### ****Table Design****

### ****User Collection****

| Field Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| \_id | ObjectId | PRIMARY KEY, AUTO-GENERATED | Unique identifier for each user |
| name | String | REQUIRED | User's full name |
| email | String | UNIQUE, REQUIRED | User's email address |
| password\_hash | String | REQUIRED | Hashed password for authentication |
| social\_media\_id | String | NULL | Linked social media ID (if applicable) |
| created\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Account creation date and time |
| updated\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Last update date and time |

### ****Event Collection****

| Field Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| \_id | ObjectId | PRIMARY KEY, AUTO-GENERATED | Unique identifier for each event |
| user\_id | ObjectId | REQUIRED | Reference to the user who created the event |
| name | String | REQUIRED | Event name |
| date | Date | REQUIRED | Event date |
| time | String | REQUIRED | Event time (formatted as string) |
| location | String | REQUIRED | Event location |
| guest\_count | Integer | REQUIRED | Expected number of guests |
| event\_type | String | REQUIRED | Type of event (e.g., party, meeting) |
| created\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Event creation date and time |
| updated\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Last update date and time |

### ****Task Collection****

| Field Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| \_id | ObjectId | PRIMARY KEY, AUTO-GENERATED | Unique identifier for each task |
| event\_id | ObjectId | REQUIRED | Reference to the associated event |
| description | String | REQUIRED | Task description |
| assigned\_to | ObjectId | REQUIRED | User ID of the person assigned to the task |
| status | String | REQUIRED | Task status (e.g., pending, completed) |
| due\_date | Date | NULL | Task due date |
| created\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Task creation date and time |
| updated\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Last update date and time |

### ****Vendor Collection****

| Field Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| \_id | ObjectId | PRIMARY KEY, AUTO-GENERATED | Unique identifier for each vendor |
| name | String | REQUIRED | Vendor name |
| service\_type | String | REQUIRED | Type of service provided (e.g., catering, AV) |
| location | String | REQUIRED | Vendor location |
| experience | String | NULL | Vendor experience description |
| pricing | Decimal | REQUIRED | Pricing information |
| profile\_picture | String | NULL | URL or path to vendor's profile picture |
| created\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Vendor registration date and time |
| updated\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Last update date and time |

### ****Booking Collection****

| Field Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| \_id | ObjectId | PRIMARY KEY, AUTO-GENERATED | Unique identifier for each booking |
| user\_id | ObjectId | REQUIRED | Reference to the user making the booking |
| vendor\_id | ObjectId | NULL | Reference to the booked vendor (if applicable) |
| service\_id | ObjectId | NULL | Reference to the booked service (if applicable) |
| event\_id | ObjectId | REQUIRED | Reference to the associated event |
| booking\_date | Date | REQUIRED | Booking date and time |
| status | String | REQUIRED | Booking status (e.g., confirmed, cancelled) |
| total\_amount | Decimal | REQUIRED | Total amount for the booking |
| payment\_method | String | REQUIRED | Payment method used |

### ****Review Collection****

| Field Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| \_id | ObjectId | PRIMARY KEY, AUTO-GENERATED | Unique identifier for each review |
| vendor\_id | ObjectId | REQUIRED | Reference to the reviewed vendor |
| user\_id | ObjectId | REQUIRED | Reference to the user who wrote the review |
| rating | Integer | REQUIRED | Rating given (e.g., 1-5) |
| comment | String | NULL | Review comment |
| created\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Review creation date and time |

### ****Service Collection****

| Field Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| \_id | ObjectId | PRIMARY KEY, AUTO-GENERATED | Unique identifier for each service |
| service\_type | String | REQUIRED | Type of service (e.g., catering, AV, logistics) |
| name | String | REQUIRED | Name of the service provider |
| description | String | REQUIRED | Description of the services offered |
| location | String | REQUIRED | Service provider's location |
| pricing | Decimal | REQUIRED | Pricing information |
| availability | Boolean | REQUIRED | Availability status (true/false) |
| picture | String | NULL | URL or path to service provider's picture |
| created\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Service addition date and time |
| updated\_at | Date | DEFAULT CURRENT\_TIMESTAMP | Last update date and time |

