web application for goal-setting and budget tracking in a company.

**1. Define Requirements**

**Core Features:**

* **Goal Management**:
  + Set goals by department.
  + Track goals' progress.
  + Filter goals by status, department, or deadline.
* **Task Management**:
  + Create tasks.
  + Set deadlines.
  + Assign tasks to employees.
  + Track task progress.
  + Filter tasks by assignee, department, or priority.
* **Budget Tracking**:
  + Assign budgets to goals.
  + Monitor budget utilization.
  + Notify managers about overspending.
* **Role-based Access Control (RBAC)**:
  + Roles: Admin, Manager, Employee.
  + Permissions based on roles.
* **Authentication**:
  + Secure login and registration.
  + OAuth integration for SSO (e.g., Google, Microsoft).
* **Dashboard**:
  + Overview of goals, tasks, and budgets.
  + Visual analytics (charts, graphs).
* **Notifications**:
  + Alerts for deadlines, budget overruns, and updates.

**2. Tech Stack**

**Frontend:**

* **Framework**: [React.js](https://reactjs.org/) (component-based, performant, and widely supported).
* **State Management**: Redux Toolkit or React Context API.
* **Styling**: Tailwind CSS or Material-UI (pre-built components for faster development).
* **Charts**: Chart.js or D3.js (for visual analytics).

**Backend:**

* **Framework**: [Node.js](https://nodejs.org/) with Express.js (lightweight and scalable).
* **Database**:
  + **Relational**: PostgreSQL (best for structured data like tasks, budgets, roles).
  + **NoSQL** (optional for flexibility): MongoDB (for unstructured data like notifications or logs).
* **Authentication**:
  + JWT (JSON Web Tokens) for secure session management.
  + Passport.js for OAuth SSO.
* **Role Management**: CASL or custom middleware.

**DevOps & Deployment:**

* **Hosting**: AWS (EC2, RDS, S3) or alternatives like Vercel/Netlify for frontend and DigitalOcean/Render for backend.
* **CI/CD**: GitHub Actions or GitLab CI/CD.
* **Containerization**: Docker for consistency across environments.
* **Monitoring**: Sentry (error tracking) and Prometheus/Grafana (performance monitoring).

**Other Tools:**

* **Version Control**: Git + GitHub/GitLab.
* **Task Management**: Jira or Trello.
* **Testing**:
  + Unit: Jest (frontend and backend).
  + E2E: Cypress.

**3. Development Plan**

**Phase 1: Planning & Setup**

1. Gather detailed requirements from stakeholders.
2. Design database schema (users, roles, tasks, goals, budgets, etc.).
3. Create wireframes and prototypes for the UI.

**Phase 2: Authentication & Authorization**

1. Implement user registration and login with password hashing (bcrypt).
2. Integrate OAuth (e.g., Google Workspace for company accounts).
3. Develop role-based access control middleware.

**Phase 3: Goal Management**

1. Backend:
   * Create APIs to add, update, delete, and fetch goals.
   * Associate goals with departments and budgets.
2. Frontend:
   * Create goal creation forms.
   * Display goals in a list with filters.
   * Add progress visualization.

**Phase 4: Task Management**

1. Backend:
   * Build task CRUD APIs.
   * Integrate tasks with goals and employees.
2. Frontend:
   * Develop task creation and assignment UI.
   * Implement task progress tracking.

**Phase 5: Budget Tracking**

1. Backend:
   * Add APIs to manage budgets for each goal.
   * Calculate remaining and utilized budgets.
2. Frontend:
   * Display budget utilization in charts.

**Phase 6: Notifications**

1. Implement server-side job scheduler (e.g., Node-cron) for alerts.
2. Send email or push notifications for deadlines and overspending.

**Phase 7: Analytics & Dashboard**

1. Backend:
   * Aggregate data for charts and summaries.
2. Frontend:
   * Develop interactive dashboards with analytics.

**Phase 8: Testing & Optimization**

1. Test all APIs and UI components.
2. Optimize database queries and front-end performance.
3. Conduct security audits.

**Phase 9: Deployment**

1. Set up environments (staging and production).
2. Deploy the application.
3. Train users and provide documentation.

**4. Database Schema (Sample)**

**Tables:**

* **Users**: id, name, email, password, role
* **Roles**: id, name (Admin, Manager, Employee)
* **Departments**: id, name
* **Goals**: id, title, description, department\_id, budget, status
* **Tasks**: id, goal\_id, title, assignee\_id, deadline, status
* **Budgets**: id, goal\_id, allocated\_amount, used\_amount

**5. Estimated Timeline**

* **Phase 1**: 1 week.
* **Phase 2**: 2 weeks.
* **Phase 3**: 2 weeks.
* **Phase 4**: 2 weeks.
* **Phase 5**: 1 week.
* **Phase 6**: 1 week.
* **Phase 7**: 2 weeks.
* **Phase 8**: 1 week.
* **Phase 9**: 1 week. **Total**: ~11-12 weeks.