**STEP 1: Folder Structure**

Create your project folder:

bash

CopyEdit

mkdir SkillSync

cd SkillSync

npx create-vite@latest client --template react

mkdir server

**🚀 STEP 2: Backend Setup (server/)**

**✅ 1. Initialize and Install Packages**

bash

CopyEdit

cd server

npm init -y

npm install express mongoose cors dotenv jsonwebtoken bcryptjs body-parser

**✅ 2. Create Basic Files**

**📄 server/index.js**

js

CopyEdit

const express = require('express');

const mongoose = require('mongoose');

const cors = require('cors');

const dotenv = require('dotenv');

dotenv.config();

const authRoutes = require('./routes/authRoutes');

const quizRoutes = require('./routes/quizRoutes');

const resultRoutes = require('./routes/resultRoutes');

const app = express();

app.use(cors());

app.use(express.json());

app.use('/api/auth', authRoutes);

app.use('/api/quizzes', quizRoutes);

app.use('/api/results', resultRoutes);

mongoose.connect(process.env.MONGO\_URI)

.then(() => {

app.listen(5000, () => console.log('Server running on port 5000'));

})

.catch(err => console.error(err));

**✅ 3. Create Models**

**📄 models/User.js**

js

CopyEdit

const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

username: String,

email: String,

password: String,

role: { type: String, default: 'participant' }

});

module.exports = mongoose.model('User', userSchema);

**📄 models/Quiz.js**

js

CopyEdit

const mongoose = require('mongoose');

const quizSchema = new mongoose.Schema({

title: String,

topic: String,

duration: Number,

questions: [{

question: String,

options: [String],

answer: String

}]

});

module.exports = mongoose.model('Quiz', quizSchema);

**📄 models/Result.js**

js

CopyEdit

const mongoose = require('mongoose');

const resultSchema = new mongoose.Schema({

userId: String,

quizId: String,

score: Number,

timeTaken: Number,

submittedAt: { type: Date, default: Date.now }

});

module.exports = mongoose.model('Result', resultSchema);

**✅ 4. Create Routes (Sample)**

**📄 routes/authRoutes.js**

js

CopyEdit

const express = require('express');

const bcrypt = require('bcryptjs');

const jwt = require('jsonwebtoken');

const User = require('../models/User');

const router = express.Router();

router.post('/register', async (req, res) => {

const { username, email, password } = req.body;

const hashed = await bcrypt.hash(password, 10);

const user = new User({ username, email, password: hashed });

await user.save();

res.status(201).json({ message: 'Registered' });

});

router.post('/login', async (req, res) => {

const { email, password } = req.body;

const user = await User.findOne({ email });

if (!user || !(await bcrypt.compare(password, user.password))) {

return res.status(400).json({ error: 'Invalid credentials' });

}

const token = jwt.sign({ id: user.\_id }, process.env.JWT\_SECRET);

res.json({ token, user });

});

module.exports = router;

(Quiz and result routes will follow similar pattern.)

**🌐 STEP 3: Frontend Setup (client/)**

**✅ 1. Install Axios & React Router**

bash

CopyEdit

cd ../client

npm install axios react-router-dom

**✅ 2. Folder Structure (src/)**

css

CopyEdit

client/

└── src/

├── components/

├── pages/

├── services/ // Axios API calls

├── App.jsx

└── main.jsx

**✅ 3. Sample Files**

**📄 services/api.js**

js

CopyEdit

import axios from 'axios';

const API = axios.create({ baseURL: 'http://localhost:5000/api' });

export default API;

**📄 pages/Login.jsx**

js

CopyEdit

import React, { useState } from 'react';

import API from '../services/api';

const Login = ({ setUser }) => {

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const handleLogin = async () => {

const res = await API.post('/auth/login', { email, password });

localStorage.setItem('token', res.data.token);

setUser(res.data.user);

};

return (

<div>

<h2>Login</h2>

<input value={email} onChange={e => setEmail(e.target.value)} placeholder="Email" />

<input value={password} onChange={e => setPassword(e.target.value)} type="password" placeholder="Password" />

<button onClick={handleLogin}>Login</button>

</div>

);

};

export default Login;

**📄 App.jsx**

js

CopyEdit

import React, { useState } from 'react';

import Login from './pages/Login';

function App() {

const [user, setUser] = useState(null);

return (

<div>

<h1>SkillSync</h1>

{!user ? <Login setUser={setUser} /> : <p>Welcome, {user.username}!</p>}

</div>

);

}

export default App;

**🔐 STEP 4: .env File (Server)**

Create .env in the server/ folder:

env

CopyEdit

MONGO\_URI=mongodb://localhost:27017/skillsync

JWT\_SECRET=yourSecretKeyHere

**✅ NEXT STEPS:**

1. Finish routes: /api/quizzes and /api/results
2. Build quiz-taking UI with timer in React
3. Add leaderboard page to frontend
4. Deploy (optional)