# Study Tool App - MVP and Development Roadmap

## Minimum Viable Product (MVP)

The MVP emphasizes delivering essential features to maximize user value with a streamlined task management interface while minimizing development time:

1. File Upload and Organization:

- Upload Word, PDF, Excel, and image files.

- Categorize files by class and assignment.

- Allow manual entry of assignment due dates.

- Present files and tasks visually with drag-and-drop functionality for reorganization.

2. Summarization:

- Summarize text from both typed files and handwritten notes in uploaded images.

- Limit summaries to key points (e.g., under 150 words).

3. Quiz Generation:

- Auto-generate quizzes with higher-level thinking questions (e.g., application and analysis).

4. Performance Analysis:

- Provide insights into quiz results and identify patterns such as topics frequently missed.

- Generate textual summaries for performance insights.

5. Search and Filtering:

- Allow users to add and use tags for files and assignments to filter and organize by criteria such as class, subject, and due date.

- Offer advanced search options for quick access.

6. Notifications:

- Send reminders for upcoming assignments and quizzes via in-app and email notifications.

- Allow users to customize alert preferences.

7. User Authentication:

- Support email/password login and social login via Google.

8. File Storage:

- Support both local device storage and cloud storage for file uploads.

## Development Roadmap

### Phase 1: Planning and Design

1. Finalize Requirements:

- Define user flows for uploading, tagging, organizing, and summarizing files.

- Specify functionality for task management, notifications, quiz generation, and performance analysis.

2. Design UI/UX:

- Create wireframes for interface design (e.g., file upload page, task boards, dashboard).

- Use tools like Figma or Adobe XD for prototyping and gather feedback.

3. Choose Tech Stack:

- Frontend: React.js (web), React Native (mobile).

- Backend: Node.js with Express.js or Python (Django/Flask).

- Database: PostgreSQL or MongoDB.

- File Storage: AWS S3 or Google Cloud Storage.

- AI Integration: OpenAI API for summarization and quiz generation.

- Notifications: Firebase Cloud Messaging and SendGrid.

### Phase 2: Development

1. Core Infrastructure:

- Set up backend API for authentication, file uploads, and data storage.

- Implement cloud storage integration with local fallback.

2. File Upload and Organization:

- Enable file uploads for Word, PDF, Excel, and image formats.

- Add functionality for assigning tags, due dates, and organizing tasks.

3. Summarization and Quiz Generation:

- Integrate summarization for typed and handwritten notes using OCR.

- Implement quiz generation via OpenAI, focusing on higher-level questions.

4. Performance Analysis:

- Add system to analyze quiz results and highlight areas/topics needing improvement.

- Generate textual summaries for performance insights.

5. Search and Filtering:

- Implement a search bar and tagging filters for files and assignments.

6. Notifications:

- Develop notification settings for reminders via in-app and email.

### Phase 3: Testing and Feedback

1. Alpha Testing:

- Test functionality internally with your kids or a small group.

2. Debug and Refine:

- Fix issues based on test results and feedback.

3. Beta Release:

- Expand testing to a broader audience.

### Phase 4: Deployment

1. Hosting and Deployment:

- Deploy the backend on AWS, Google Cloud, or Heroku.

- Launch the web app and Progressive Web App (PWA).

2. App Store Submission:

- Develop and submit the mobile app for iOS and Android.

### Phase 5: Post-Launch and Iteration

1. Monitor Usage:

- Track user engagement and feature adoption.

2. Iterate on Feedback:

- Improve features based on user input.

3. Monetization Strategy:

- Introduce premium features like advanced analytics and LMS support.

4. Future Enhancements:

- Add gamification elements such as badges and milestones to encourage consistent usage.