1d16rr0nf5hmvugfgtkiuogfgfh1ch.apps.goofgleusercontent.com //this one for google sign in, change with your google client API key

Now run *npm run dev*, it will run your kachabazar on local server on http://localhost:3000

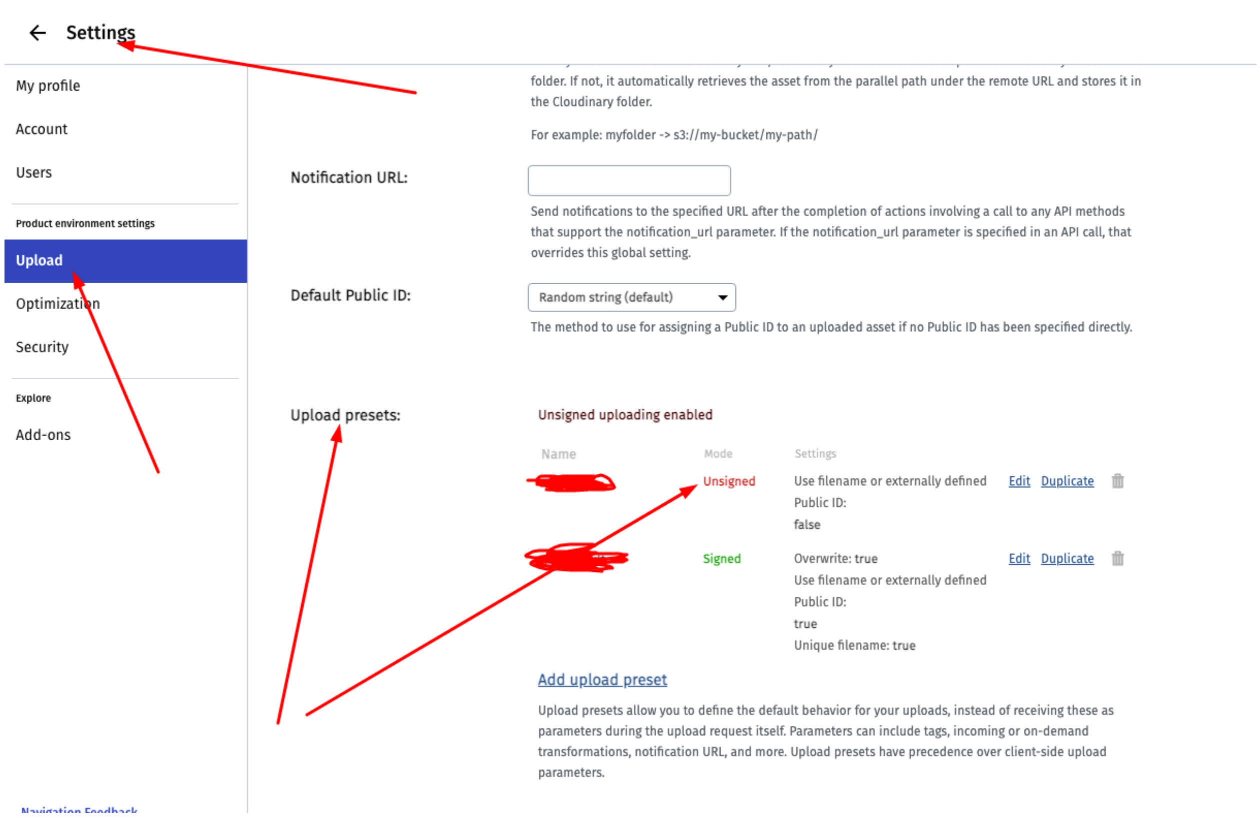
Admin Configuration (Dashtar)*[ignore this if you not purchase dashtar]*

NPM Packages:

- After downloading the template, unzip it
- Open the Terminal /Vs code terminal
- Go to the folder (dashtar): (cd path/to/template)
- Run: npm install, it will install all used packages for this dashtar.
- Make sure that there is no error.

1. Rename .env.example to .env.local

2. Please watch this video for Cloudinary configuration Cloudinary configuration, doc , (We use Cloudinary for image upload).Check bellow screenshot as well.



folder. If not, it automatically retrieves the asset from the parallel path under the remote URL and stores it in the Cloudinary folder.

For example: myfolder -> s3://my-bucket/my-path/

Notification URL:

Send notifications to the specified URL after the completion of actions involving a call to any API methods that support the notification_url parameter. If the notification_url parameter is specified in an API call, that overrides this global setting.

