**ROADMAP OF GIGHUB PROJECT**

**Phase 1: Problem Understanding & Industry Analysis**

This initial phase establishes the project's foundation by defining its purpose, scope, and key stakeholders.

* **Project Charter**: A formal document will be created to act as the project's "north star", outlining the following:
  + **Project Title**: GigHub
  + **Industry**: Professional Services or any industry utilizing freelance talent.
  + **Problem Statement**: The current process of managing freelancers with spreadsheets is chaotic and inefficient , leading to difficulties in finding skilled talent, manual handoffs from sales , and a lack of real-time profitability insights for executives.
  + **Proposed Solution**: A custom Salesforce application to centralize and automate the management of freelance projects.
  + **Key Use Cases**: Automating project creation from a won deal, searching for freelancers by skill, assigning freelancers to projects, and reviewing project profitability.
* **Stakeholders**: The primary users of the application are:
  + **Resource Manager**: Finds and assigns freelancers.
  + **Salesperson**: Manages client deals in Opportunities.
  + **Executive**: Reviews dashboards for profitability.
  + **System Administrator**: Builds and maintains the system.
* **Core Business Process**: The application will support the flow: A deal is won -> A project is created -> A freelancer with specific skills is found and assigned -> The project is completed -> Profitability is reviewed.

**Phase 2: Org Setup & Configuration**

This phase involves preparing the Salesforce Developer Org for the application build.

* **Company Information**: Ensure company name, address, and time zone are correctly set up in Setup -> Company Information.
* **User Profiles**: Create three new profiles by cloning the "Standard User" profile:
  1. Resource Manager
  2. Salesperson
  3. Executive
* **Users**: Create at least three sample users, one for each new profile (e.g., Ria Manager, Sam Sales, Eva Executive).
* **Role Hierarchy**: Establish a simple role hierarchy (e.g., CEO > VP of Sales > Sales Team and CEO > VP of Services > Resource Manager Team) and assign the sample users to their respective roles.

**Phase 3: Data Modeling & Relationships**

This phase creates the custom objects that form the application's data foundation.

* **Custom Objects**: Create the following custom objects, enabling reporting and activities where specified:
  + **Project**: To track project details.
  + **Freelancer**: To store freelancer information.
  + **Skill**: To maintain a list of skills.
  + **Freelancer Skill**: A junction object to link Freelancers and Skills.
* **Fields & Relationships**:
  + **On Project**: Create fields for Status (Picklist), Project Revenue (Currency), Total Cost (Currency), Start Date (Date), End Date (Date), a lookup to Account (Client), and a lookup to Freelancer (Assigned Freelancer).
  + **On Freelancer**: Create fields for Email, Hourly Rate (Currency), Status (Picklist), and a lookup to Contact.
  + **On Freelancer Skill**: Create two master-detail relationships: one to Freelancer and one to Skill, forming a many-to-many relationship.

**Phase 4: Process Automation (Admin)**

This phase brings the application to life with declarative automation.

* **Flow 1 (Auto-Create Project)**: A record-triggered flow that fires when an Opportunity's stage is updated to "Closed Won." The flow will create a new Project record and map the Opportunity's Amount to Project Revenue and the Account to the Project's Client field.
* **Flow 2 (Update Freelancer Status)**: A record-triggered flow that fires when a Freelancer is assigned to a Project. This flow updates the Freelancer's status field to "On Project."
* **Screen Flow ("Find a Freelancer" Tool)**: A screen flow that allows a user to select a skill from a picklist , finds all available freelancers with that skill , and displays them in a data table.
* **Validation Rule**: A rule on the Project object that prevents the status from being changed to "Completed" if the Assigned\_Freelancer\_\_c field is blank.

**Phase 5: Apex Programming (Developer)**

This phase outlines potential future enhancements that would require custom code.

* **Potential Enhancement 1 (Apex Trigger)**: For more complex profitability calculations, an Apex trigger could be written on the Project object to calculate the Total\_Cost\_\_c field.
* **Potential Enhancement 2 (Batch Apex)**: A nightly Batch Apex job could be created to automatically change a freelancer's status back to "Available" if their assigned project's end date has passed.

**Phase 6: User Interface Development**

This phase focuses on creating a positive user experience.

* **Create the GigHub App**: Build a new Lightning App named "GigHub." Add tabs for Home, Projects, Freelancers, Skills, Accounts, Contacts, Reports, and Dashboards to the navigation. Assign the app to the custom profiles.
* **Customize Page Layouts**: Organize fields and add related lists (like "Freelancer Skills" on the Freelancer page) for a logical user interface.
* **Activate the Screen Flow**: Embed the "Find a Freelancer" screen flow directly onto the Project Lightning Record Page for easy access by Resource Managers.

**Phase 7: Integration & External Access**

This phase outlines potential future integrations with external systems.

* **Potential Enhancement 1 (REST API)**: A custom REST API could be created to allow an external company website to pull and display a list of available skills from the Skill object.
* **Potential Enhancement 2 (Platform Events)**: Platform Events could be used to broadcast a message when a new Project is created, allowing external systems like an HR application to subscribe and trigger their own processes.

**Phase 8: Data Management & Deployment**

This phase involves populating the org with data and outlining the deployment strategy.

* **Data Import**: Prepare sample data in CSV files for Skills, Freelancers, and Freelancer Skills. Use Data Loader to insert the records in the correct order (parents first, then the junction object linking them).
* **Deployment Concept**: All development (objects, fields, flows, etc.) should be done in a Sandbox org first. The completed components would then be moved to the Production org using Change Sets.

**Phase 9: Reporting, Dashboards & Security Review**

This phase focuses on data visualization and securing the application.

* **Custom Report Types**: Create a custom report type that links Freelancers with Freelancer Skills and Skills to allow for comprehensive reporting on which freelancers have which skills.
* **Reports**: Build reports such as a "Freelancer Skill Inventory" and a "Project Status Report" grouped by status.
* **Dashboard**: Create a "GigHub Management Dashboard" with charts and graphs using the reports created in the previous step.
* **Security Review**:
  + **Object Permissions**: Edit the custom profiles to set the correct object-level access (e.g., Executive has Read-only).
  + **Field-Level Security**: Hide sensitive fields like Hourly\_Rate\_\_c from profiles that do not need to see it.

**Phase 10: Final Presentation & Demo Day**

The final phase involves preparing and delivering a demonstration of the completed application.

* **Prepare a Narrative**: Structure the presentation as a story that explains the business problem (the chaos of spreadsheets) and presents the GigHub app as the solution.
* **Demonstrate the Core Process**: Walk through the main business process during the demo: show a Salesperson closing a deal, see the Project automatically created, log in as the Resource Manager to use the "Find a Freelancer" tool, assign the freelancer, and finally, log in as the Executive to show the updated dashboard.
* **Document Work**: Take screenshots of the data model from Schema Builder, the flows, and the final dashboard to use in a portfolio and for handoff documentation.