



# Story Sello

Ecommerce Project

Submitted to : Dr. Reena Maam

Created by: Bhomic Kaushik, 08

# Contents

---

01 Idea

02 Features : Must Have

03 Features : Good To Have

04 Technology to be used





# Idea

We will create a platform that will allow writers to write content and a gist of that content will be shown. The magazine owners/publishers/readers can then look for the gist of content on this website and if they found it interesting they can pay for it and the writers will be then able to create some profit by writing. This platform will not be limited to english stories and will also allow hindi stories to be sold as well... including hindi / english poems... in order to encourage writers.



## Features : Must Have

- Browse without Login
- Authentication Page :
  - Login/Sign up via Email
  - Login/Sign up via Phone Number
- List Product/Service Page
  - Search Bar
- Product/Service Detail Page
- My Cart
  - payment gateway
- Email notification system
  - For updates as well as sending the soft copy of the content purchased
- Profile Page for Writers
  - Writer's bio and their works



# Features : Good To Have



- Star Ratings for different products by users
- Recommender system based on the previous searches and purchases by users
- Job Postings for Writers [In demand content by publishers]
- Desktop Alerts [whenever any writer writes a new story]
- Collaboration of multiple writers to work on a single story/ content

# Revenue Model

- We will charge the users as subscription fees for publishing their content on the platform.
- We will charge some extra fees from the users who are buying the content.



# Technologies to be used:

React [For Scaleable UI]

HTML

CSS

JS

Mongo DB/Firebase

Node Js [For API's: express]

Figma [Website Wireframing]

POSTMAN [API testing]

temp-mail.org [Dummy Email Service]



# Roles

## Admin

Manages the users/  
payments/ shipping/  
Job postings

## Writer

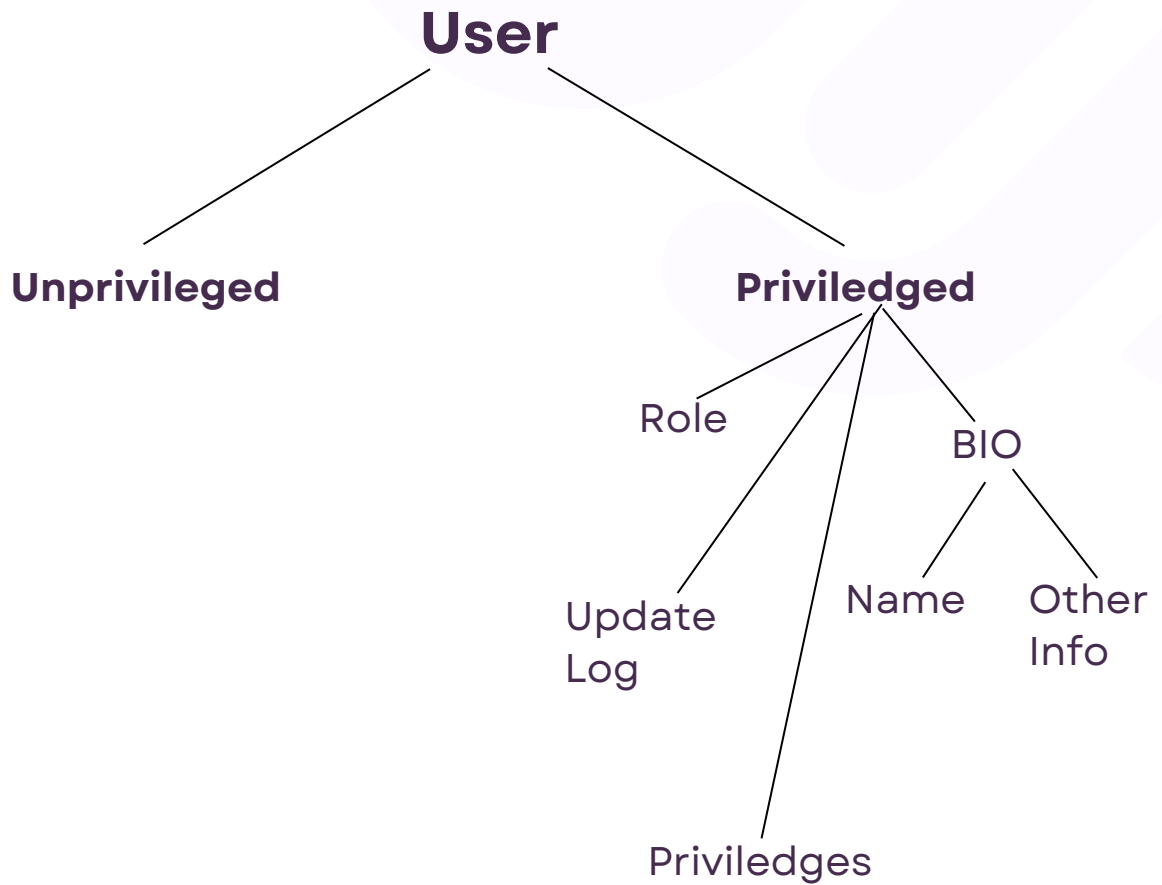
Can write story /  
content and display a  
gist of same on platform