Default Public ID: Random string (default)

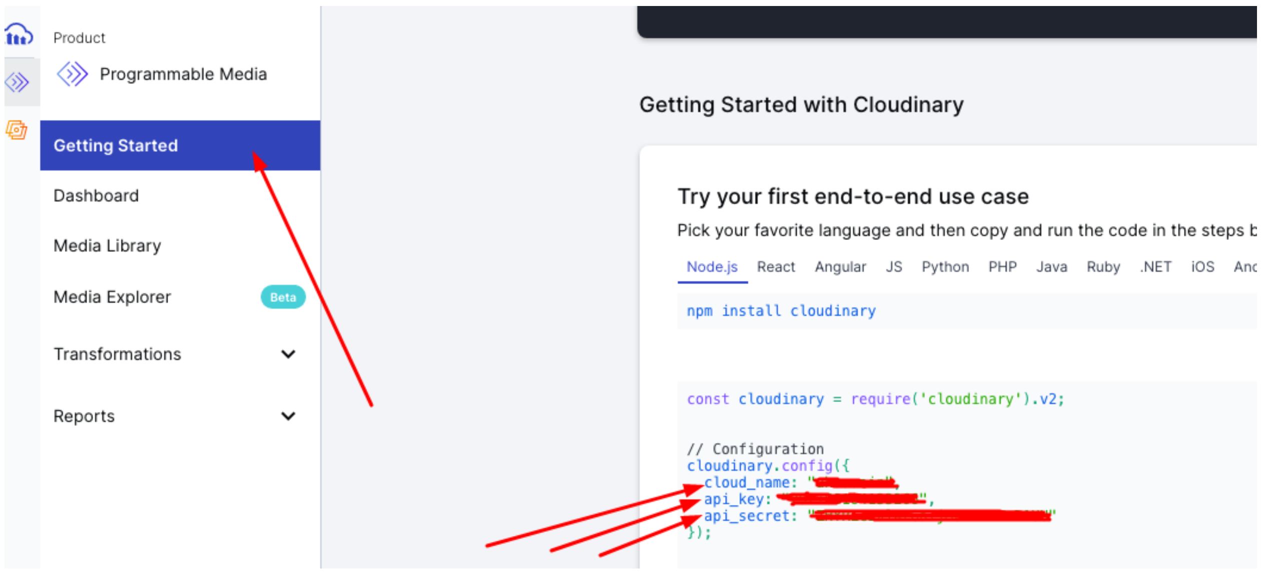
The method to use for assigning a Public ID to an uploaded asset if no Public ID has been specified directly.

Upload presets: Unsigned uploading enabled

Name	Mode	Settings		
[redacted]	Unsigned	Use filename or externally defined Public ID: false	Edit	Duplicate
[redacted]	Signed	Overwrite: true Use filename or externally defined Public ID: true Unique filename: true	Edit	Duplicate

Add upload preset

Upload presets allow you to define the default behavior for your uploads, instead of receiving these as parameters during the upload request itself. Parameters can include tags, incoming or on-demand transformations, notification URL, and more. Upload presets have precedence over client-side upload parameters.



Getting Started with Cloudinary

Try your first end-to-end use case

Pick your favorite language and then copy and run the code in the steps b

Node.js React Angular JS Python PHP Java Ruby .NET iOS Anc

```
npm install cloudinary
```

```
const cloudinary = require('cloudinary').v2;

// Configuration
cloudinary.config({
  cloud_name: "[redacted]",
  api_key: "[redacted]",
  api_secret: "[redacted]"
});
```

<u>After Configure your .env.local file will look like this:</u>

```
REACT_APP_API_BASE_URL=http://localhost:5055/api
REACT_APP_STORE_DOMAIN=http://localhost:3000

REACT_APP_CLOUD_NAME=you cloud name
REACT_APP_CLOUDINARY_API_KEY=your cloudinary api key
REACT_APP_CLOUDINARY_API_SECRET=api secret
REACT_APP_CLOUDINARY_UPLOAD_PRESET=upload preset
REACT_APP_CLOUDINARY_URL=https://api.cloudinary.com/v1_1/you_cloud_name/image/upload
```

Now run *npm start*, it will run your dashtar on local server on http://localhost:4000

Deploy On Vercel

As a next.js project vercel is recommended for deployment because If you deploy on vercel then it will automatically do everything for us and there will be no need for customization.

