Service Charter Document

Sequelize Model Designer

Purpose

To **automate** the process of writing code for **creating models and relationships** between those models for **Sequelize ORM.**

Objective

To make the process of writing code for model creation and database relationship mapping:

- Time saving
- Precise and accurate result
- Well structured and managed

Demand / Opportunity

Its **opportunity** as till now no such application is available in the market and also its required by most of the developers who are working in backend/database and need to create long and complex models.

Business Requirement:

- **User Authentication**
  - Users who are having keys named username and password for opening the lock named login, allowing them to enter into the website
- **Model Generation**
  - Providing the name of model, attributes to create the base model for a particular database table
- **Relationship mapping**
  - Creating associations for the base models generated
- **Model Reusability**
  - Base model created in the past can be used again with the changes needed

Technical Requirement:

- **User Authentication**
  - UI and API Routes for registration, login, forget password, reset password connected to the database
  - Middleware for session handling
- **Model Generation**
  - UI consisting of dropdowns and input boxes for attribute inputs and backend logic for generating the model and saving it in the db
- **Relationship mapping**
  - UI consisting of dropdowns for attribute select from the models and selecting the type of relationship

- **Model Reusability**
    - Past model duplication and editing the same

Technological Requirement:

- **Frontend**
    - React Js
    - Tailwind Css
- **Backend**
    - Express Js
- **Database**
    - SQL → MySQL
- **Tools**
    - Visual Studio Code [Code IDE]
    - GIT & GitHub [version control]
    - Postman [api testing]

Stakeholder:

Stakeholder	Name	Count
Developers	Srivaths Iyer Amrik Bhadra Riddhi Dethe Ramani Vemula	4
DB Designers	Riddhe Dethe Ramani Vemula	2
Testers	Shruti Deokar Aaradhya Chaudhari	2
Cloud Service Provider	AWS Cloud	1
Investor	TBD	0

Resources Needed:

- **Documentation**
    - React Js
    - Tailwind Css
    - Sequelize
    - Express Js
- **Cloud Account**
    - AWS
- **Human Resource**

Role	Count	Name
UI/UX Designer	2	Amrik Bhadra Ramani Vemula
Frontend Developer	2	UI Development - Amrik Bhadra Ramani Vemula API Integration - Srivaths Iyer Riddhi Dethe
Backend Developer	3	API Development - Amrik Bhadra Riddhi Dethe API Testing - Ramani Vemula
DB Designer	2	Riddhi Dethe Srivaths Iyer
Project Management	4	Srivaths Iyer Amrik Bhadra Ramani Vemula Riddhi Dethe
Documentation	4	Creator - Srivaths Iyer Ramani Vemula Reviewer - Amrik Bhadra Riddhi Dethe
Tester	2	Shruti Deokar Aaradhya Chaudhari
Deployment	2	Amrik Bhadra Riddhi Dethe

PESTEL Analysis:

- **Political:** No political impact on our project
- **Economic:**
    - Economical for users: Rise in economic as time for writing the code and debugging will be saved which can be utilised to do more task or for business logic
    - Employability:
        - Companies can cut down developer resources, reducing cost
- **Social**:
    - High adoption potential as developers prefer automation
    - useful for junior developers, as they can efficiently generate error free code
- **Technological**:
    - Automates Sequelize Model generation and database design, reducing manual effort and improving efficiency.
    - The application relies on Sequelize ORM, cloud databases, and external APIs. If these services change pricing, update policies, or discontinue support, it may impact functionality.
- **Legal**:

    - Open-source **Sequelize license** allows usage but should be checked before commercialization.
    - **ISO 20000** compliance when launching as commercial Saas Service
- **Environmental**: No environmental impact due to our application

Risk Analysis:

Risk	Description	Mitigation
Software compatibility	Sequelize keeps on upgrading its versions,	Periodical review to check new updates
Database Server Downtime	Unable to connect to database instance and fetch data	Maintain a secondary database as a replica of the primary database and promote it to the primary database in case of failure.
Cloud Server Downtime	Service disruption due to cloud outages or network failures, even due to natural calamities.	Use multi-region deployments, so that if server running in one region goes down, others are still working
Over Budget	1. Extra cost due to extended development time. 2. Rise in cost of third party services	1. Implement strict budget planning, cost monitoring, and optimize resource usage. 2. Purchasing long term service plan
Security Risks	Weak authentication or improper access control could expose sensitive data.	Hash the password using the MD5 algorithm and enhance security with two-factor authentication (2FA)
Scalability Challenges	High number user requests can cause breakdown of the application	Enable auto-scaling and load balancing on EC2 instances to efficiently manage fluctuating user requests and ensure optimal performance.

