**Software Requirements Specification (SRS)**

**Excel Analytics Platform**

**1. Introduction**

**1.1 Purpose**

This document outlines the software requirements for the Excel Analytics Platform, a web application designed to allow users to upload Excel files, analyze data, and generate interactive 2D and 3D charts.

**1.2 Scope**

The Excel Analytics Platform will enable users to upload Excel files (.xls/.xlsx), select data columns for analysis, generate various types of charts, download these visualizations, and maintain a history of their analyses. The system will also include an admin panel for user management.

**1.3 Definitions and Acronyms**

* **MERN**: MongoDB, Express.js, React.js, Node.js
* **JWT**: JSON Web Token
* **API**: Application Programming Interface
* **UI**: User Interface
* **SRS**: Software Requirements Specification

**2. Overall Description**

**2.1 Product Perspective**

The Excel Analytics Platform is a standalone web application that provides visualization and analysis services for Excel data. It functions as a comprehensive tool for both casual users and professionals who need to extract insights from spreadsheet data.

**2.2 Product Features**

* User and admin authentication
* Excel file upload and parsing
* Dynamic data mapping for visualization
* Interactive 2D and 3D chart generation
* Chart download functionality (PNG/PDF)
* User dashboard with upload history
* Optional AI-powered data insights

**2.3 User Classes and Characteristics**

1. **Regular Users**: Individuals who upload Excel files, generate charts, and maintain personal dashboards
2. **Admin Users**: System administrators who manage user accounts and monitor platform usage

**2.4 Operating Environment**

* Web browsers: Chrome, Firefox, Safari, Edge (latest 2 versions)
* Responsive design for desktop and tablet devices
* Internet connection required

**2.5 Design and Implementation Constraints**

* MERN stack implementation
* RESTful API architecture
* JWT for authentication
* MongoDB for data storage
* Chart.js and Three.js for visualizations

**2.6 Assumptions and Dependencies**

* Users have basic understanding of data visualization concepts
* Users have properly formatted Excel files
* Reliable internet connection for uploads and visualization rendering

**3. Specific Requirements**

**3.1 External Interface Requirements**

**3.1.1 User Interfaces**

* **Login/Registration Page**
  + Email/password authentication
  + Password reset functionality
* **Dashboard**
  + Upload history
  + Quick access to create new visualizations
  + Profile settings
* **File Upload Interface**
  + Drag and drop functionality
  + File format validation
  + Progress indicator
* **Chart Configuration Screen**
  + Column selection for axes
  + Chart type selection
  + Chart customization options
* **Chart Viewer**
  + Interactive chart display
  + Download options
  + Share functionality
* **Admin Panel**
  + User management
  + System statistics
  + Activity logs

**3.1.2 Hardware Interfaces**

* Not applicable, web-based application

**3.1.3 Software Interfaces**

* MongoDB database
* Optional AI API (Gemini AI or similar)

**3.1.4 Communication Interfaces**

* RESTful API endpoints
* WebSocket for real-time updates (optional)

**3.2 Functional Requirements**

**3.2.1 User Authentication**

* **FR1.1**: Users shall be able to register with email and password
* **FR1.2**: Users shall be able to log in using email and password
* **FR1.3**: Users shall be able to reset their password
* **FR1.4**: The system shall use JWT for maintaining authentication sessions
* **FR1.5**: Users shall be able to log out

**3.2.2 File Management**

* **FR2.1**: Users shall be able to upload Excel files (.xls, .xlsx)
* **FR2.2**: System shall validate file format and size
* **FR2.3**: System shall parse Excel files and extract column headers
* **FR2.4**: System shall store parsed data in the database
* **FR2.5**: Users shall be able to view their upload history
* **FR2.6**: Users shall be able to delete previously uploaded files

**3.2.3 Data Visualization**

* **FR3.1**: Users shall be able to select X and Y axes from available columns
* **FR3.2**: Users shall be able to choose from multiple chart types (bar, line, pie, scatter, 3D column)
* **FR3.3**: System shall generate visualizations based on user selections
* **FR3.4**: Users shall be able to customize chart appearance (colors, labels, etc.)
* **FR3.5**: Users shall be able to save generated charts to their dashboard
* **FR3.6**: Users shall be able to download charts as PNG or PDF

**3.2.4 AI Integration (Optional)**

* **FR4.1**: System shall be able to generate insights from uploaded data
* **FR4.2**: Users shall be able to request AI-powered summary reports
* **FR4.3**: System shall store and display AI-generated insights alongside visualizations

**3.2.5 Admin Features**

* **FR5.1**: Admins shall be able to view all user accounts
* **FR5.2**: Admins shall be able to disable/enable user accounts
* **FR5.3**: Admins shall be able to view system usage statistics
* **FR5.4**: Admins shall be able to manage database storage

**3.3 Non-Functional Requirements**

**3.3.1 Performance**

* **NFR1.1**: File upload processing shall complete within 30 seconds for files up to 10MB
* **NFR1.2**: Chart generation shall complete within 5 seconds
* **NFR1.3**: The system shall support up to 100 concurrent users
* **NFR1.4**: The system shall handle up to 1000 stored files

