To create a MongoDB database with Prisma for a projects collection that supports Arabic (ar) and English (en)

To create a MongoDB database with Prisma for a projects collection that supports multilingual fields (e.g., title, description) in both Arabic (ar) and English (en), follow these steps:

Define the Prisma Schema

```
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "mongodb"
  url      = env("DATABASE_URL") // Add your MongoDB connection string in .env
}

model Project {
  id          String   @id @default(auto()) @map("_id") @db.ObjectId
  title       Json     // Multilingual field for title (e.g., { en: "Title", ar: "العنوان" })
  description Json     // Multilingual field for description (e.g., { en: "Desc", ar: "الوصف" }
  images      String[] // Array of image URLs
  category    String
  country     String
  city        String
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
}
```

Add a Project Using Prisma

```ts
import { PrismaClient } from '@prisma/client'

const prisma = new PrismaClient()

export async function addProject(data: {
  title: { en: string; ar: string }
  description: { en: string; ar: string }
  images: string[]
  category: string
  country: string
  city: string
}) {
  return await prisma.project.create({
    data: {
      title: data.title,
      description: data.description,
      images: data.images,
      category: data.category,
      country: data.country,
      city: data.city,
    },
  })
}
```

Fetch Projects with Multilingual Support

```
export async function getProjects(locale: 'en' | 'ar') {
        const projects = await prisma.project.findMany()
        return projects.map((project) => ({
                id: project.id,
                title: project.title[locale],
                description: project.description[locale],
                images: project.images,
                category: project.category,
                country: project.country,
                city: project.city,
                createdAt: project.createdAt,
                updatedAt: project.updatedAt,
        }))
}
```

Use the Functions in Your Next.js Pages

You can now use the addProject and getProjects functions in your Next.js API routes or pages.

Example: Add a Project

```
import { NextApiRequest, NextApiResponse } from 'next'
import { addProject } from '../../lib/projectService'

export default async function handler(
        req: NextApiRequest,
        res: NextApiResponse
) {
        if (req.method === 'POST') {
                const data = req.body
                const project = await addProject(data)
                res.status(201).json(project)
        } else {
                res.status(405).json({ message: 'Method not allowed' })
        }
}
```

Example: Fetch Projects

```
import { NextApiRequest, NextApiResponse } from 'next'
import { getProjects } from '../../lib/projectService'

export default async function handler(req: NextApiRequest, res: NextApiResponse) {
  const locale = req.query.locale || 'en'
  const projects = await getProjects(locale as 'en' | 'ar')
  res.status(200).json(projects)
}
```

Example Data

```
{
  "title": { "en": "New Project", "ar": "مشروع جديد" },
  "description": { "en": "This is a new project.", "ar": "هذا مشروع جديد." },
  "images": ["https://example.com/image1.jpg", "https://example.com/image2.jpg"],
  "category": "Technology",
  "country": "USA",
  "city": "New York"
}
```

Query Example

To fetch projects in Arabic (ar):

```
GET /api/getProjects?locale=ar
```

Response:

```
;[
        {
                id: '1',
                title: 'مشروع جديد',
                description: 'هذا مشروع جديد.',
                images: [
                        'https://example.com/image1.jpg',
                        'https://example.com/image2.jpg',
                ],
                category: 'Technology',
                country: 'USA',
                city: 'New York',
                createdAt: '2025-05-01T12:00:00.000Z',
                updatedAt: '2025-05-01T12:00:00.000Z',
        },
]
```

Create a Form Page to Add Data

```
import { useState } from 'react'

export default function AddProject() {
        const [formData, setFormData] = useState({
                titleEn: '',
                titleAr: '',
                descriptionEn: '',
                descriptionAr: '',
                images: '',
                category: '',
                country: '',
                city: '',
        })

        const handleChange = (
                e: React.ChangeEvent<HTMLInputElement | HTMLTextAreaElement>
        ) => {
                setFormData({ ...formData, [e.target.name]: e.target.value })
        }

        const handleSubmit = async (e: React.FormEvent) => {
                e.preventDefault()

                const data = {
                        title: { en: formData.titleEn, ar: formData.titleAr },
                        description: { en: formData.descriptionEn, ar: formData.descriptionAr },
                        images: formData.images.split(',').map((img) => img.trim()), // Split ir
                        category: formData.category,
                        country: formData.country,
                        city: formData.city,
                }

                const response = await fetch('/api/addProject', {
                        method: 'POST',
                        headers: { 'Content-Type': 'application/json' },
                        body: JSON.stringify(data),
                })

                if (response.ok) {
                        alert('Project added successfully!')
                        setFormData({
```

```
                        titleEn: '',
                        titleAr: '',
                        descriptionEn: '',
                        descriptionAr: '',
                        images: '',
                        category: '',
                        country: '',
                        city: '',
                })
        } else {
                alert('Failed to add project.')
        }
}

return (
        <div className='container'>
                <h1>Add Project</h1>
                <form onSubmit={handleSubmit}>
                        <div>
                                <label>Title (English):</label>
                                <input
                                        type='text'
                                        name='titleEn'
                                        value={formData.titleEn}
                                        onChange={handleChange}
                                        required
                                />
                        </div>
                        <div>
                                <label>Title (Arabic):</label>
                                <input
                                        type='text'
                                        name='titleAr'
                                        value={formData.titleAr}
                                        onChange={handleChange}
                                        required
                                />
                        </div>
                        <div>
                                <label>Description (English):</label>
                                <textarea
                                        name='descriptionEn'
                                        value={formData.descriptionEn}
```

```
                onChange={handleChange}
                required
            />
        </div>
        <div>
            <label>Description (Arabic):</label>
            <textarea
                name='descriptionAr'
                value={formData.descriptionAr}
                onChange={handleChange}
                required
            />
        </div>
        <div>
            <label>Images (comma-separated URLs):</label>
            <input
                type='text'
                name='images'
                value={formData.images}
                onChange={handleChange}
                required
            />
        </div>
        <div>
            <label>Category:</label>
            <input
                type='text'
                name='category'
                value={formData.category}
                onChange={handleChange}
                required
            />
        </div>
        <div>
            <label>Country:</label>
            <input
                type='text'
                name='country'
                value={formData.country}
                onChange={handleChange}
                required
            />
        </div>
```

```
                        <div>
                                <label>City:</label>
                                <input
                                        type='text'
                                        name='city'
                                        value={formData.city}
                                        onChange={handleChange}
                                        required
                                />
                        </div>
                        <button type='submit'>Add Project</button>
                </form>
        </div>
    )
}
```