

Propsk / Coworking

<> Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Commit 9943176

InverskProperty committed 11 minutes ago · 0 / 1

removed category errors

main

1 parent cdaa7da commit 9943176

13 files changed +713 -575 lines changed

Filter files...

app

api/auth/[...nextauth]

route.js

debug-session

page.jsx

components

PropertyAddForm.jsx

PropertyCard.jsx

PropertyDetails.jsx

PropertyEditForm.jsx

models

Property.js

package-lock.json

package.json

scripts

investigate-api.js

investigate-auth.js

investigate.js

utils

authOptions.js

Customizable line height

The default line height has been increased for improved accessibility. You can choose to enable a more compact line height from the view settings menu.

Enable compact line height

Diamas

... @@ -1,6 +1,49 @@

1 - import { authOptions } from "@utils/authOptions";

2 - import NextAuth from "next-auth/react";

1 + // app/api/auth/[...nextauth]/route.js

2 + import NextAuth from "next-auth";

3 + import GoogleProvider from "next-auth/providers/google";

4 + import connectDB from "@config/database";

5 + import User from "@models/user";

3 6

7 + export const authOptions = {

8 + debug: process.env.MODE_ENV === "development",

9 + secret: process.env.NEXTAUTH_SECRET,

10 + providers: [

11 + GoogleProvider({

12 + clientId: process.env.GOOGLE_CLIENT_ID,

13 + clientSecret: process.env.GOOGLE_CLIENT_SECRET,

14 + authorization: {

15 + params: {

16 + prompt: "consent",

17 + access_type: "offline",

18 + response_type: "code",

19 + },

20 + },

21 +)

22 +],

23 + callbacks: {

24 + async signIn({ profile }) {

25 + await connectDB();

26 + const exists = await User.findOne({ email: profile.email });

27 + if (!exists) {

28 + await User.create({

29 + email: profile.email,

30 + username: profile.name.slice(0, 20),

31 + image: profile.picture,

32 + });

33 + }

34 + return true;

35 + },

36 + async session({ session }) {

37 + await connectDB();

38 + const user = await User.findOne({ email: session.user.email });

39 + session.user.id = user._id.toString();

40 + return session;

41 + },

42 + },

43 + };

44 +

45 + // Create a single NextAuth handler

46 + const handler = NextAuth(authOptions);

5 47

6 - export { handler as GET, handler as POST };

⊖

48 + // Export it for both GET and POST

49 + export { handler as GET, handler as POST };

... @@ -0,0 +1,32 @@

1 + // pages/debug-session.jsx (or app/debug-session/page.jsx if using the app folder)

2 + "use client"

3 + import { useSession, signIn, signOut } from "next-auth/react"

4 +

5 + export default function DebugSession() {

6 + const { data: session, status } = useSession()

7 +

8 + if (status === "loading") {

9 + return <loading></p>

10 + }

11 +

12 + if (!session) {

13 + return (

14 + <div>

15 + <p>Not signed in</p>

16 + <button onClick={() => signIn()}>Sign In</button>

17 + </div>

18 +)

19 + }

20 +

21 + return (

22 + <div>

23 + <p>

24 + Signed in as {session.user.email}

25 + </p>

26 + <pre style={{ background: "black", padding: "1em" }}>

27 + {JSON.stringify(session, null, 2)}

28 + </pre>

29 + <button onClick={() => signOut()}>Sign Out</button>

30 + </div>

31 +)

32 + }

https://github.com/Propsk/Coworking/commit/994317683557835004553dd8c49c7a016b479a1d

