**Project Planning Document**

**Excel Analytics Platform**

**1. Project Scope**

**1.1 Project Overview**

The Excel Analytics Platform is a MERN stack web application that enables users to upload Excel files, analyze data, and generate interactive 2D and 3D visualizations. The platform includes user authentication, file management, chart generation, and an admin panel.

**1.2 In Scope**

* User registration and authentication system
* Excel file upload and parsing functionality
* Data visualization with multiple chart types
* Chart customization and download options
* User dashboard with upload history
* Admin panel for user management
* Gemini AI-powered insights

**1.3 Out of Scope**

* Mobile application development
* Real-time collaborative editing
* Integration with external data sources beyond Excel files
* Advanced statistical analysis tools
* Custom chart type creation

**2. Technology Stack**

**2.1 Frontend**

* **Framework**: React.js
* **State Management**: Redux Toolkit
* **UI Framework**: Tailwind CSS
* **Chart Libraries**:
  + Chart.js for 2D visualizations
  + Three.js for 3D visualizations
* **Additional Libraries**:
  + Axios for API requests
  + React Router for navigation
  + SheetJS/xlsx for client-side Excel parsing
  + React-to-print or jsPDF for chart downloads

**2.2 Backend**

* **Runtime**: Node.js
* **Framework**: Express.js
* **Database**: MongoDB
* **Authentication**: JWT (JSON Web Tokens)
* **File Handling**: Multer
* **Excel Parsing**: SheetJS/xlsx
* **Optional**: Gemini AI API for AI insights

**2.3 DevOps & Deployment**

* **Version Control**: Git/GitHub
* **Backend Deployment**: Render
* **Frontend Deployment**: Vercel
* **Database Hosting**: MongoDB Atlas

**3. Project Timeline**

The project will be completed over a 5-week period, with each week focused on specific milestones:

**3.1 Week 1: Project Setup & Authentication**

* Project repository setup
* Frontend and backend boilerplate configuration
* Database connection setup
* User authentication implementation (registration, login, JWT)
* Basic dashboard layout

**Deliverables:**

* GitHub repository with initial commit
* Working authentication system
* Basic UI structure

**3.2 Week 2: File Upload & Data Processing**

* Implement file upload functionality
* Develop Excel parsing logic
* Create database schema for storing parsed data
* Build API endpoints for file operations
* Implement file validation

**Deliverables:**

* File upload interface
* Backend APIs for file processing
* Database structure for Excel data

**3.3 Week 3: Chart Generation & Visualization**

* Implement Chart.js and Three.js integration
* Develop dynamic data mapping interface
* Create chart type selection functionality
* Build chart customization options
* Implement chart rendering

**Deliverables:**

* Working visualization system
* Multiple chart type options
* Dynamic data mapping interface

**3.4 Week 4: User Features & AI Integration**

* Implement upload history functionality
* Develop chart download feature
* Create user profile management
* Optional: Integrate AI API for data insights
* Implement chart saving functionality

**Deliverables:**

* Complete user dashboard
* Chart download functionality
* Optional AI insights feature

**3.5 Week 5: Admin Panel & Deployment**

* Develop admin dashboard
* Implement user management for admins
* Perform comprehensive testing
* Fix bugs and optimize performance
* Deploy frontend and backend

**Deliverables:**

* Admin management system
* Fully tested application
* Deployed application on Render and Vercel

**4. Resource Allocation**

**4.1 Human Resources**

* 1 Full-stack Developer (primary)
* Optional: 1 UI/UX Designer for consultation

**4.2 Development Environment**

* Local development machines
* GitHub for version control
* MongoDB Atlas for database hosting
* Render for backend hosting
* Vercel for frontend hosting

**4.3 External Services**

* Gemini AI API (optional)
* MongoDB Atlas
* Render
* Vercel

**5. Risk Assessment & Mitigation**

**5.1 Identified Risks**

| **Risk** | **Probability** | **Impact** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| Complex Excel files causing parsing issues | Medium | High | Implement robust validation and error handling, with clear user feedback |
| Performance issues with large datasets | Medium | High | Implement pagination and lazy loading, limit file size |
| Browser compatibility issues with 3D charts | Medium | Medium | Include fallback to 2D charts, provide browser recommendations |
| API rate limits for AI integration | Low | Medium | Implement caching for AI responses, consider paid API tiers if necessary |
| Security vulnerabilities | Low | High | Implement proper input validation, use secure authentication, regular security reviews |

**5.2 Contingency Plans**

* If Excel parsing proves too complex, consider limiting supported formats or features
* If 3D visualization is problematic, focus on robust 2D charts first
* If AI integration timeline is challenged, move to post-MVP feature

**6. Quality Assurance**

**6.1 Testing Strategy**

* Unit testing for core functionality
* Integration testing for API endpoints
* Manual testing for UI/UX
* Cross-browser testing

**6.2 Performance Metrics**

* Page load time < 3 seconds
* Chart generation < 5 seconds
* File upload and processing < 30 seconds for files up to 10MB

**6.3 Acceptance Criteria**

* All functional requirements from SRS are implemented
* Application passes all tests
* UI/UX is intuitive and responsive
* Performance metrics are met

**7. Communication Plan**

**7.1 Progress Tracking**

* Weekly progress reports
* Daily commit activity
* Issue tracking on GitHub

**7.2 Stakeholder Updates**

* Weekly demos or screenshots of progress
* Regular feedback collection
* Milestone completion notifications

**8. Cost Estimation**

**8.1 Development Costs**

* Developer time: 5 weeks of development
* Optional designer consultation

**8.2 Operational Costs**

* MongoDB Atlas: Free tier initially, scaling as needed
* Render: Free tier initially, scaling as needed
* Vercel: Free tier
* Gemini AI API: Free model

**9. Post-Deployment Plan**

**9.1 Maintenance**

* Regular security updates
* Bug fixes as reported
* Monitoring for performance issues

**9.2 Future Enhancements**

* Mobile application
* Enhanced AI features
* Additional chart types
* Data export options
* Team collaboration features

**10. Conclusion**

This planning document outlines the approach to developing the Excel Analytics Platform over a 5-week period. The project will follow the MERN stack architecture and focus on delivering a user-friendly interface for Excel data analysis and visualization. Regular progress tracking and quality assurance will ensure the project meets all requirements within the specified timeline.