**🧱 Technology Stack Breakdown**

Here’s how you can structure your technology stack for this use case:

**1. 💻 Frontend (User Interface Layer)**

| **Component** | **Technology** | **Purpose** |
| --- | --- | --- |
| **Service Portal** | ServiceNow Service Portal | User-friendly web UI for expense submission and reports |
| **Now Mobile App** | ServiceNow Mobile Studio | Mobile interface for submitting and viewing expenses |
| **Custom UI Pages (optional)** | HTML + Jelly or GlideForm APIs | Enhanced UI/UX if needed beyond portal widgets |
| **Widgets** | AngularJS (ServiceNow widgets) | Used to create dynamic components like charts or forms |

**2. 🛠️ Backend (Application Logic Layer)**

| **Component** | **Technology** | **Purpose** |
| --- | --- | --- |
| **Business Rules** | GlideScript (JavaScript) | Validation and triggers on form submissions |
| **Script Includes** | GlideScript | Reusable server-side logic for expense calculations |
| **Flow Designer** | No-code/low-code | Automations: alert workflows, approvals, reminders |
| **Workflows (optional)** | ServiceNow Workflow Engine | For multi-step processes like approval chains |
| **Access Control Rules** | ServiceNow ACL | Role-based access control for users and admins |

**3. 📊 Data Layer (Storage and Structure)**

| **Component** | **Technology** | **Purpose** |
| --- | --- | --- |
| **Expense Table** | Custom ServiceNow Table (x\_expenses) | Stores each family expense entry |
| **Budget Table** | Custom Table | Tracks monthly limits per category |
| **User Table** | sys\_user | Stores family member user info and roles |
| **Reports Table** | Custom or system table | Stores monthly summaries or trends |

**4. 🔁 Automation & Integration Layer**

| **Component** | **Technology** | **Purpose** |
| --- | --- | --- |
| **Notifications** | Notification Engine + Email/SMS | Sends alerts for budget thresholds |
| **Scheduled Jobs** | GlideScheduledScript | For daily/weekly report generation |
| **Integration (optional)** | REST APIs or IntegrationHub | Connect to external apps like Google Sheets or bank APIs |
| **Performance Analytics** | ServiceNow PA | For trend analysis and KPI tracking (optional if licensed) |

**5. 🔐 Security Layer**

| **Component** | **Technology** | **Purpose** |
| --- | --- | --- |
| **Roles & Permissions** | ACLs, Roles (admin, member) | Restrict access to data/views |
| **Data Policies** | ServiceNow | Enforce required fields or rules |
| **Audit Logs** | Platform Logs | Track data changes or user activity |

**6. 🎨 Optional Tools for Development & Design**

| **Tool** | **Use Case** |
| --- | --- |
| **Lucidchart / Draw.io** | For mapping workflows and diagrams |
| **Figma / Adobe XD** | UI wireframes/mockups for Service Portal |
| **VS Code + ServiceNow CLI** | For scoped app development and version control |
| **GitHub (optional)** | Version control of scripts or documentation |

**📦 Sample Summary Table (Condensed View)**

| **Layer** | **Tech/Tool** |
| --- | --- |
| Frontend | Service Portal, Now Mobile, Widgets |
| Backend | Flow Designer, Business Rules, Scripts |
| Data | Custom Tables, sys\_user, GlideRecord |
| Automation | Notifications, Scheduled Jobs |
| Integration | IntegrationHub, REST APIs (optional) |
| Security | ACLs, Roles, Audit Logs |

**✅ Final Deliverable: Your Technology Stack Document Should Include**

* Component summary (as above)
* Architecture diagram (optional)
* Notes on licensing (e.g., PA or IntegrationHub)
* Future enhancements (AI insights, external integrations, etc.)