1/8

```

1  // @ -1,7 +1,12 @@
2  "use Client";
3  +
4  import { useEffect, useState } from 'react';
5  import { useReducer, useParams } from 'next/navigation';
6  import { fetchProperty } from '@utils/requests';
7  import { MESSAGES_TYPES } from '@models/Property';
8  +
9  const AMENITIES = ['Wi-Fi', 'Kitchen', 'Meeting Room', 'Parking', 'Air Conditioning', 'Printing'];
10 const num = (v) => (v === '' ? 0 : Number(v));
11
12 const PropertyEditForm = () => {
13   const router = useRouter();
14   @ -12,20 +17,14 @@ const PropertyEditForm = () => {
15     type: '',
16     name: '',
17     description: '',
18     location: '',
19     + location: { street: '', city: '', state: '', zipcode: '' },
20     square_feet: '',
21     - bath: '',
22     - bath: '',
23     + desk_capacity: '',
24     + rooms: '',
25     amenities: [],
26     images: [],
27     - rates: {
28       - daily: '',
29       - weekly: '',
30       - monthly: '',
31     },
32     - seller_name: '',
33     - seller_email: '',
34     - seller_phone: '',
35     + rates: { daily: '', weekly: '', monthly: '' },
36     + contact: { name: '', email: '', phone: '' },
37   });
38
39   const [newImages, setNewImages] = useState([]);
40   @ -41,41 +40,46 @@ const PropertyEditForm = () => {
41     weekly: property.rates.weekly || '',
42     monthly: property.rates.monthly || '',
43   },
44   + location: property.location || { street: '', city: '', state: '', zipcode: '' },
45   + contact: property.contact || { name: '', email: '', phone: '' },
46   + desk_capacity: property.desk_capacity || '',
47   + rooms: property.rooms || '',
48   images: property.images || [],
49   });
50   getProperties();
51   }, [id]);
52
53   const handleChange = (e) => {
54     - const { name, value, type, checked } = e.target;

```

3/8

<https://github.com/Propsk/Coworking/commit/994317683557835004553dd8c49c7a016b479a1d>

```
313 -       name="seller_phone"
314 -       value={formData.seller_phone}
315 -       onChange={handleChange}
316 -       className="w-full border px-3 py-2 rounded"
317 -     />
225 +     <input type="tel" name="contact_phone" value={formData.contact_phone} onChange={handleChange} className="w-full border px-3 py-2 rounded" />
318 226   </div>
319 227 </div>
320 228
321 229   <div className="text-center">
322 -     <button
323 -       type="submit"
324 -       className="bg-blue-600 text-white py-2 px-6 rounded hover:bg-blue-700"
325 -     />
230 +     <button type="submit" className="bg-blue-600 text-white py-2 px-6 rounded hover:bg-blue-700">
326 231       Update Workspace
327 232     </button>
328 233 </div>
T
```

```
*** @@ -1,88 +1,92 @@
1 1   import { Schema, model, models } from 'mongoose';
2 2
3 3   - const PropertySchema = new Schema({
4 4     + // Allowed workspace categories
5 5     + export const WORKSPACE_TYPES = [
6 6     +   'Hot Desk',
7 7     +   'Dedicated Desk',
8 8     +   'Private Office',
9 9     +   'Meeting Room',
10 10    +   'Whole Office',
11 11    + ];
12 12    + const PropertySchema = new Schema(
13 13    + {
14 14    +   // ----- Ownership -----
4 15    +   owner: {
5 16     -     type: Schema.Types.ObjectId,
6 17     -     ref: 'User',
7 18     -     required: true,
16 16     +     type: Schema.Types.ObjectId,
17 17     +     ref: 'User',