Here is your guide for deploying KachaBazar and Dashtar on vercel:

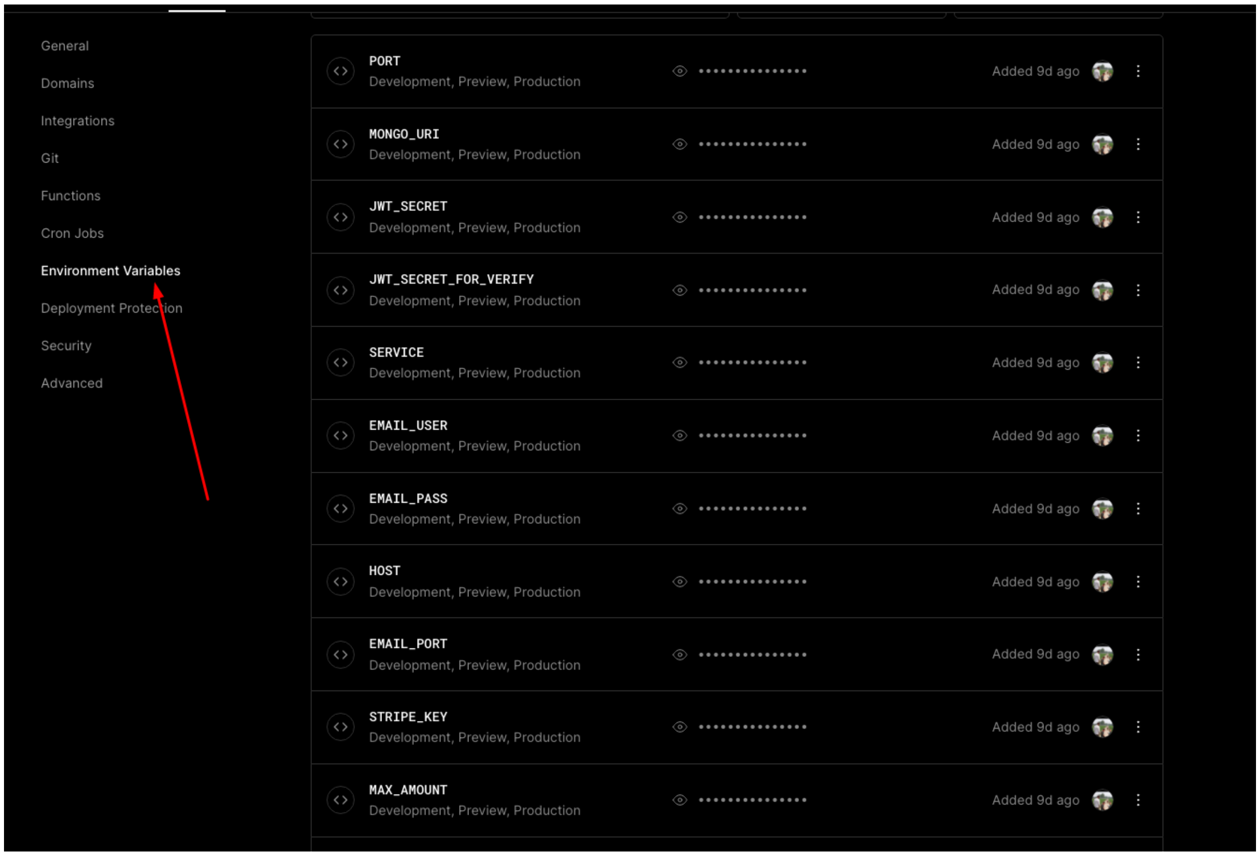
1. Create a GitHub account, go to vercel and sign up with that GitHub

account.

2. Create three private repositories on GitHub, then push your kachabazar code in one and backend code in one, and admin code in another repository.

3. Watch this video deploy on vercel, do according to.

4. When you import your GitHub repository on vercel by creating a project, you will see an option for Environment Variables, just click on that and give you a local .env all variable with the value. then click on deploy. Note first you will need to import and deploy backend, so that you can use that backend live url as kachabazar and dashtar base url.



