

```

const express = require('express');
const http = require('http');
const mongoose = require('mongoose');
const cors = require('cors');
const jwt = require('jsonwebtoken');
const bcrypt = require('bcryptjs');
const { Server } = require('socket.io');

// App setup
const app = express();
const server = http.createServer(app);
const io = new Server(server, {
  cors: { origin: '*' }
});

// Middleware
app.use(cors());
app.use(express.json());

// MongoDB connection
mongoose.connect('mongodb://localhost:27017/chatapp', {
  useNewUrlParser: true,
  useUnifiedTopology: true
});

// MongoDB models
const User = mongoose.model('User', new mongoose.Schema({
  username: String,
  password: String,
}));

const Message = mongoose.model('Message', new mongoose.Schema({
  username: String,
  message: String,
  time: String
}));

// Authentication routes
app.post('/api/register', async (req, res) => {
  const { username, password } = req.body;
  const hashed = await bcrypt.hash(password, 10);
  const user = await User.create({ username, password: hashed });
  res.json({ status: 'ok', user });
});

app.post('/api/login', async (req, res) => {
  const { username, password } = req.body;
  const user = await User.findOne({ username });

```

```
if (!user) return res.status(400).json({ error: 'User not found' });

const isPassValid = await bcrypt.compare(password, user.password);
if (!isPassValid) return res.status(400).json({ error: 'Invalid password' });

const token = jwt.sign({ username: user.username }, 'secret123');
res.json({ status: 'ok', token });
});

// Socket.IO chat
io.on('connection', (socket) => {
  console.log('A user connected');

  socket.on('chat message', async ({ username, message }) => {
    const time = new Date().toLocaleTimeString();
    await Message.create({ username, message, time });
    io.emit('chat message', { username, message, time });
  });

  socket.on('disconnect', () => {
    console.log('A user disconnected');
  });
});

// Start server
server.listen(3001, () => console.log('Server listening on http://localhost:3001'));
```