User Stories for Product Backlog
User Stories for Product Backlog
As a user, I want to verify the authenticity of news articles to ensure the information I consume is reliabl
As a user, I want the front end to seamlessly communicate with the back end to ensure a smoot
As a user, I want to view my activity history on the home page for quick reference.

As a developer, I want to ensure that the back end functions as expected and handles various s
As a data scientist, I want to implement a neural network model in the backend for accurate pre
As a system administrator, I want to manage user information securely in a database.
As a user, I want my activity history to be stored securely and retrievable.
As a data scientist, I want access to quality training data for the neural network model.

Acceptance Criteria
Scenario: Validating News Given: I am on the home page. When: I input a text containing more than 50 words and less than 500 words. Then: The system should accurately determine whether the news is fake or true. Scenario: Text Input Validation Given: I am on the home page. When: I input a text with fewer than 50 words. Then: The system should display an error message indicating that the input is too short.
The front-end components make API requests to the back end. API responses are handled appropriately in the front end. Error handling is implemented to manage communication failures. Data exchanged between front end and back end is secure and follows best practices.
The home page displays a section for user activity history. Each activity entry includes relevant details such as timestamp and type of activity. Users can interact with the history section, e.g., clear history or filter by date.

Comprehensive unit tests cover each backend function. Integration tests validate the interactions between different backend components. Test coverage is regularly reviewed and maintained. Performance testing is conducted to ensure scalability.
The neural network model is integrated into the backend architecture. Model training scripts are implemented and tested. The model's accuracy is regularly monitored and maintained above 90%. The model can handle new data inputs and adapt accordingly.
User data is securely stored in a database. CRUD operations (Create, Read, Update, Delete) are implemented for user data. Passwords are stored securely using hashing and salting techniques. User data can be retrieved and updated as needed.
Activity history data is stored in a dedicated database. The database schema allows for efficient retrieval and querying of historical data. Data retention policies are implemented to manage storage requirements.
Training data is collected and preprocessed to meet model requirements. Data augmentation techniques are applied to enhance the model's generalization. The training dataset is regularly updated to incorporate new patterns and trends.