18 18     +     required: true,
8 19    },
20 20    +
21 21    + // ----- Core details -----
9 22    +   name: {
10 23     -     type: String,
11 24     -     required: true,
23 23     +     type: String,
24 24     +     required: true,
25 25     +     trim: true,
12 26    },
13 27    +   type: {
14 28     -     type: String,
15 29     -     required: true,
28 28     +     type: String,
29 29     +     enum: WORKSPACE_TYPES,
30 30     +     required: true,
16 31    },
17 32    +   description: {
18 33     -     type: String,
34 34     +     type: String,
35 35     +     trim: true,
36 36    },
37 37    +   // ----- Location -----
20 38    +   location: {
21 39     -     street: {
22 40     -       type: String
23 41     -     },
24 42     -     city: {
25 43     -       type: String
26 44     -     },
27 45     -     state: {
28 46     -       type: String
29 47     -     },
30 48     -     zipcode: {
31 49     -       type: String
32 50     -     }
39 39     +     street: String,
40 40     +     city: { type: String, required: true },
41 41     +     state: String,
42 42     +     zipcode: String,
33 43    },
34 44    +   beds: {
35 45     -     type: Number,
36 46     -     required: true,
44 44    +
45 45    +   // ----- Capacity & size -----
46 46    +   desk_capacity: {
47 47     -     type: Number,
48 48     -     required: true,
49 49     -     min: 0,
37 50    },
38 51    +   baths: {
39 52     -     type: Number,
40 53     -     required: true,
51 51    +   rooms: {
52 52     -     type: Number,
53 53     -     required: true,
54 54     -     min: 0,
41 55    },
42 56    +   square_feet: {
43 57     -     type: Number,
44 58     -     required: true,
57 57     +     type: Number,
58 58     +     required: true,
59 59     +     min: 0,
45 60    },
46 61    +   amenities: [
47 62     -     {
48 63     -       type: String
49 64     -     }
50 65    ],
61 61    +   // ----- Amenities -----
62 62    +   amenities: [String],
64 64    +   // ----- Rates -----
51 66    +   rates: {
52 67     -     nightly: {
53 68     -       type: Number
54 69     -     },
55 69     -     weekly: {
56 70     -       type: Number
57 71     -     },
58 72     -     monthly: {
59 73     -       type: Number
60 74     -     }
67 67     +     daily: { type: Number, min: 0 },
68 68     +     weekly: { type: Number, min: 0 },
69 69     +     monthly: { type: Number, min: 0 },
61 70    },
62 71    +   seller_info: {
63 72     -     name: {
64 73     -       type: String
65 74     -     },
66 75     -     email: {
67 76     -       type: String
68 77     -     },
69 78     -     phone: {
70 79     -       type: String
71 80     -     }
72 81    }
```

```
71 + // ----- Contact -----
72 + // ----- Contact -----
73 + contact: {
74 +   name: String,
75 +   email: String,
76 +   phone: String,
77 + },
78 - images: [
79 -   {
80 -     type: String
81 -   }
82 - ],
83 - is_featured: {
84 -   type: Boolean,
85 -   default: false
86 - }
87 - }, {
88 -   timestamps: true
89 - });
90 + // ----- Media -----
91 + images: [String],
92 + // ----- Flags -----
93 + is_featured: {
94 +   type: Boolean,
95 +   default: false,
96 + },
97 + },
98 + { timestamps: true },
99 + );
100
101 const Property = models.Property || model('Property', PropertySchema);
102
103 ***
```

