**Sprint Backlog Document**

**Project Name: BeBetter**

**Tech Stack: MERN (MongoDB, Express.js, React.js, Node.js)**

**Sprint 1: Setup and Basic Features**

**Tasks:**

1. **Environment Setup:**
   * Initialize React project.
   * Set up Node.js and Express backend.
   * Configure MongoDB Atlas for database storage.
2. **User Registration and Login:**
   * **Frontend:**
     + Create login and signup forms in React.
     + Use form validation libraries like Formik or React Hook Form.
   * **Backend:**
     + Develop REST APIs for user registration and login.
     + Implement bcrypt for password hashing and jsonwebtoken for JWT-based authentication.
3. **Profile Creation:**
   * **Frontend:**
     + Create profile input forms for user details (e.g., interests, goals).
   * **Backend:**
     + Develop an API to save and retrieve user profiles.
     + Use Mongoose to define user profile schemas.

**Deliverable:**

* A functional app skeleton with user login and profile creation capabilities.

**Sprint 2: Dashboard and AI Integration**

**Tasks:**

1. **Personalized Dashboard:**
   * **Frontend:**
     + Design dashboard layout with React components.
     + Integrate basic stats and placeholders for dynamic data.
   * **Backend:**
     + Create APIs to fetch user-specific data (e.g., progress stats).
2. **AI Recommendation Engine:**
   * Integrate OpenAI API or similar for personalized recommendations.
   * Store AI-generated recommendations in MongoDB for future use.
3. **Backend Integration:**
   * Use axios or fetch for API calls between React frontend and Node.js backend.

**Deliverable:**

* A working dashboard displaying static and dynamic user data with initial AI recommendations.

**Sprint 3: Advanced Features**

**Tasks:**

1. **Progress Tracking:**
   * **Frontend:**
     + Add progress visualization using libraries like Chart.js or D3.js.
   * **Backend:**
     + Develop APIs to store and fetch progress tracking data.
2. **Community Connections:**
   * **Frontend:**
     + Implement UI for forums or chatrooms.
   * **Backend:**
     + Use socket.io for real-time communication.
     + Set up APIs for forum posts and replies.
3. **Notifications:**
   * **Backend:**
     + Implement email notifications using nodemailer.
   * **Frontend:**
     + Add toast or snackbar components for in-app alerts.
4. **Content Library:**
   * **Frontend:**
     + Build a searchable UI for articles and resources.
   * **Backend:**
     + Develop APIs for content retrieval from MongoDB.

**Deliverable:**

* Advanced functionality with progress tracking, community engagement, and notifications.

**Sprint 4: Finalization and Deployment**

**Tasks:**

1. **Security and Compliance:**
   * Use helmet and cors for backend security.
   * Enforce HTTPS and secure cookies.
   * Validate user inputs with express-validator.
2. **Responsive Design:**
   * Ensure responsiveness using CSS frameworks like Tailwind or Bootstrap.
   * Test across devices (desktop, tablet, mobile).
3. **Scalability:**
   * Optimize database queries and add pagination for large datasets.
   * Implement caching for frequently accessed data using Redis.
4. **UI/UX Optimization:**
   * Refine UI components for accessibility (e.g., WCAG standards).
   * Use tools like Lighthouse for performance optimization.
5. **Deployment:**
   * **Frontend:** Deploy on Vercel or Netlify.
   * **Backend:** Deploy on Heroku, AWS EC2, or Render.
   * Use MongoDB Atlas for live database hosting.

**Deliverable:**

* A polished, secure, and scalable web application ready for production.