

EXP 3: Evaluation of Prompting Tools Across Diverse AI Platforms: ChatGPT, Claude, Bard, Cohere Command, and Meta

Name: ANISH M J

Reg.No: 212221230005

Aim:

- To compare the performance, user experience, and response quality of different AI platforms (ChatGPT, Claude, Bard, Cohere Command, and Meta) within a specific use case, such as summarizing text or answering technical questions.

Instructions:

1. Define the Use Case:

- Select a specific task for evaluation across platforms (e.g., summarizing a document, answering a technical question, or generating a creative story / Code).
- Ensure the use case is applicable to all platforms and will allow for comparison across response quality, accuracy, and depth.

2. Create a Set of Prompts:

- Prepare a uniform set of prompts that align with the chosen use case.
- Each prompt should be clear and precise, ensuring that all platforms are evaluated using the same input.
- Consider multiple prompts to capture the versatility of each platform in handling different aspects of the use case.

3. Run the Experiment on Each AI Platform:

- Input the prompts into each AI tool (ChatGPT, Claude, Bard, Cohere Command, and Meta) and gather the responses.
- Ensure the same conditions are applied for each platform, such as input format, time to respond, and prompt delivery.

- Record response times, ease of interaction with the platform, and any technical issues encountered.

4. **Evaluate Response Quality:**

- Assess each platform's responses using the following criteria:
Accuracy, Clarity, Depth, Relevance

5. **Compare Performance:**

- Compare the collected data to identify differences in performance across platforms.
- Identify any platform-specific advantages, such as faster response times, more accurate answers, or more intuitive interfaces.

Deliverables:

- A comparison table outlining the performance of each platform (ChatGPT, Claude, Bard, Cohere Command, and Meta) based on accuracy, clarity, depth, and relevance of responses.
- A final report summarizing the findings of the experiment, including recommendations on the most suitable AI platform for different use cases based on performance.

Input -

I need to develop a project management app like notion with minimal complexity and design which would be more useful for college students

Chat-GPT:

To develop a project management app like Notion for college students with minimal complexity and design, consider these key features and guidelines:

Key Features

1. **User-Friendly Interface:**

- Simple navigation and clean layout.
- Dark and light themes for better visibility.

2. **Task Management:**

- Create, edit, and delete tasks.
- Set due dates and reminders.
- Prioritize tasks (e.g., high, medium, low).
- 3. **Project Organization:**
 - Create and categorize projects or subjects.
 - Use tags or labels for better organization.
- 4. **Collaborative Features:**
 - Allow users to invite peers to collaborate on projects.
 - Share notes and tasks.
- 5. **Note-Taking:**
 - Simple text editor for taking notes.
 - Support for images, lists, and links.
- 6. **Calendar Integration:**
 - Visual calendar to see deadlines and tasks.
 - Sync with Google Calendar or other calendar apps.
- 7. **Templates:**
 - Provide templates for common college projects (e.g., essays, group projects).
- 8. **Mobile Compatibility:**
 - Ensure a responsive design for easy access on mobile devices.
 - Consider developing a mobile app version.

Design Guidelines

- **Minimalist Design:** Focus on essential features without overwhelming users.
- **Color Palette:** Use calming colors to reduce visual strain, making it easier to focus.
- **Typography:** Use clear and readable fonts for better readability.
- **Feedback Mechanisms:** Provide visual feedback (like color changes) when tasks are completed or modified.

Development Tips

- **MVP Approach:** Start with a Minimum Viable Product, focusing on core functionalities, and gather user feedback for improvements.
- **User Testing:** Involve potential users in testing to understand their needs and preferences.
- **Technology Stack:** Consider using React or Vue.js for the frontend and Node.js or Python (Flask/Django) for the backend.

CLAUDE AI :

I'll help you design a simplified project management app tailored for college students, focusing on essential features and user experience. I'll create a user-friendly design using a Mermaid diagram to outline the core structure.

