CampusConnect – 6-Week Daily Roadmap (2 hrs/day)

This roadmap is structured like corporate sprints — every week ends with a working feature.

# Week 1 – Backend Setup & PDF Parsing

* Day 1: Install & set up Node.js + Express project. Set up GitHub repo (corporate branching: main/dev).
* Day 2: Install & configure PostgreSQL. Create DB tables (users, pdf\_uploads).
* Day 3: Set up Sequelize or Knex for DB interaction. Write sample seed scripts for users.
* Day 4: Add file upload endpoint using Multer.
* Day 5: Integrate pdf-parse to extract student details (Reg No, Name, Branch).
* Day 6: Save parsed student data into the DB as unverified users.
* Day 7: Test end-to-end: Upload PDF → See users in DB.

# Week 2 – Authentication & Profile Management

* Day 8: Add JWT authentication (login/register endpoints).
* Day 9: Implement Nodemailer for email-based OTP.
* Day 10: Add profile claim flow (verify via Reg No + OTP).
* Day 11: Create profile update endpoint (bio, skills, projects).
* Day 12: Add admin routes (approve achievements, upload PDFs).
* Day 13: Write middleware for role-based access (student/admin).
* Day 14: End-to-end test: Freshers can claim & edit profiles.

# Week 3 – Frontend Setup & Core Pages

* Day 15: Initialize React project with Bootstrap.
* Day 16: Build Login & Signup pages (connected to API).
* Day 17: Build Profile Edit page (bio, skills).
* Day 18: Build Student Directory page (search + filters).
* Day 19: Build Admin Panel (upload PDF, verify achievements).
* Day 20: Integrate Axios for API calls & handle JWT in frontend.
* Day 21: Test full flow: Login → Edit Profile → View Directory.

# Week 4 – Unique Features (Skill Graph & Chat)

* Day 22: Add skills table & user\_skills mapping.
* Day 23: Build skill analytics API (count students by skill).
* Day 24: Display Skill Graph (React chart library).
* Day 25: Set up WebSocket server for chat.
* Day 26: Build chat APIs (send/receive messages).
* Day 27: Create chat UI in React (basic 1-to-1 messaging).
* Day 28: Test: Analytics + Chat end-to-end.

# Week 5 – Optimization & Deployment

* Day 29: Add Redis caching for search results.
* Day 30: Optimize queries (indexes on frequently searched columns).
* Day 31: Add input validation & error handling.
* Day 32: Write API documentation (Swagger/Postman).
* Day 33: Dockerize backend & frontend.
* Day 34: Deploy backend (Render/Railway).
* Day 35: Deploy frontend (Vercel/Netlify).

# Week 6 – Polish & Portfolio Prep

* Day 36: Add proper UI/UX touches (loading states, error messages).
* Day 37: Write project README (for GitHub).
* Day 38: Create short video demo of the project.
* Day 39: Conduct code review & cleanup.
* Day 40: Prepare resume bullet points & portfolio write-up.
* Day 41–42: Buffer days for fixing bugs & final testing.