Story ID	Feature	User Story	Acceptance Criteria	Story Points
US1	Basic Dashboard UI	Aarav Sharma wants to see a basic dashboard layout, So that he can understand the structure of the app.	1.Dashboard has sections for recommendations, trends, and news (placeholders). 2.Layout is clean with a header, sidebar, and main content area. 3.UI is responsive for desktop and mobile.	5
US2	Profile Setup UI	Aarav Sharma wants to input his risk tolerance and budget in a form, So that he can simulate setting up his profile.	1.Form includes dropdown for risk tolerance (Low, Medium, High) and input for budget. 2.Form has a “Save” button that shows a confirmation message (no backend saving yet). 3.Error message shown if budget is invalid (e.g., negative).	3
US3	Recommendation Placeholder UI	Thomas Shelby wants to see a placeholder for recommendations, So that he can visualize where his investment suggestions will appear.	1.Dashboard has a section titled “Recommendations”. 2.Section shows 3 mock recommendations (e.g., “Stock: AAPL, Suggestion: Buy”). 3.Each recommendation has a button labeled “View Details” (non-functional for now).	3
TS1	Front-End Framework Setup	As a developer needs to set up a front-end framework (e.g., React), So that the team can build the UI efficiently.	1.Project is initialized with React (or similar framework). 2.Basic routing is set up for dashboard and profile pages. 3.CSS framework (e.g., Tailwind or Bootstrap) is integrated for styling.	5
US4	User Registration	Aarav Sharma wants to register with his email and password, So that he can access the InvestIQ platform.	1.Aarav can enter email and password on a registration form. 2. System validates email format and ensures password is at least 8 characters. 3.Upon successful registration, Aarav is redirected to login page. 4. Error messages shown if email is in use or inputs are invalid.	3
US5	User Login	Aarav Sharma wants to log in with his email and password, So that he can access his dashboard.	1.Aarav can enter email and password on a login form. 2.System validates credentials and logs Aarav in if correct. 3. Aarav is redirected to the dashboard upon successful login. 4.Error message shown if credentials are incorrect.	2
US6	Save Profile Data	Aarav Sharma wants to save his profile data, So that the system can use it for recommendations.	1.Profile form (built in Sprint 1) now saves data to the backend. 2.Aarav can edit and update his profile. 3.Confirmation message shown after saving.	3
TS2	Backend API Integration	As a developer needs to integrate a free API (e.g., Yahoo Finance) to fetch historical stock data, So that the team can use it for recommendations.	1.Backend fetches historical stock data for 5 sample stocks (e.g., AAPL, GOOGL). 2.Data includes 30 days of historical prices and trends. 3.Data is cached in the database. 4.Error handling for API failures.	5
TS3	Database Schema Setup	Sanjana Reddy needs to create a database schema for users, profiles, and historical data, So that the team can store and retrieve data.	1.Schema includes tables for Users, Profiles, and HistoricalData. 2.Database uses a free solution (e.g., SQLite). 3.Basic CRUD operations implemented for user and profile data.	3
US7	Historical Trends Display	Thomas Shelby wants to see historical trends for recommended stocks, So that he can understand the basis for recommendations.	1.Dashboard displays a line chart showing 30-day price trend for each stock. 2.Chart uses historical data from the database. 3.Chart includes labels for dates and prices.	5
US8	Market News Display	Thomas Shelby wants to see recent market news, So that he can stay informed.	1.Dashboard includes a section with market news headlines. 2.News fetched from a free API (e.g., NewsAPI) and displayed as a list. 3.Each news item includes title, source, and link to the article.	5
US9	Personalized Recommendations	Thomas Shelby wants to see personalized investment recommendations, So that he can make informed decisions.	1.Recommendations replace the placeholder UI from Sprint 1. 2.Recommendations are based on historical trends and user profile (e.g., risk tolerance). 3.If no recommendations are available, a message like “No recommendations at this time” is shown.	5
TS4	Recommendation Logic	As a developer needs to implement a rule-based recommendation system, So that users receive personalized suggestions.	1.Algorithm uses historical trends (e.g., 30-day moving average) and user risk tolerance. 2.Example rule: Recommend stocks with upward trends for low-risk users. 3.Logic is tested with sample profiles and data.	5
TS5	News API Integration	As a developer needs to integrate a free news API (e.g., NewsAPI), So that the team can display market news.	1.Backend fetches news articles related to finance. 2.News data is cached for 24 hours. 3.Error handling for API failures.	3