Inter Finder

1. Project Overview

1.1 Objective

Develop **Inter Finder**, a web-based job listing platform designed to connect interns with businesses and individuals seeking internship services. The platform supports two internship types: free internships (requiring client feedback or reviews) and paid internships. The project will be executed by a team of five interns (two backend, two frontend, one UI/UX designer) over two months, delivering a functional minimum viable product (MVP) for internal use and educational purposes.

1.2 Scope

The MVP will include:

- **RESTful API**: For managing internship listings, user registrations, and feedback.
- **User Registration**: Separate registration flows for interns (free/paid options) and clients (businesses/individuals).
- **Internship Listings**: Clients can create, edit, and delete internship postings (free or paid).
- **Application System**: Interns can apply to listings, specifying their internship type preference.
- **Feedback/Review System**: Mandatory feedback or reviews for free internships; optional for paid internships.
- **Search and Filter**: Allow users to search and filter internship listings by type, location, or skills.
- **Dashboard**: Display user-specific data (e.g., intern applications, client postings, feedback).
- **Production-Ready Architecture**: Use minimal dependencies for scalability, excluding complex authentication (e.g., OAuth) or rate limiting for simplicity.
- Deliver a functional Inter Finder MVP within the internship period.

2. Team Structure and Roles

- **Backend Developers (2)**:
    - Build a Node.js/Express.js server with RESTful API endpoints for user management, internship listings, applications, and feedback.

    - Integrate MongoDB with Mongoose for data storage (users, listings, applications, reviews).
- **Frontend Developers (2)**:
    - Develop a React.js frontend for user registration, internship listings, application submission, feedback, and dashboard views.
    - Integrate with backend APIs for seamless data flow.
- **UI/UX Designer (1)**:
    - Design an intuitive, responsive web interface for interns and clients.
    - Create wireframes for registration, listing creation, search, and dashboard pages.
    - Ensure accessibility and user-friendly navigation.

3. Technology Stack

- **Backend**: Node.js with Express.js for the RESTful API, Mongoose for MongoDB integration.
- **Frontend**: React.js with Chakra UI for a responsive interface, axios for API calls, react-router-dom for navigation.
- **Database**: MongoDB for storing user profiles, internship listings, applications, and feedback.
- **Deployment**: Vercel for the React frontend, Heroku for the Node.js backend.
- **Version Control**: GitHub for collaborative development.
- **Project Management Tools**: ClickUp or Jira for task tracking, Slack for communication (aligned with your prior interest in these tools).

Rationale: Node.js and Express.js are beginner-friendly and production-ready, Mongoose simplifies MongoDB interactions, and React with Chakra UI enables rapid frontend development. The stack minimizes dependencies while supporting the required functionality.

4. Project Timeline (8 Weeks)

Week 1: Setup and Planning

- **Tasks**:
    - Set up Node.js/Express.js backend and MongoDB with Mongoose.
    - Set up React.js frontend with Chakra UI and react-router-dom.
    - UI/UX: Create wireframes for registration, listing creation, search, and dashboard pages.
    - Backend: Define MongoDB schemas (e.g., User, Listing, Application, Feedback).
    - Frontend: Build basic homepage with navigation.
    - Team: Set up ClickUp/Jira for task tracking and Slack for communication.
- **Deliverables**: Project setup, GitHub repo, MongoDB connection, wireframes.

Week 2: User Registration

- **Tasks**:
    - Backend: Implement API endpoints:
        - POST /register/intern: Register interns with free/paid option.
        - POST /register/client: Register clients (businesses/individuals).
    - Frontend: Build registration forms for interns (with free/paid toggle) and clients.
    - UI/UX: Design registration UI with clear input fields and validation feedback.
- **Deliverables**: Functional user registration for interns and clients.

Week 3: Internship Listings

- **Tasks**:
    - Backend: Implement API endpoints:
        - POST /listings: Create internship listings (title, description, type, skills, location).
        - PUT /listings/:id: Edit listings.
        - DELETE /listings/:id: Delete listings.
    - Frontend: Create forms for clients to manage listings and display listings for interns.
    - UI/UX: Design listing creation and browsing UI.
- **Deliverables**: Internship listing creation, editing, and deletion functionality.