College Project Management App Structure

Click to open diagram

Let me explain the key design principles and features:

1. Core User Interface:
 - Clean, minimalist design with a sidebar navigation
 - Dark/light mode toggle for different study environments
 - Mobile-responsive layout for on-the-go access
2. Key Features for Students:
 - Dashboard: Shows upcoming deadlines, current projects, and quick access to recent files
 - Project/Assignment Tracker:
 - Simple status options: Not Started, In Progress, Review, Complete
 - Priority levels: High, Medium, Low
 - File attachments for course materials and submissions
 - Calendar Integration:
 - Visual timeline of assignments and deadlines
 - Class schedule integration
 - Exam date highlighting
3. Productivity Features:
 - Quick Notes: Rapid note-taking during classes
 - Task Lists: Simple checkboxes with due dates
 - File Organization: Course-wise folder structure
 - Reminder System: Customizable notification settings
4. Unique Student-Focused Elements:
 - Study Session Timer
 - Grade Tracker
 - Course Material Organization
 - Collaboration Options for Group Projects
5. Design Philosophy:
 - Minimalist interface to reduce distractions
 - Focus on speed and efficiency
 - Intuitive navigation with minimal learning curve
 - Essential features only to avoid overwhelm

Would you like me to elaborate on any of these aspects or create a more detailed view of a specific feature?

BARD:

Creating a Minimal Notion-Like Project Management App for College Students

Understanding the Core Needs of College Students

Before diving into development, let's identify the key features that would be most beneficial for college students:

- Task Management:
 - Clear task creation and organization
 - Prioritization features (e.g., deadlines, labels)
 - Ability to set reminders or notifications
- Note-Taking:
 - Simple note creation and formatting
 - Attachment capabilities (e.g., PDFs, images)
 - Option to link notes to specific tasks
- Collaboration:
 - Shared project spaces
 - Real-time collaboration (e.g., commenting, editing)
 - Permission settings to control access
- Calendar Integration:
 - Sync tasks with a personal or academic calendar
 - Visual representation of deadlines and commitments

Technology Stack Recommendations

Given the need for a minimal and efficient app, consider the following technology stack:

- Frontend:
 - Framework: React or Vue.js for a modern, component-based approach
 - UI Library: Material-UI or Bootstrap for pre-built components and a familiar look and feel
- Backend:
 - Framework: Node.js for scalability and ease of use
 - Database: MongoDB for a flexible, NoSQL database that can handle unstructured data (e.g., notes, attachments)
- Cloud Platform: Firebase or AWS Amplify for hosting, authentication, and cloud storage

Key Features and Implementation

1. Task Management:
 - Use a Kanban board or list view for task organization
 - Implement drag-and-drop for easy task reordering
 - Allow users to set due dates, priorities, and add labels
2. Note-Taking:
 - Provide a rich text editor for formatting
 - Enable attachment of files (e.g., PDFs, images)
 - Allow users to link notes to specific tasks
3. Collaboration:
 - Create shared project spaces with different permission levels
 - Implement real-time commenting and editing
 - Use a version control system to track changes
4. Calendar Integration:
 - Integrate with popular calendar apps (e.g., Google Calendar, Outlook)
 - Sync task deadlines and due dates to the calendar

Additional Considerations

- **Mobile Compatibility:** Ensure the app is responsive and works well on both desktop and mobile devices.
- **Offline Functionality:** Consider implementing offline capabilities for users who may not always have internet access.
- **User Testing:** Conduct user testing with college students to gather feedback and make improvements.
- **Security:** Implement appropriate security measures to protect user data.

By focusing on these core features and using a streamlined technology stack, you can create a project management app that effectively addresses the needs of college students while maintaining a minimal and user-friendly design.

Would you like to delve deeper into any specific aspect of the development process, such as the frontend framework, backend database, or cloud platform?