Load Diff

Some generated files are not rendered by default. Learn more about [customizing how changed files appear on GitHub](#).

package-lock.json

+112 -54

```
@@ -11,10 +11,12 @@
11 11   "dependencies": {
12 12     "bcryptjs": "2.4.3",
13 13     "cloudinary": "2.13.0",
14 14     "dotenv": "16.4.5",
15 15     "mongoose": "8.8.1",
16 16     "mongoose": "8.13.2",
17 17     "next": "15.3.4",
18 18     "next-auth": "4.24.1",
19 19     "node-fetch": "3.3.2",
20 20     "photoswipe": "5.4.4",
21 21     "react": "18.3.1",
22 22     "react-dom": "18.3.1",
```

```
*** @ -0.0 +1,72 @
1 + // scripts/investigate-api.js
2 + require('dotenv').config({ path: '.env.local' });
3 + const mongoose = require('mongoose');
4 +
5 + // Single probe function
6 + async function probe(method, path, body = null) {
7 +   const base = process.env.NEXT_PUBLIC_BASE_URL || process.env.NEXTAUTH_URL;
8 +   if (!base) {
9 +     console.error('▲ NEXT_PUBLIC_BASE_URL / NEXTAUTH_URL not set');
10 +    return;
11 +  }
12 +   const url = base.replace(/\/$/, '') + path;
13 +   process.stdout.write(`\n ⚡ [method] ${path} - `);
14 +   const start = Date.now();
15 +   try {
16 +     const res = await fetch(url, {
17 +       method,
18 +       headers: body
19 +       ? { 'Content-Type': 'application/json' }
20 +       : undefined,
21 +       body: body ? JSON.stringify(body) : undefined,
22 +     });
23 +     const ms = Date.now() - start;
24 +     const ct = res.headers.get('content-type') || '';
25 +     const text = await res.text();
26 +     console.log(`status: ${res.status} (${ms}ms), content-type: ${ct}`);
27 +     console.log(`%s snippet`, text.slice(0, 200).replace(/\n/g, ' '));
28 +   } catch (err) {
29 +     console.error('X Fetch error:', err);
30 +   }
31 + }
32 +
33 + async function testMongo() {
34 +   console.log('\n 🧪 Testing MongoDB connection...');
35 +   const url = process.env.MONGODB_URI;
36 +   if (!url) {
37 +     console.warn('▲ MONGODB_URI not set');
38 +     return;
39 +   }
40 +   try {
41 +     const ts = Date.now();
42 +     await mongoose.connect(url);
43 +     console.log(`✓ MongoDB connected (${Date.now() - ts}ms)`);
44 +   } catch (err) {
45 +     console.error('X mongoose.connect error:', err);
46 +   } finally {
47 +     await mongoose.disconnect();
48 +   }
49 + }
50 +
51 + ;(async () => {
52 +   await testMongo();
53 +
54 +   // Health endpoints (we know these are OK, but for completeness)
55 +   await probe('GET', '/api/auth/session');
56 +   await probe('GET', '/api/auth/providers');
57 +   await probe('GET', '/api/auth/callback');
58 +
59 +   // Your property routes
60 +   await probe('GET', '/api/properties');
61 +   await probe('GET', '/api/properties?page=1&pageSize=1');
62 +   await probe('GET', '/api/properties/123INVALID'); // invalid ID
63 +   await probe('POST', '/api/properties', { foo: 'bar' }); // should 401 or 500
64 +   await probe('PUT', '/api/properties/123INVALID', { foo: 'bar' });
65 +   await probe('DELETE', '/api/properties/123INVALID');
66 +
67 +   // Upload route
68 +   await probe('POST', '/api/upload');
69 +
70 +   console.log('\n 🏁 All done.');
```

```
*** @ -0.0 +1,52 @
1 + // scripts/investigate-auth.js
2 + require('dotenv').config({ path: '.env.local' });
3 + const mongoose = require('mongoose');
4 +
5 + async function probe(path) {
6 +   const base = process.env.NEXTAUTH_URL?.replace(/\/$/, '');
7 +   if (!base) {
8 +     console.error('▲ NEXTAUTH_URL not set');
```


```
7 + if (!base) {
8 +   console.error("▲ NEXTAUTH_URL not set in .env.local");
9 +   return;
10 + }
11 + const url = base + path;
12 + process.stdout.write(`\n ⚡ GET ${path} - `);
13 + const start = Date.now();
14 + try {
15 +   const res = await fetch(url);
16 +   const ms = Date.now() - start;
17 +   const ct = res.headers.get('content-type') || '';
18 +   const text = await res.text();
19 +   console.log(`status: ${res.status} (${ms}ms), content-type: ${ct}`);
20 +   console.log(`~ body snippet:`, text.slice(0, 200).replace(/\n/g, ' '));
21 + } catch (err) {
22 +   console.error("X Fetch error:", err);
23 + }
24 + }
25 +
26 + async function testMongo() {
27 +   console.log(`\n ⚡ Testing MongoDB connection.`);
28 +   const url = process.env.MONGODB_URI;
29 +   if (!url) {
30 +     console.warn("▲ MONGODB_URI not set");
31 +     return;
32 +   }
33 +   try {
34 +     const start = Date.now();
35 +     await mongoose.connect(url, { dbName: 'coworking' });
36 +     console.log(`✓ MongoDB connected (${Date.now() - start}ms)`);
37 +   } catch (err) {
38 +     console.error("X mongoose.connect error:", err);
39 +   } finally {
40 +     await mongoose.disconnect();
41 +   }
42 + }
43 +
44 + ;(async () => {
45 +   // NextAuth public endpoints
46 +   await probe('/api/auth/session');
47 +   await probe('/api/auth/providers');
48 +   await probe('/api/auth/callback');
49 +   await testMongo();
50 +   console.log(`\n 🏁 Investigation complete.`);
51 +   process.exit(0);
52 + })();
```


```
*** @ -8.0 v1.56 @
1 + // scripts/investigate.js