### ****Service Details Collection****

| Field Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| service\_id | ObjectId | REQUIRED | Unique identifier for each service detail |
| detail\_key | String | REQUIRED | Key for the detail (e.g., "cuisine\_type", "equipment") |
| detail\_value | String | REQUIRED | Value for the detail |

### ****Feasibility Study for Nexus****

#### ****1. Introduction****

**Nexus** is a comprehensive event management platform designed to streamline the planning and execution of events by integrating multiple services into a single, user-friendly interface. The platform incorporates features for managing events, booking vendors, and handling various services like catering, audio-visual teams, sweets and desserts, and logistics.

#### ****2. Technical Feasibility****

**2.1 Technology Stack**

* **Frontend:** Web-based interface using frameworks such as React, Angular, or Vue.js for a responsive user experience.
* **Backend:** Node.js with Express.js to handle server-side logic and API requests.
* **Database:** MongoDB for flexible and scalable data storage.
* **Authentication:** OAuth for social media login and standard username/password authentication with hashing.
* **Payment Processing:** Integration with payment gateways like Stripe or PayPal.
* **Machine Learning:** Future integration for recommendation engines, price prediction, and sentiment analysis.

**2.2 Database Design**

* **Collections:** User, Event, Task, Vendor, Booking, Review, Service, and Service Details.
* **Data Types:** Adjusted for MongoDB, using ObjectId for primary keys, String for text fields, Decimal for pricing, Date for timestamps, and Boolean for availability status.

**2.3 Integration**

* **External Services:** Integration with third-party APIs for payment processing, email notifications, and possibly SMS alerts.
* **Machine Learning:** Future plans to integrate ML models for personalized recommendations, price predictions, and sentiment analysis.

#### ****3. Operational Feasibility****

**3.1 User Experience**

* **Intuitive Interface:** User-friendly design for event creation, vendor booking, and task management.
* **Mobile Compatibility:** Responsive design to ensure accessibility from various devices (desktops, tablets, smartphones).

**3.2 Scalability**

* **Database:** MongoDB's schema-less nature allows for flexible and scalable data management.
* **Server:** Node.js and Express.js provide a scalable backend architecture capable of handling increasing traffic and data volume.

**3.3 Security**

* **Authentication:** Secure user login with optional 2FA and encrypted password storage.
* **Data Protection:** HTTPS for secure data transmission and regular security audits.

#### ****4. Financial Feasibility****

**4.1 Cost Estimation**

* **Development Costs:** Initial costs for development include hiring developers, purchasing necessary software licenses, and possibly cloud hosting fees.
* **Operational Costs:** Ongoing costs include server maintenance, database management, payment gateway fees, and customer support.

**4.2 Revenue Model**

* **Service Fees:** Commission on vendor bookings and rental items.
* **Premium Features:** Subscription model for advanced features or enhanced functionalities.
* **Advertising:** Potential revenue from featured listings or advertisements on the platform.

**4.3 Funding Sources**

* **Investors:** Seeking investment from venture capitalists or angel investors.
* **Grants:** Potential funding from grants or innovation competitions.

#### ****5. Schedule Feasibility****

**5.1 Development Timeline**

* **Phase 1: Planning and Design (1 months)**
  + Requirement gathering, wireframing, and finalizing tech stack.
* **Phase 2: Development (2-4months)**
  + Backend and frontend development, database setup, and initial integration.
* **Phase 3: Testing (2 week)**
  + Functional, performance, and security testing.
* **Phase 4: Deployment (1 week)**
  + Final deployment, user training, and initial launch.
* **Phase 5: Post-Launch Support (continue after the launch)**
  + Continuous monitoring, bug fixes, and feature updates.

