```css
/* styles.css */
/* Add your custom styles here */
```

```javascript
// registration.js
function registerUser() {
    const username = document.getElementById('username').value;
    const email = document.getElementById('email').value;
    const password = document.getElementById('password').value;

    fetch('/register', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({ username, email, password }),
    })
    .then(response => response.text())
    .then(data => {
        document.getElementById('result').innerText = data;
    })
    .catch(error => {
        console.error('Error:', error);
    });
}
```

```javascript
// app.js
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');

const app = express();

mongoose.connect('mongodb://localhost/user_registration', { useNewUrlParser: true,
useUnifiedTopology: true });

const userSchema = new mongoose.Schema({
    username: String,
    email: String,
    password: String
});

const User = mongoose.model('User', userSchema);

app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json()); // Parse JSON requests

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

app.get('/', (req, res) => {
    res.sendFile(__dirname + '/index.html');
});

app.post('/register', (req, res) => {
```

```javascript
    const newUser = new User({
        username: req.body.username,
        email: req.body.email,
        password: req.body.password
    });

    newUser.save((err) => {
        if (err) {
            res.send('Error registering user.');
        } else {
            res.send('User registered successfully.');
        }
    });
});

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