META :

Creating a project management app like Notion, tailored for college students, can be an exciting project. To simplify the design and features, focus on the essential functionalities that cater to students' needs. Here's a suggested roadmap:

Core Features:

1. Note-taking and organization
2. Task management (to-do lists, deadlines)
3. Project collaboration (group assignments)
4. Calendar integration (academic schedule)
5. File sharing and storage

Simplified Design:

1. Clean and minimalistic UI
2. Easy navigation (tabbed interface)
3. Prominent search bar
4. Customizable dashboard (widgets)

Student-centric Features:

1. Course management (syllabus, assignments)
2. Study group formation

3. Peer-to-peer messaging
4. Integration with popular academic tools (e.g., Google Classroom)
5. Mobile optimization

Technical Requirements:

1. Front-end: React, Angular, or Vue.js
2. Back-end: Node.js, Express, or Django
3. Database: MongoDB, PostgreSQL, or Firebase
4. Authentication: OAuth (Google, Facebook) or email/password

Development Roadmap:

Phase 1 (4-6 weeks)

1. Research and planning
2. UI/UX design
3. Front-end development (basic features)

Phase 2 (8-12 weeks)

1. Back-end development (API, database integration)
2. Feature implementation (task management, collaboration)

3. Testing and debugging

Phase 3 (4-6 weeks)

1. Polish UI/UX
2. Implement student-centric features
3. Testing and iteration

Resources:

1. Developers: 2-3 (front-end, back-end, full-stack)
2. Designers: 1 (UI/UX)
3. Project manager: 1

Timeline:

Approximately 20-30 weeks (~5-7 months)

Budget:

Dependent on development team size, technology stack, and location.

Monetization Strategies:

1. Freemium model (basic features free, premium features paid)
2. Subscription-based (monthly/yearly)
3. Advertising (non-intrusive)

Competitive Analysis:

Research existing solutions:

1. Notion
2. Trello
3. Asana
4. Evernote
5. Google Classroom

Identify gaps and areas for differentiation.

Next Steps:

1. Refine your idea and create a detailed specification document.
2. Assemble a development team or partner with a development agency.
3. Begin designing and building your MVP (Minimum Viable Product).

PERPLEXITY AI :

To develop a project management app tailored for college students with minimal complexity and design, consider the following key features and strategies based on existing alternatives to Notion:

Key Features

1. User-Friendly Interface

- **Simplicity:** Ensure the app has an intuitive layout that minimizes the learning curve. A clean design helps users focus on tasks without distractions.
- **Mobile Compatibility:** Optimize for mobile devices, as students often use smartphones for quick access.

2. Task Management

- **To-Do Lists:** Allow users to create and manage simple to-do lists for daily tasks.
- **Kanban Boards:** Implement a visual board system for tracking project progress, which is especially helpful for group projects.

3. Collaboration Tools

- **Real-Time Editing:** Enable multiple users to collaborate on documents simultaneously, similar to features in Nifty and Nuclino.
- **Comments and Mentions:** Incorporate commenting features where users can discuss tasks or documents directly within the app.

4. Document Management

- **Centralized Document Storage:** Provide a space for students to upload and organize their notes, assignments, and project files.
- **Templates:** Offer pre-built templates for common college projects (e.g., research papers, presentations) to streamline the setup process.

5. Calendar Integration

- **Deadline Tracking:** Include a calendar feature that allows students to visualize deadlines and important dates, helping them manage their time effectively.

Development Strategies

1. Minimal Viable Product (MVP) Approach

- Start with core functionalities that address the most pressing needs of college students—task management and collaboration—before adding more complex features.

2. Feedback Loop

- Engage with potential users (students) during development to gather feedback on features and usability. This will ensure the app meets their needs effectively.

3. Pricing Model

- Consider a freemium model that allows basic features for free, with optional paid upgrades for advanced functionalities. This approach can attract more users initially.

4. Marketing Focus

- Highlight how the app simplifies project management specifically for students, emphasizing ease of use and collaboration capabilities.

Conclusion

Creating a project management app for college students requires a focus on simplicity, collaboration, and effective task management. By leveraging insights from existing tools like Nifty and Nuclino, you can design an application that meets the unique needs of this demographic while maintaining a minimalistic approach.

In short, ChatGPT's comprehensive, structured, and flexible design makes it the ideal choice for developers seeking a robust foundation for App creation.