**5.2 Milestones**

* **Project Kickoff**
* **Completion of Initial Development**
* **Beta Release**
* **Official Launch**
* **Feature Enhancements and Updates**

#### ****6. Legal and Compliance Feasibility****

**6.1 Data Privacy**

* **Compliance:** Adherence to GDPR, CCPA, and other relevant data protection regulations.
* **User Consent:** Obtaining user consent for data collection and processing.

**6.2 Terms of Service**

* **Agreements:** Clear terms of service and privacy policy for users and vendors.
* **Liability:** Liability clauses to protect the platform against potential disputes.

#### ****7. Conclusion****

**Nexus** presents a feasible project with significant potential to streamline event planning and management. The integration of multiple services into a single platform, combined with future machine learning enhancements, positions Nexus as a powerful tool for both users and vendors. The project’s technical, operational, financial, and legal aspects are well-considered, supporting a robust plan for successful implementation and growth.

### ****Questionnaire for Nexus****

#### ****1. General Information****

1. **What is the primary objective of the Nexus platform?**
   * Streamline event planning, integrate multiple service bookings, etc.
2. **Who are the target users for Nexus?**
   * Event planners, individual users, vendors
3. **What types of events will Nexus cater to?**
   * Corporate events, weddings, parties,Small scale events etc.

#### ****2. User Requirements****

1. **What are the key features you expect in the User Login and Authentication module?**
   * Social media login, 2FA, session management
2. **What specific functionalities should be included in the Event Management module?**
   * Event creation, task management, communication tools
3. **How should the Vendor Marketplace module function?**
   * Vendor registration, search and filtering, booking and payment
4. **What types of services should be included under the Rental Inventory Management module?**
   * Catering, audio-visual equipment, sweets and desserts, bakers,rentals,logistics ,photographers & videographers etc.

#### ****3. Technical Requirements****

1. **Which technologies or frameworks do you prefer for the frontend and backend development?**
   * React, Node.js, Express.js
2. **Are there any specific requirements for the database design or structure?**
   * MongoDB schema design, indexing
3. **What are your expectations for machine learning integration in the future?**
   * Recommendation engine, price prediction, sentiment analysis
4. **What external services or APIs need to be integrated with Nexus?**
   * Payment gateways, email services, SMS services(optional)

#### ****4. Security and Compliance****

1. **What are your requirements for user data security and privacy?**
   * Data encryption, compliance with Digital Personal Data Protection Act, 2023 (DPDPA),GDPR (General Data Protection Regulation),CCPA (California Consumer Privacy Act),APPI (Act on the Protection of Personal Information)
2. **What should be included in the terms of service and privacy policy?**
   * User consent, data usage, liability clauses
3. **What measures will be taken to ensure platform security?**
   * Regular security audits, secure authentication methods

#### ****5. Operational Considerations****

1. **What is the expected timeline for the project development phases?**
   * Planning, development, testing, deployment
2. **What are your expectations for post-launch support and maintenance?**
   * Bug fixes, feature updates, user support
3. **What scalability considerations should be taken into account for future growth?**
   * Database scaling, server load management

#### ****6. Financial Aspects****

1. **What is the estimated budget for the development and initial launch of Nexus?**
   * Development costs, operational costs, marketing
2. **What are your planned revenue models for Nexus?**
   * Service fees, subscriptions, advertising
3. **What potential funding sources are being considered for the project?**
   * Investors, grants, crowdfunding

#### ****7. Legal and Compliance****

1. **What legal requirements must Nexus comply with?**
   * Data protection laws, business regulations
2. **What are the key components of the user agreements and contracts with vendors?**
   * Service agreements, dispute resolution
3. **How will Nexus handle liability and risk management?**
   * Insurance, legal safeguards

#### ****8. Additional Information****

1. **Are there any specific design or usability preferences for the user interface?**
   * Design aesthetics, user experience
2. **What are the most critical success factors for the Nexus project?**
   * User satisfaction, vendor adoption, market penetration
3. **Do you foresee any potential challenges or risks for the project?**
   * Technical hurdles, market competition

**Questionnaire Source**

* **Name:**
* **Position/Role:**
* **Organization:**
* **Contact Information:**

**Date:** August 18, 2024