CURSOR AI WORKSHOP – DAY 3: SETUP, ACTIVITY & DEPLOYMENT

AGENDA

- Environment Setup
- Clone & Install Dependencies
- Run the Project Locally
- Hands-on Activity
- Common Issues & Fixes
- Vercel Deployment
- Q&A

INSTALLATIONS
- Node.js (>=18) – verify: **node --version**
- Install dependencies:
- **npm install**
- **pip install -r requirements.txt**
- Clone & Install Dependencies

RUN THE PROJECT
COMMANDS:

```
npm run dev        # Starts the frontend
```

You are a Senior Frontend Architect building a premium SaaS product.

Create a complete, production-ready College Society Management System frontend in **React + Vite + TypeScript** using:

- Tailwind CSS (v3.4+)
- Framer Motion
- React Icons (lucide-react)
- Context API only (no Redux)
- Recharts for charts
- React Router v6.4+

Core Requirements (do NOT simplify):

1. **Theme System**
   - Light + Dark mode with `dark` class on `<html>`
   - Animated Sun/Moon toggle in navbar (sun rotates & sets, moon rises, 500ms smooth transition)
   - Background gradient transitions smoothly
   - Persist in localStorage
   - Glassmorphism cards (`backdrop-blur-xl bg-white/10 dark:bg-black/10 border border-white/20`)

2. **Design System**
   - 8px spacing grid
   - `rounded-3xl` everywhere
   - Soft shadows + hover glows
   - Inter font (via fontsource or next/font)
   - Neon accent: `#00f5ff` (cyan) + `#a855f7` (purple)

3. **Layout**
   - Collapsible glass sidebar (desktop) → bottom nav (mobile)
   - Top navbar with: logo, search, notifications (pulse), profile dropdown, animated theme toggle
   - Main area with padding that respects sidebar collapse

4. **Pages to build**
   - `/` → Dashboard (animated stat cards, Recharts line + doughnut, recent activity, upcoming events)
   - `/societies` → Grid of society cards with 3D hover tilt (use framer-motion + CSS perspective)
   - `/events` → Event cards with animated capacity bars + status pulse badges
   - `/ai` → AI Recommendation glass card with sparkle animation

5. **Animations (Framer Motion)**
   - Page fade + slide up on route change
   - Card hover: scale 1.03 + lift
   - Sidebar collapse with smooth width animation
   - Button ripple + hover glow
   - Event cards: hover lift + gradient border glow
   - Floating chatbot button with continuous gentle bounce + outer ring pulse

6. **Components (create reusable)**
   - `ThemeToggle.tsx` (fully animated)
   - `GlassCard.tsx`
   - `StatCard.tsx` (with count-up animation)
   - `SocietyCard.tsx` (3D tilt)
   - `EventCard.tsx`
   - `Sidebar.tsx` (collapsible)
   - `Navbar.tsx`

7. **Extras**
   - Fully responsive (mobile-first)
   - Lazy loaded routes
   - Clean folder structure (src/components, src/pages, src/context, src/hooks, etc.)
   - No inline styles, pure Tailwind + clsx
   - Include `tailwind.config.js` with custom colors, fonts, shadows

Generate the **full project** in this exact order (so Cursor can build it step-by-step quickly):

1. Vite + TS + Tailwind setup + folder structure
2. ThemeContext + ThemeToggle
3. Layout (Sidebar + Navbar + Main)
4. Dashboard page
5. Societies page + SocietyCard
6. Events page + EventCard
7. AI page + floating chatbot

Start now.

POTENTIAL ISSUES STUDENTS MAY FACE AFTER RUNNING PROMPTS:

Issue / Error	Cause	Solution / Fix
ModuleNotFoundError or missing React/TS modules	Dependencies not installed (npm install) or wrong Node version	Ensure Node ≥18, run npm install, check package.json dependencies
Cannot find module 'tailwindcss'	Tailwind not installed or postcss config missing	Install Tailwind: npm install -D tailwindcss postcss autoprefixer and npx tailwindcss init -p
Tailwind classes not applying	Tailwind config paths missing or CSS import missing	In tailwind.config.js, include content: ['./index.html','./src/**/*.{ts,tsx}']; import index.css in main.tsx
Dark mode toggle not working	ThemeContext not set up, localStorage missing, or <html> class not updated	Ensure ThemeContext wraps App; use document.documentElement.classList.toggle('dark') and persist in localStorage
Framer Motion animations not showing	Component not wrapped with motion.div or variants not defined	Wrap target elements in motion.div and define initial, animate, exit variants
Recharts charts blank	Data array empty or incorrect	Verify chart data passed as props; check structure for line/doughnut chart
Routing issues (404 on /societies)	React Router not configured or lazy-loaded route missing	Ensure BrowserRouter wraps App, routes defined with Routes + Route and element props, lazy-loaded components use Suspense
Component import errors (Cannot find module)	Wrong file path or missing .tsx	Double-check file structure (src/components/...) and imports; use correct relative paths

GITHUB & GIT PUSH – CURSOR AI PROJECT

Step 1: Initialize & Link Repository

- git init   # Initialize Git in your project
- git add .   # Stage all files
- git commit -m "Day 3: Cursor AI setup"
- git branch -M main   # Ensure main branch
- git remote add origin https://github.com/your-username/cursor-ai.git

Step 2: Push to GitHub

git push -u origin main

Tips:

- First push may require GitHub login / PAT (Personal Access Token)
- Always pull first if repo is not empty: git pull origin main --rebase

Step 3: Commit Workflow

1. Make code changes
2. Stage files: git add <file> or git add .
3. Commit changes: git commit -m "Message"
4. Push to GitHub: git push

Common Issues

Issue	Cause	Solution
Authentication failed	Wrong GitHub password	Use **PAT** or SSH key
Updates were rejected	Remote has commits you don't have	git pull --rebase origin main then push
git: command not found	Git not installed	Install Git & check PATH (git --version)
Large file push blocked	GitHub file size limit	Use **Git LFS** or split large files

VERCEL DEPLOYMENT – CURSOR AI PROJECT

Step 1: Install & Login Vercel CLI

```
npm i -g vercel      # Install Vercel CLI globally
vercel login         # Login with GitHub/Email
```

Step 2: Link Project

```
vercel link          # Link local project to your Vercel account
```

- Connect to your GitHub repo
- Select correct project name and root directory

Deploy Project

```
vercel --prod        # Deploy to production
```

- Vercel will build your React + Vite project
- Live URL will be generated after deploy

Common Issues

Issue	Cause	Solution
Build fails	Incorrect build command or output folder	Ensure Vite build output is dist, set in Vercel project settings
Environment variables missing	API keys required for backend	Add env vars in Vercel dashboard (Project → Settings → Environment Variables)
Frontend shows blank page	Vite base path misconfigured	Check vite.config.ts base path; set base: './' if needed
Deployment stuck / timeout	Backend API slow or misconfigured	Use **serverless functions** or optimize backend calls