2 + require('dotenv').config({ path: '.env.local' });
3 + const mongoose = require('mongoose');
4 +
5 + async function testSession() {
6 +   console.log(`\n ⚡ Testing /api/auth/session.`);
7 +
8 +   const base = process.env.NEXTAUTH_URL?.replace(/\/$/, '');
9 +   if (!base) {
10 +     console.error("▲ NEXTAUTH_URL not set in .env.local");
11 +     return;
12 +   }
13 +
14 +   const url = base + '/api/auth/session';
15 +   const start = Date.now();
16 +   try {
17 +     const res = await fetch(url); // use global fetch
18 +     const ms = Date.now() - start;
19 +     const text = await res.text();
20 +     console.log(` ~ GET ${url}`);
21 +     console.log(` ~ status: ${res.status} (${ms}ms)`);
22 +     console.log(` ~ content-type: ${res.headers.get('content-type')}`);
23 +     console.log(` ~ body (first 200 chars):\n ~ ` +
24 +       text.slice(0, 200) +
25 +       `(text.length > 200 ? '..' : '')`);
26 +   } catch (err) {
27 +     console.error(" X Fetch error:", err);
28 +   }
29 + }
30 +
31 + async function testDB() {
32 +   console.log(`\n ⚡ Testing MongoDB connection.`);
33 +
34 +   const url = process.env.MONGODB_URI;
35 +   if (!url) {
36 +     console.warn("▲ MONGODB_URI not set in .env.local");
37 +     return;
38 +   }
39 +
40 +   const start = Date.now();
41 +   try {
42 +     await mongoose.connect(url, { dbName: 'coworking' });
43 +     console.log(` ~ connected to MongoDB in ${Date.now() - start}ms`);
44 +   } catch (err) {
45 +     console.error(" X mongoose.connect error:", err);
46 +   } finally {
47 +     await mongoose.disconnect();
48 +   }
49 + }
50 +
51 + (async () => {
52 +   await testSession();
53 +   await testDB();
54 +   console.log(`\n 🏁 Done.`);
55 +   process.exit(0);
56 + })();
```

```
*** @ -1.18 v1.55 @
1 - import GoogleProvider from 'next-auth/providers/google';
2 - import connectDB from '@config/database';
3 - import User from '@models/User';
4
5 + // utils/authOptions.js
6
7 - export const authOptions = {
8 -   providers: [
9 -     GoogleProvider({
10 -       clientId: process.env.GOOGLE_CLIENT_ID,
11 -       clientSecret: process.env.GOOGLE_CLIENT_SECRET,
12 -       authorization: {
13 -         params: {
14 -           prompt: "consent",
15 -           access_type: "offline",
16 -           response_type: "code"
17 -         },
18 -       },
19 -     }),
20 -   ],
21 -   secret: process.env.NEXTAUTH_SECRET,
22 -   callbacks: {
23 -     // Invoked on successful login
24 -     async signIn({ profile }) {
25 -
26 -       // 1. connect to database
27 -       await connectDB();
28 -
29 -       // 2. Check if user exists
30 -       const userExists = await User.findOne({ email: profile.email });
31 -
32 -       // 3. If not, then add user to database
33 -       if (!userExists) {
34 -
35 -         // truncate user name if too long
36 -         const username = profile.name.slice(0, 20);
37 -         await User.create({
```

```
36 -         email: profile.email,
37 -         username,
38 -         image: profile.picture
39 -       });
40 -     }
41 + import GoogleProvider from "next-auth/providers/google"
42 + import connectDB from "@/config/database"
43 + import User from "@/models/user"
44
45 - // 4. Return true to allow sign in
46 - return true;
47 + export const authOptions = {
48 +   providers: [
49 +     GoogleProvider({
50 +       clientId: process.env.GOOGLE_CLIENT_ID,
51 +       clientSecret: process.env.GOOGLE_CLIENT_SECRET,
52 +       authorization: {
53 +         params: {
54 +           prompt: "consent",
55 +           access_type: "offline",
56 +           response_type: "code",
57 +         },
58 +       },
59 +     }),
60 +   ],
61 +   // Modify the session object
62 +   async session({session}) {
63 +     // 1. Get user from database
64 +     const user = await User.findOne({ email: session.user.email });
65 +
66 +     // 2. Assign the user id to the session
67 +     session.user.id = user._id.toString();
68 +
69 +     // 3. return session
70 +     return session;
71 +   },
72 + };
73
74 + // 1) On initial sign-in, create the user in your DB if needed:
75 + async signIn({ profile }) {
76 +   await connectDB();
77 +   const exists = await User.findOne({ email: profile.email });
78 +   if (!exists) {
79 +     const username = profile.name.slice(0, 20);
80 +     await User.create({
81 +       email: profile.email,
82 +       username,
83 +       image: profile.picture,
84 +     });
85 +   }
86 +   return true;
87 + },
88 + // 2) After sign-in, persist the DB user_id onto the JWT
89 + async jwt({ token, user }) {
90 +   if (user) {
91 +     // user.id or user._id depending on what NextAuth hands you
92 +     token.id = user.id ?? user._id.toString();
93 +   }
94 +   return token;
95 + },
96 + // 3) Whenever a session is checked, copy token.id = session.user.id
97 + async session({ session, token }) {
98 +   session.user.id = token.id;
99 +   return session;
100 + },
101 + };
102 + }
```

Comments 0 Lock conversation

 Comment

 Subscribe You're not receiving notifications from this thread