Render Database Seeding Instructions

Overview

This guide explains how to seed the PostgreSQL database on Render with test users and sample data.

Test User Credentials

After seeding, you can log in with any of these accounts:

Standard Test Users

1. **Test User (Individual)**
   - Email: `test@example.com`
   - Password: `Test123!`
   - Role: INDIVIDUAL (Job Seeker / Donation Receiver)

2. **Donor User (Corporate)**
   - Email: `donor@example.com`
   - Password: `Donor123!`
   - Role: CORPORATE (Employer / Food Donor)

3. **Job Seeker**
   - Email: `seeker@example.com`
   - Password: `Seeker123!`
   - Role: INDIVIDUAL (Job Seeker)

Turkish Named Users

1. **Mehmet Yılmaz (Individual)**
   - Email: `mehmet@example.com`
   - Password: `user123`
   - Role: INDIVIDUAL

2. **Ayşe Demir (Individual)**
   - Email: `ayse@example.com`
   - Password: `user123`
   - Role: INDIVIDUAL

Corporate/Admin Users

1. **Tech Solutions Ltd.**
   - Email: `company1@workigom.com`
   - Password: `company123`
   - Role: CORPORATE

2. **Restoran Lezzet**
   - Email: `company2@workigom.com`
   - Password: `company123`
   - Role: CORPORATE

3. **Admin User**
   - Email: `admin@workigom.com`
   - Password: `admin123`
   - Role: ADMIN

Method 1: Using Render Shell (Recommended)

Step 1: Access Render Shell

1. Go to your Render dashboard: https://dashboard.render.com/
2. Navigate to your **backend service** (workigom-backend)
3. Click on the **Shell** tab in the left sidebar
4. Wait for the shell to connect

Step 2: Run Seed Command

In the Render shell, run the following commands:

```
# Navigate to the backend directory (if needed)
cd /opt/render/project/src/backend

# Run Prisma migrations (if not already applied)
npx prisma migrate deploy

# Run the seed script
npx prisma db seed
```

Step 3: Verify Success

You should see output like:

```
🌱 Seeding database...
✅ Admin user created: admin@workigom.com
✅ Corporate user 1 created: company1@workigom.com
✅ Corporate user 2 created: company2@workigom.com
✅ Individual user 1 created: mehmet@example.com
✅ Individual user 2 created: ayse@example.com
✅ Test user created: test@example.com
✅ Donor user created: donor@example.com
✅ Seeker user created: seeker@example.com
✅ Job 1 created: Yazılım Geliştirici
✅ Job 2 created: Garson
...
🎉 Database seeding completed successfully!
```

Method 2: Using Render Environment Variables (Alternative)

If the shell method doesn’t work, you can create a one-time job:

Step 1: Add Seed Script to Start Command (Temporary)

1. Go to your backend service settings
2. Find the **Start Command** field
3. Temporarily change it to:
   `bash`
   `npm run prisma:migrate:deploy && npm run prisma:seed && npm start`
4. Click **Save Changes**
5. Wait for the service to redeploy

Step 2: Revert Start Command

After the seed runs successfully:
1. Change the **Start Command** back to:
`bash`
`npm run prisma:migrate:deploy && npm start`
2. Click **Save Changes**

Method 3: Run Seed Script via Deploy Hook (For CI/CD)

Step 1: Create a Seed Endpoint (Optional)

If you want to seed on-demand, you can create an admin endpoint in your backend:

```
// backend/src/routes/admin.routes.ts
import { Router } from 'express';
import { exec } from 'child_process';
import { promisify } from 'util';

const router = Router();
const execAsync = promisify(exec);

router.post('/seed', async (req, res) => {
  // Add authentication check here
  const adminSecret = req.headers['x-admin-secret'];

  if (adminSecret !== process.env.ADMIN_SECRET) {
    return res.status(401).json({ error: 'Unauthorized' });
  }

  try {
    const { stdout, stderr } = await execAsync('npm run prisma:seed');
    res.json({ success: true, output: stdout, errors: stderr });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});

export default router;
```

Then call it via:

```
curl -X POST https://workigom-backend.onrender.com/api/admin/seed \
  -H "x-admin-secret: YOUR_SECRET"
```

Troubleshooting

Error: “Cannot connect to database”

- Check that your `DATABASE_URL` environment variable is correctly set
- Verify the PostgreSQL database is running

Error: “prisma command not found”

- Make sure you’re in the correct directory
- Try using `npx prisma` instead of just `prisma`

Error: “Unique constraint failed”

- The seed script uses `upsert`, so it won’t create duplicate users
- If you still get this error, users might already exist
- You can reset the database and reseed:

  ```
  bash
    npx prisma migrate reset --force
    npx prisma db seed
  ```

  ⚠️ **WARNING**: This will DELETE ALL DATA!

Error: “Module not found: ts-node”

- Install ts-node in the backend directory:

  ```
  bash
    npm install --save-dev ts-node
  ```

Verifying the Seed

Option 1: Using Prisma Studio (Local Only)

```
cd backend
npx prisma studio
```

Option 2: Test Login on Frontend

1. Go to: https://workigom-frontend1.onrender.com
2. Try logging in with: `test@example.com` / `Test123!`
3. You should be successfully authenticated

Option 3: Using Database Client

1. Get your `DATABASE_URL` from Render environment variables
2. Connect using a PostgreSQL client (e.g., pgAdmin, DBeaver, psql)
3. Run query:

   ```
   sql
      SELECT email, name, role FROM users;
   ```

Important Notes

1. **Seed is Idempotent**: Running the seed multiple times won't create duplicates (uses `upsert`)
2. **Production Warning**: Only seed databases in development/staging. For production, create users through the app or admin panel.
3. **Environment Variables**: Make sure all required environment variables are set in Render:
   - `DATABASE_URL` (set automatically by Render PostgreSQL)
   - `JWT_SECRET`
   - `JWT_EXPIRES_IN`
   - `REFRESH_TOKEN_SECRET`
   - `REFRESH_TOKEN_EXPIRES_IN`

After Seeding

Once seeding is complete, users should be able to log in at:
- Frontend: https://workigom-frontend1.onrender.com
- Backend API: https://workigom-backend.onrender.com

Test the login endpoint directly:

```
curl -X POST https://workigom-backend.onrender.com/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"email":"test@example.com","password":"Test123!"}'
```

You should get a response with a JWT token and user information.

Contact & Support

If you encounter any issues:
1. Check the Render logs for error messages
2. Verify all environment variables are set
3. Ensure the PostgreSQL database is properly connected
4. Review the seed script output for any errors