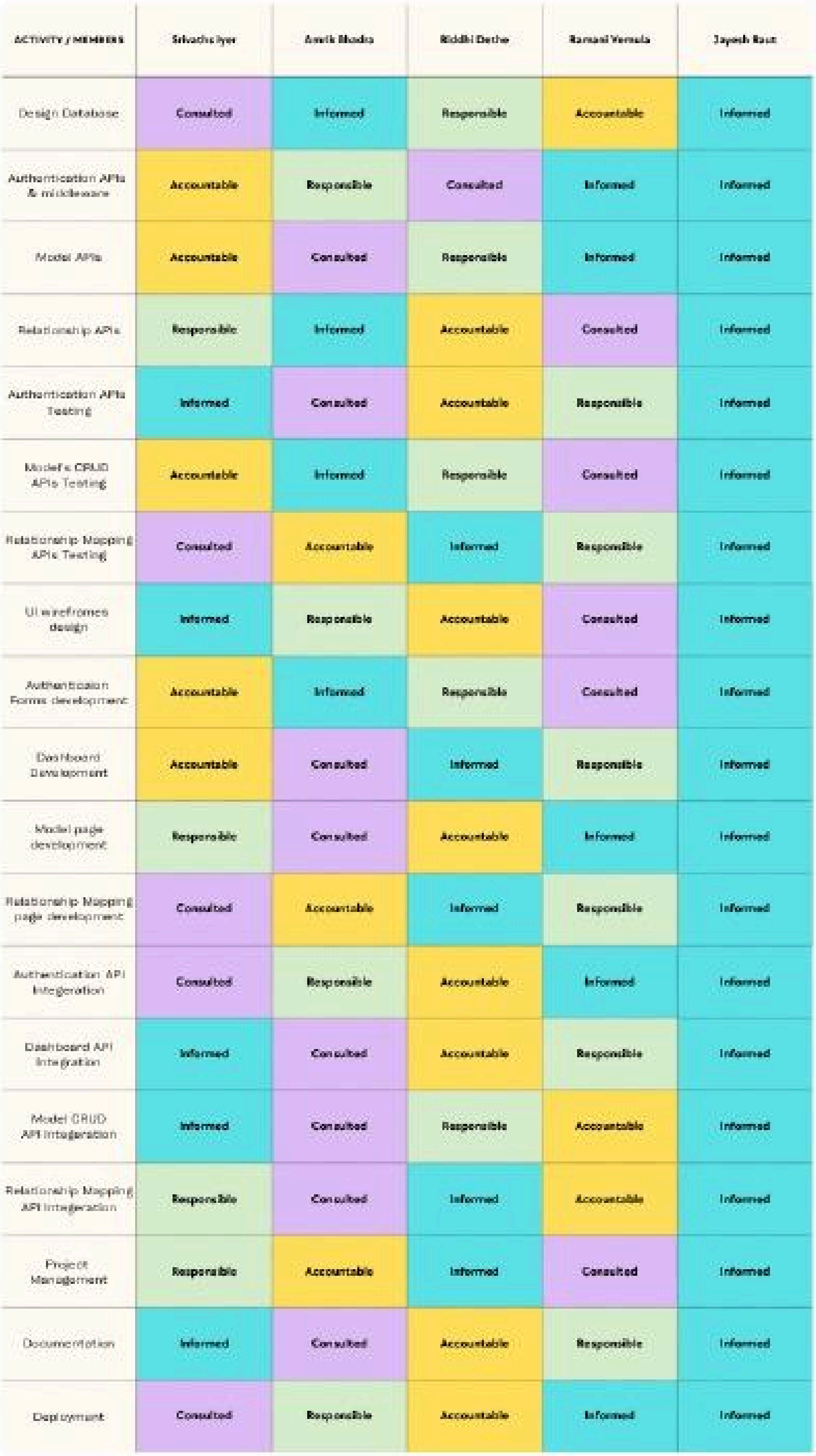
Timeline / Milestone:

Phase	Milestone	Tasks	Timeline
1	Requirement Analysis & Planning	- Gather and Document project requirements - Define & construct functional flow of the application - Identify third-party services and dependencies - Design UI mockups and wireframes	Week 1
2	Database Design and API Development	- Design database schema - Set up database configurations	Week 2

		- Set up middleware and API structure - Develop API endpoints - Implement user authentication setup - Perform API testing using postman	
2	UI Development and API Integration	- Develop UI components and pages - Implement state management - Connect frontend with backend - Implement error handling & validation - Optimize API calls	Week 3
4	Testing & Debugging	- Perform UI/UX testing - Perform API integration testing - Fix bugs and optimize performance	Week 4
5	Deployment & Final Review	- Set up cloud hosting (AWS) - Implement CI/CD pipelines - Monitor and resolve deployment issues - Conduct final review and documentation	Week 4

RACI Chart:

https://www.canva.com/design/DAGgjUN72ic/twQfAphPF18JdNUPNTGL7g/view?utm_content=DAGgjUN72ic&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=had65b5ec8



ACTIVITY / MEMBERS	Srivats Iyer	Amik Bhadra	Riddhi Dethe	Ramani Vemula	Jayesh Raut
Design Database	Consulted	Informed	Responsible	Accountable	Informed
Authentication APIs & middleware	Accountable	Responsible	Consulted	Informed	Informed
Model APIs	Accountable	Consulted	Responsible	Informed	Informed
Relationship APIs	Responsible	Informed	Accountable	Consulted	Informed
Authentication APIs Testing	Informed	Consulted	Accountable	Responsible	Informed
Model's CRUD APIs Testing	Accountable	Informed	Responsible	Consulted	Informed
Relationship Mapping APIs Testing	Consulted	Accountable	Informed	Responsible	Informed
UI wireframes design	Informed	Responsible	Accountable	Consulted	Informed
Authenticaion Forms development	Accountable	Informed	Responsible	Consulted	Informed
Dashboard Development	Accountable	Consulted	Informed	Responsible	Informed
Model page development	Responsible	Consulted	Accountable	Informed	Informed
Relationship Mapping page development	Consulted	Accountable	Informed	Responsible	Informed
Authentication API Integeration	Consulted	Responsible	Accountable	Informed	Informed
Dashboard API Integration	Informed	Consulted	Accountable	Responsible	Informed
Model CRUD API Integeration	Informed	Consulted	Responsible	Accountable	Informed
Relationship Mapping API Integeration	Responsible	Consulted	Informed	Accountable	Informed
Project Management	Responsible	Accountable	Informed	Consulted	Informed
Documentation	Informed	Consulted	Accountable	Responsible	Informed
Deployment	Consulted	Responsible	Accountable	Informed	Informed

Budget:

Service	**Monthly Cost (INR)**
AWS EC2	567.07
AWS RDS for MySQL	7119.80
AWS API GateWay	3053.46
Total	**10740.33**