**Project Planning Document - Expense Tracker**

# Product Backlog and Sprint Schedule

| Sprint Functional Requiremet (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
| --- | --- | --- | --- | --- | --- |
| User Authentication | USN-1 | As a user, I can register by entering email, password, and confirmation. | 2 | High | Backend Dev |
| User Authentication | USN-2 | As a user, I will receive a confirmation email after registration. | 1 | High | Backend Dev |
| User Authentication | USN-3 | As a user, I can register through Gmail. | 2 | Medium | Frontend Dev |
| Login | USN-4 | As a user, I can login with email & password. | 1 | High | Frontend Dev |
| Dashboard | USN-5 | As a user, I can view my dashboard with visual charts. | 5 | High | Frontend Dev |
| Transaction Entry | USN-6 | As a user, I can add income and expenses with forms. | 3 | High | Frontend Dev, Backend Dev |
| Data Visualization | USN-7 | As a user, I can see expense trends in a bar chart. | 2 | Medium | Frontend Dev |
| Excel Export | USN-8 | As a user, I can download expense data as an Excel file. | 2 | Medium | Backend Dev |

# Sprint Tracker & Velocity

Sprint Summary Table

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date | Story Points Completed |
| --- | --- | --- | --- | --- | --- |
| Sprint-1 | 12 | 6 Days | 1 March 2025 | 6 March 2025 | 12 |
| Sprint-2 | 12 | 6 Days | 7 March 2025 | 12 March 2025 | 12 |

Velocity = Total Story Points Completed / Number of Sprints = 24 / 2 = 12  
  
  
  
  
  
  
  
  
  
  
  
**Project Planning**

The Expense Tracker project, a full-stack MERN (MongoDB, Express.js, React, Node.js) web application, was planned with a user-centered design approach. The aim was to develop a digital financial assistant to help individuals—particularly students, young professionals, and freelancers—track their income and expenses, visualize financial trends, and manage their budget securely and intuitively.

#### Objectives

* Design a responsive, intuitive user interface using React, Vite, and Tailwind CSS.
* Implement secure authentication with JWT and encrypted passwords.
* Create dynamic dashboards with charts (pie, bar, line) for financial insights.
* Allow users to log, delete, and export financial records (income and expenses).
* Enable Excel import/export for income and expenses.
* Support profile customization (e.g., avatar upload).

#### Key Planning Components

**Scope Definition:**

* Functional Requirements: User registration/login, dashboard visualization, transaction management, file uploads, profile settings.
* Non-Functional Requirements: Responsive design, secure data handling, fast load times, scalability.

**Technology Stack:**

* Frontend: React, Vite, Tailwind CSS, Recharts, Axios
* Backend: Express.js, MongoDB, Mongoose, Multer, JWT, Bcrypt
* Tools: ESLint, moment.js, Cloudflare security script, GitHub for version control

**Milestones & Deliverables:**

* **Project Initialization and Authentication Setup:** The project commenced with setting up the foundational environment for both frontend and backend development. Core components such as routing and layouts (e.g., DashboardLayout, AuthLayout) were established. Simultaneously, user authentication using JWT and bcrypt was integrated to ensure secure access.
* **User Interface and Dashboard Construction:** Emphasis was placed on designing a user-centric dashboard interface. This phase included the development of the SideMenu and Navbar components for navigation, alongside the integration of data visualization through charting libraries.
* **Feature Integration and Backend Connectivity:** Focus shifted to implementing dynamic forms and modals for transaction handling. The backend was expanded with schema definitions (Income, Expense) and RESTful APIs to process user data.
* **Testing, Visualization, and Deployment Enhancements:** The final phase integrated frontend and backend elements to create cohesive visual summaries. Excel import/export features were added, and thorough testing ensured usability across devices and scenarios.

**Team Roles:**

* Frontend Developer: UI implementation, chart integration, routing
* Backend Developer: API design, database schemas, authentication
* UX Designer: Empathy map creation, wireframes, user flows
* QA Tester: Manual testing, performance monitoring, bug logging

**Tools & Resources:**

* Visual Studio Code, Postman, MongoDB Atlas, GitHub, Canva (for wireframes), Google Sheets (for budget tracking)

**Challenges and Risk Mitigation:**

* **Data Privacy:** Addressed using JWT, bcrypt, and CORS.
* **Excel Parsing:** Managed using xlsx and multer, with MIME type filtering.
* **User Experience:** Improved with Tailwind for responsiveness, react-hot-toast for feedback.

**Outcome:** The planning ensured that the Expense Tracker addressed the target users' pain points: lack of visualization, insecure platforms, and complex navigation. Through structured planning, milestone tracking, and use of modern technologies, the project was developed as a robust, user-friendly financial tool.