PORT
Development, Preview, Production ••••••••••••••• Added 9d ago

MONGO_URI
Development, Preview, Production ••••••••••••••• Added 9d ago

JWT_SECRET
Development, Preview, Production ••••••••••••••• Added 9d ago

JWT_SECRET_FOR_VERIFY
Development, Preview, Production ••••••••••••••• Added 9d ago

SERVICE
Development, Preview, Production ••••••••••••••• Added 9d ago

EMAIL_USER
Development, Preview, Production ••••••••••••••• Added 9d ago

EMAIL_PASS
Development, Preview, Production ••••••••••••••• Added 9d ago

HOST
Development, Preview, Production ••••••••••••••• Added 9d ago

EMAIL_PORT
Development, Preview, Production ••••••••••••••• Added 9d ago

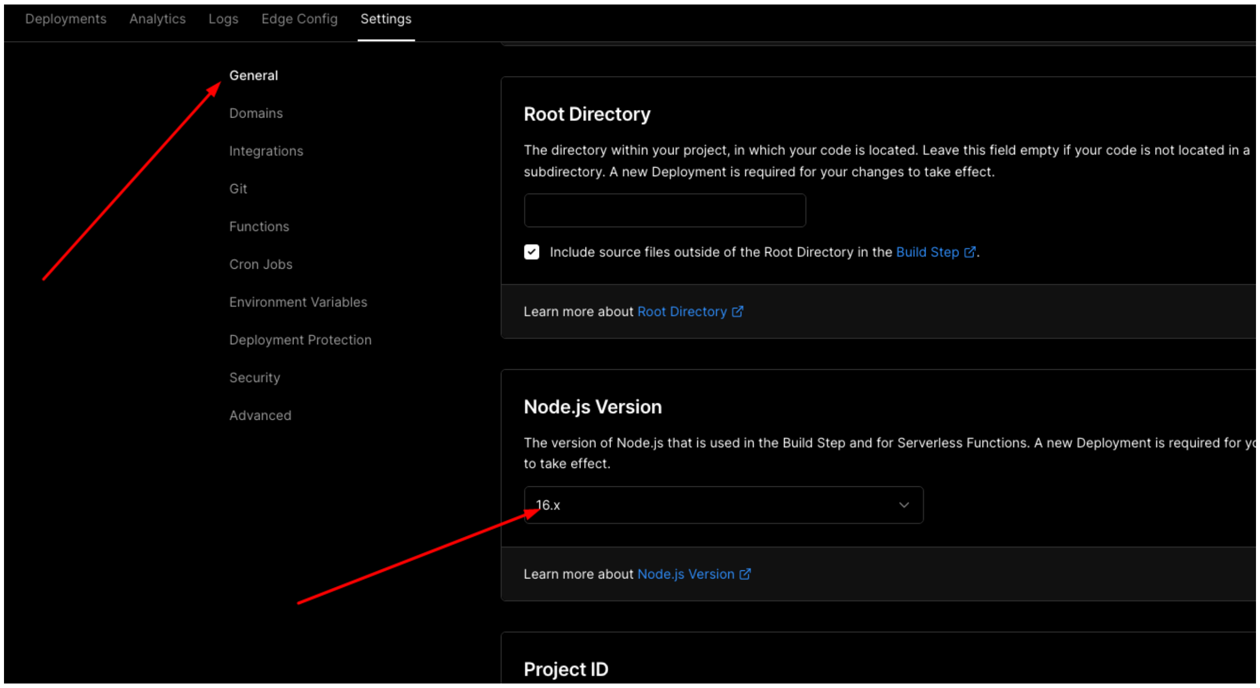
STRIPE_KEY
Development, Preview, Production ••••••••••••••• Added 9d ago

MAX_AMOUNT
Development, Preview, Production ••••••••••••••• Added 9d ago

First you have to deploy a backend project.

5. After the backend is deployed successfully, you will find a URL for your API route that will like this https://kachabazar.vercel.app/, and now change that like this https://kachabazar.vercel.app/api and use this as a NEXT_PUBLIC_API_BASE_URL and REACT_APP_API_BASE_URL when you deploy your store and admin project.

6. Now create another two projects for store and admin. Deploy one by one same as backend project and put all .env.local variable before clicking on deploy button, then click deploy, it will take some time for build and after that build, you will see you live version of Kacha Bazar store and Dashtar admin. For admin make sure you select node version 16x, only for admin.



Root Directory

The directory within your project, in which your code is located. Leave this field empty if your code is not located in a subdirectory. A new Deployment is required for your changes to take effect.

✓ Include source files outside of the Root Directory in the Build Step.

Learn more about Root Directory

Node.js Version

The version of Node.js that is used in the Build Step and for Serverless Functions. A new Deployment is required for y to take effect.

16.x

Learn more about Node.js Version

Project ID

7. If you do accordingly, then everything will be okay, for now when you make any changes on your local file, you just need to push your code on GitHub, vercel will automatically detect those changes and will redeploy your project with updated features.

You will find many videos on youtube and also articles on google about how to deploy next.js and express apps on vercel.