Week 4: Application System

- **Tasks**:
    - Backend: Implement API endpoints:
        - POST /listings/:id/apply: Submit applications with intern type (free/paid).
    - Frontend: Build an application form for interns to apply to listings.
    - UI/UX: Design application UI with confirmation messages.
- **Deliverables**: Functional application system.

Weeks 5-6: Feedback and Dashboard

- **Tasks**:
    - Backend: Implement API endpoints:
        - POST /feedback: Submit feedback/reviews (mandatory for free internships).
        - GET /dashboard/intern/:id: Fetch intern applications and feedback.
        - GET /dashboard/client/:id: Fetch client listings and received applications.
    - Frontend: Build a dashboard to display:
        - For interns: Applied listings, feedback received.
        - For clients: Posted listings, applications received, feedback submitted.
    - UI/UX: Design dashboard with tabs for interns and clients, using tables or cards.
- **Deliverables**: Feedback system and user-specific dashboards.

Week 7: Search and Filter

- **Tasks**:
  - Backend: Implement GET /listings/search endpoint with query parameters (e.g., type, location, skills).
  - Frontend: Add search bar and filter options (e.g., dropdowns for type, location).
  - UI/UX: Design search UI with intuitive filters and results display.
  - Team: Test end-to-end workflow (register → create listing → apply → submit feedback → view dashboard).
- **Deliverables**: Functional search and filter system, bug-free MVP.

Week 8: Deployment and Documentation

- **Tasks**:
  - Deploy frontend to Vercel and backend to Heroku.
  - Write documentation: setup guide, API endpoints, user instructions, code overview.
  - Prepare and deliver team demo showcasing registration, listings, applications, feedback, and dashboard.
- **Deliverables**: Deployed app, documentation, demo presentation.

5. Expected Deliverables

- A web-based platform with:
  - RESTful API for user registration, internship listings, applications, and feedback.
  - MongoDB storage for users, listings, applications, and reviews.
  - React dashboard for interns and clients.
  - Search and filter functionality for internship listings.
- Deployed application on Vercel/Heroku.
- Documentation and demo presentation.

6. Resources and Support

- **Learning Resources**:
  - Node.js/Express: The Net Ninja Express.js tutorials (YouTube).
  - Mongoose: Mongoose official documentation.
  - React: FreeCodeCamp React course.
  - Chakra UI: Official Chakra UI documentation.
- **Tools**:
  - Figma (UI/UX design).
  - GitHub (version control).
  - MongoDB Atlas (cloud database, free tier).
  - Postman (API testing).

    - ClickUp or Jira (task tracking, per your preference).
    - Slack (team communication).
- **Mentorship**: Weekly check-ins with a senior developer to guide on API development, MongoDB integration, and dashboard design.

7. Challenges and Mitigation

- **Feedback Enforcement**: Ensure mandatory feedback for free internships via backend validation; provide clear UI prompts for clients.
- **Search Performance**: Optimize MongoDB queries for search/filter with indexes; limit results if needed.
- **Team Coordination**: Use ClickUp/Jira for task tracking and Slack for communication, aligning with your prior tool preferences.
- **Time Constraints**: Prioritize registration, listings, and applications if delays occur; simplify feedback UI if needed.

8. Ethical and Legal Considerations

The app is for educational purposes and will not be distributed publicly. Ensure compliance with data privacy (e.g., secure storage of user data) and avoid using real user data without consent during testing.

9. Success Metrics

- Functional MVP delivered within 8 weeks.
- All features (registration, listings, applications, feedback, search, dashboard) implemented and tested.
- Positive feedback from demo on usability and functionality.
- Interns gain experience with Node.js, Express.js, MongoDB, React, and full-stack development.

10. Conclusion

Inter Finder provides a valuable opportunity for the intern team to build a practical job listing platform tailored for internships, using modern web technologies. The focused scope, beginner-friendly tools, and structured timeline ensure a successful MVP delivery in two months, aligning with your interest in collaborative development and project management tools.

Prepared by: [Your Name]
Date: July 1, 2025