**3.3.2 Security**

* **NFR2.1**: All communications shall use HTTPS
* **NFR2.2**: Passwords shall be hashed using bcrypt
* **NFR2.3**: JWT tokens shall expire after 24 hours
* **NFR2.4**: The system shall implement rate limiting for API endpoints
* **NFR2.5**: User data shall be isolated and protected from unauthorized access

**3.3.3 Usability**

* **NFR3.1**: The UI shall be intuitive and follow modern design principles
* **NFR3.2**: The system shall be responsive and support various screen sizes
* **NFR3.3**: Error messages shall be clear and actionable
* **NFR3.4**: Help documentation shall be available for all features

**3.3.4 Reliability**

* **NFR4.1**: The system shall have 99% uptime
* **NFR4.2**: The system shall implement data backup mechanisms
* **NFR4.3**: The system shall handle errors gracefully with proper user feedback

**4. User Stories**

**4.1 Authentication**

1. As a new user, I want to create an account so that I can use the platform
2. As a registered user, I want to log in so that I can access my dashboard
3. As a forgetful user, I want to reset my password so that I can regain access to my account

**4.2 File Management**

1. As a user, I want to upload an Excel file so that I can analyze its data
2. As a user, I want to see my upload history so that I can access my previous analyses
3. As a user, I want to delete files I no longer need so that I can manage my storage

**4.3 Data Visualization**

1. As a user, I want to select which columns to use for X and Y axes so that I can create relevant visualizations
2. As a user, I want to choose different chart types so that I can represent my data appropriately
3. As a user, I want to customize my charts so that they match my preferences
4. As a user, I want to download my charts so that I can use them in presentations or reports

**4.4 AI Features**

1. As a user, I want to get AI-generated insights so that I can better understand my data
2. As a user, I want to save AI insights so that I can reference them later

**4.5 Admin Features**

1. As an admin, I want to manage user accounts so that I can maintain system security
2. As an admin, I want to view system statistics so that I can monitor platform usage

**5. Use Cases**

**5.1 UC-01: User Registration**

**Actor**: New User **Description**: User creates a new account **Preconditions**: User is not logged in **Main Flow**:

1. User navigates to registration page
2. User enters email and password
3. System validates input
4. System creates new user account
5. System sends verification email (optional)
6. System redirects user to login page **Alternative Flows**:

* Input validation fails: System displays error message
* Email already exists: System prompts user to log in instead **Postconditions**: User account is created in the system

**5.2 UC-02: User Login**

**Actor**: Registered User **Description**: User logs into the system **Preconditions**: User has registered account **Main Flow**:

1. User navigates to login page
2. User enters email and password
3. System validates credentials
4. System generates JWT token
5. System redirects user to dashboard **Alternative Flows**:

* Invalid credentials: System displays error message
* Forgotten password: System offers password reset option **Postconditions**: User is authenticated and has access to the dashboard

**5.3 UC-03: Upload Excel File**

**Actor**: Authenticated User **Description**: User uploads an Excel file for analysis **Preconditions**: User is logged in **Main Flow**:

1. User navigates to upload section
2. User selects or drags an Excel file
3. User confirms upload
4. System validates file format and size
5. System processes and parses the file
6. System displays confirmation and shows parsed data preview **Alternative Flows**:

* Invalid file format: System displays error message
* File size too large: System prompts user to upload smaller file **Postconditions**: File is stored in the database and associated with user account

**5.4 UC-04: Generate Chart**

**Actor**: Authenticated User **Description**: User creates a visualization from uploaded data **Preconditions**: User has uploaded an Excel file **Main Flow**:

1. User selects a file from their history
2. System displays available columns
3. User selects X and Y axes
4. User chooses chart type
5. User customizes chart appearance (optional)
6. System generates and displays the chart **Alternative Flows**:

* Selected columns are incompatible with chart type: System suggests alternative **Postconditions**: Chart is generated and displayed to the user

**5.5 UC-05: Download Chart**

**Actor**: Authenticated User **Description**: User downloads a generated chart **Preconditions**: User has generated a chart **Main Flow**:

1. User views a generated chart
2. User selects download option
3. User chooses download format (PNG/PDF)
4. System generates file in selected format
5. Browser initiates file download **Alternative Flows**:

* Generation fails: System displays error message **Postconditions**: User has a local copy of the chart

**5.6 UC-06: Request AI Insights (Optional)**

**Actor**: Authenticated User **Description**: User requests AI-generated insights for their data **Preconditions**: User has uploaded an Excel file **Main Flow**:

1. User selects a file from their history
2. User requests AI insights
3. System processes data with AI API
4. System displays generated insights **Alternative Flows**:

* AI processing fails: System displays error message **Postconditions**: AI insights are displayed and saved with the file

**5.7 UC-07: Manage Users (Admin)**

**Actor**: Admin User **Description**: Admin views and manages user accounts **Preconditions**: User is logged in with admin privileges **Main Flow**:

1. Admin navigates to admin panel
2. Admin views list of user accounts
3. Admin selects user to manage
4. Admin enables/disables account or resets password
5. System applies changes **Alternative Flows**:

* None **Postconditions**: User account status is updated

**6. Conclusion**

This SRS document outlines the requirements for the Excel Analytics Platform. It will serve as the foundation for development and testing throughout the project lifecycle. The requirements may be refined based on stakeholder feedback and technical constraints during development.