Optimized Recruitment Website Code

```
// Frontend: Next.js (pages/index.js)
import { useState } from 'react';
import Header from '../components/Header';
import CandidateSearch from '../components/CandidateSearch';
import Testimonials from '../components/Testimonials';

export default function Home() {
  return (
    <div className="min-h-screen bg-gray-50">
      <Header />
      <main className="container mx-auto px-4 py-8">
        <CandidateSearch />
        <Testimonials />
      </main>
    </div>
  );
}

// components/CandidateSearch.js
import { useState, useEffect } from 'react';

export default function CandidateSearch() {
  const [candidates, setCandidates] = useState([]);
  const [filters, setFilters] = useState({ skills: [], experience: '', location: '' });

  useEffect(() => {
    const fetchCandidates = async () => {
                                                                        const            query            =
`?skills=${filters.skills.join(',')}&experience=${filters.experience}&location=${filters.location}`;
      const res = await fetch(`/api/candidates${query}`);
      const data = await res.json();
      setCandidates(data);
    };

    fetchCandidates();
  }, [filters]);

  return (
    <div className="bg-white p-6 rounded-lg shadow-md">
              <input  type="text"  placeholder="Skills"  onChange={(e)  =>  setFilters({...filters,  skills:
e.target.value.split(',')})} />
      {candidates.map(candidate => (
        <div key={candidate._id}>
          <h3>{candidate.name}</h3>
          <p>{candidate.skills.join(', ')}</p>
                <a  href={`mailto:a.recruiter.team@gmail.com?subject=Inquiry  about  ${candidate.name}`}>Contact
Recruiter</a>
        </div>
      ))}
```

```
    </div>
  );
}

// Backend: Node.js/Express (server.js)
const express = require('express');
const mongoose = require('mongoose');
const jwt = require('jsonwebtoken');
const userRoutes = require('./routes/users');
const candidateRoutes = require('./routes/candidates');

const app = express();
app.use(express.json());

mongoose.connect(process.env.MONGODB_URI, { useNewUrlParser: true, useUnifiedTopology: true });

const authenticate = (req, res, next) => {
  const token = req.header('Authorization')?.replace('Bearer ', '');
  if (!token) return res.status(401).send('Access denied');
  try {
    req.user = jwt.verify(token, process.env.JWT_SECRET);
    next();
  } catch (err) {
    res.status(400).send('Invalid token');
  }
};

app.use('/api/users', userRoutes);
app.use('/api/candidates', authenticate, candidateRoutes);

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

// routes/candidates.js
const express = require('express');
const router = express.Router();
const Candidate = require('../models/Candidate');

router.get('/', authenticate, async (req, res) => {
  try {
    const filters = {};
    if (req.query.skills) {
      filters.skills = { $in: req.query.skills.split(',') };
    }
    const candidates = await Candidate.find(filters);
    res.json(candidates);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

module.exports = router;

// models/Candidate.js
```

```
const mongoose = require('mongoose');

const candidateSchema = new mongoose.Schema({
  name: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  skills: [{ type: String }],
  experience: { type: Number, required: true },
  location: String,
  resumeUrl: String
});

module.exports = mongoose.model('Candidate', candidateSchema);
```