server.js

```
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const User = require('./models/User');
// const uri =
"mongodb+srv://tanishkhot29:XvyCNzRNtwkKmOQ1@wadl.cxhbz.mongodb.net/userdb?
retryWrites=true&w=majority";

const uri = "mongodb://localhost:27017/userdb";

const app = express();
app.use(express.json());
app.use(cors({
    origin: 'http://localhost:3000',
    methods: ['GET', 'POST', 'PUT', 'DELETE'],
    credentials: true,
    allowedHeaders: ['Content-Type', 'Authorization']
  }));


// Update the mongoose connection
// mongoose.connect(uri, {
//   serverSelectionTimeoutMS: 5000,
//   socketTimeoutMS: 45000,
//   family: 4
// })
// .then(() => console.log('Connected to MongoDB'))
// .catch(err => console.error('MongoDB connection error:', err));

mongoose.connect(uri)
  .then(() => console.log('Connected to MongoDB'))
  .catch(err => console.error('MongoDB connection error:', err));

app.post('/api/register', async (req, res) => {
  try {
    const { name, email, password } = req.body;

    const existingUser = await User.findOne({ email });
    if (existingUser) {
      return res.status(400).json({ message: 'User already exists' });
    }

    const user = new User({
      name,
      email,
      password
    });

    await user.save();
    res.status(201).json({ message: 'User registered successfully' });
  } catch (error) {
    res.status(500).json({ message: 'Server error' });
```

```
  }
});

app.post('/api/login', async (req, res) => {
  try {
    const { email, password } = req.body;

    const user = await User.findOne({ email, password });
    if (!user) {
      return res.status(400).json({ message: 'Invalid credentials' });
    }

    res.json({ message: 'Login successful', userId: user._id });
  } catch (error) {
    res.status(500).json({ message: 'Server error' });
  }
});

app.get('/api/user/:id', async (req, res) => {
  try {
    const user = await User.findById(req.params.id);
    if (!user) {
      return res.status(404).json({ message: 'User not found' });
    }
    res.json(user);
  } catch (error) {
    res.status(500).json({ message: 'Server error' });
  }
});

app.put('/api/user/:id', async (req, res) => {
  try {
    const { name, email } = req.body;
    const user = await User.findByIdAndUpdate(
      req.params.id,
      { name, email },
      { new: true }
    );

    if (!user) {
      return res.status(404).json({ message: 'User not found' });
    }
    res.json(user);
  } catch (error) {
    res.status(500).json({ message: 'Server error' });
  }
});

app.get('/api/users', async (req, res) => {
  try {
    const users = await User.find().select('-password');
    res.json(users);
  } catch (error) {
    res.status(500).json({ message: 'Server error' });
```

```
  }
});

app.use(express.static('public'));

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```