# Development Plan (SkillSprint MVP)

**Goal**: Build a working MVP (login → pick a skill → complete daily tasks) within 3 weeks.

### Timeline (week by week tasks)

| Week | Frontend Tasks | Backend Tasks |
|---|---|---|
| **Week 1** | - Set up repo<br>- Design wireframes<br>- Build Landing page, Login/Signup page | - Setup backend server + DB<br>- Build User Authentication (Login/Signup API) |
| **Week 2** | - Build Dashboard page (view current challenge)<br>- Build 30-Day Challenge page + Daily checklist UI | - Create Challenge API (generate and store daily tasks)<br>- Connect frontend with backend (basic fetch data) |
| **Week 3** | - Polish UI (progress, error handling)<br>- Demo Prep | - Finalize API (update user progress, streaks)<br>- Fix bugs, write simple backend documentation<br>- Help frontend connect edge cases (errors, loading states) |

### Roles (who does what)

**Recommend**: use templates — don't build complex admin upload features yet.

```
Frontend
├── Displays pages (Learning Path, Progress, Goals)
└── Calls backend APIs to fetch/save user data

Backend (minimal, use supabase)
├── User auth
├── Save user progress (lesson completion, goals)
└── Save to-do list items
```

### Tech Stack (Tools used)

- **Frontend:** Next.js (React.js) + Tailwind CSS
- **Backend:** Node.js + Express.js + MongoDB (or Supabase/Firebase optional for auth)
- **Tools:** GitHub (version control), Jira (task tracking), Lovable (UI prototyping)
- **Other:** OpenAI API (basic AI prompt generation)

### **MVP Definition of Done** (MVP checklist)

- User can log in/register
- User can pick a skill challenge
- User can view and check off daily tasks
- Progress is saved and visible (progress bar or streak)
- Data stored and retrieved from database
- Clean UI with responsive layout