## Reader/ Publisher

Can purchase a content  
as a reader or as a  
publisher

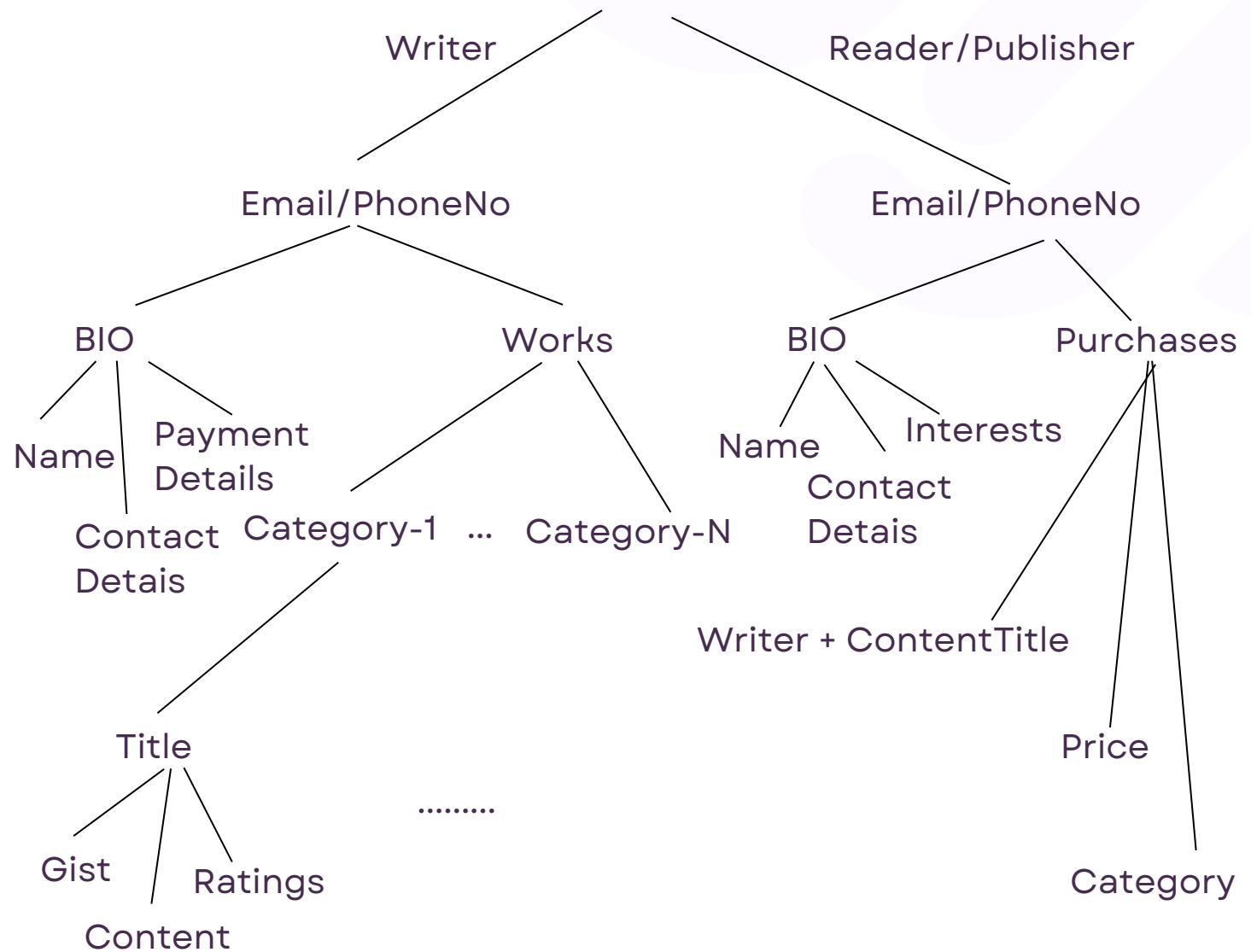


# Design



# Design

## Unpriviledged



# API's

## Authentication

- `Check_if_already_registered()`
  - To check if a user making account is already registered, if yes: ask to login instead
- `Get_user_details()`
  - Get the user details to display on page
- `Create_a_new_user()`
  - Register a new user with details supplied
- `Login()`
  - Set the session variables to contain the logged in user details and redirect to same page
- `Logout()`
  - Remove the user details from session and redirect to main page

# API's

## Main Page

- **Get\_Latest\_Content():**
  - To get the latest content submitted by writers
- **Search\_Content\_by\_Filter():**
  - Returns the content results based on category/ author/ date/ popularity
- **Check\_Login():**
  - Check if user is logged in
    - if(logged in): return the user details + content based on preferences
    - else: return normal content

# API's

## Content Page

- **Get\_Content\_Details():**
  - To get the details about the content including the gist of the content, author details, ratings.
- **Get\_Reviews():**
  - To retrieve the reviews given by the users for a particular content
- **Add\_to\_cart():**
  - To add the content to the cart present in the session also in the database cart in order to buy later
- **Buy\_now():**
  - Redirect to the cart page for payment and shipping options

# API's

## Payment Page

- **Get\_Price():**
  - Retrieve the price of the Product based on Writer and Title
- **Get\_User\_address():**
  - Retrieve the User's address if entered earlier or ask to enter the address
- **Get\_Payment\_Details():**
  - To fetch the payment details if user has done payment earlier also
- **Place\_Order():**
  - Make an entry in database to place order of a content and proceed for further action like Sending Content via email/ Hard Copy
- **Send\_Notification():**
  - Fetch User's Email and Status Details and send via email ID and Desktop Notifs

# API's

## Content Posting Page

- **Fetch\_In\_Demand\_Themes():**
  - To retrieve the themes in demand according to the job postings created
- **Fetch\_Contents():**
  - To Fetch the writer's previous set of works if submitted earlier with their details like ratings and reviews
- **Post\_Content():**
  - Submit the content + gist in user's content's section with 0 rating and reviews
- **Get\_Stats():**
  - Get the statistics of the user's content like total purchases, average rating, subscribers

# API's

## Job Posting Page

- **Post\_Job():**
  - To post the demanded content supplied with category detail along with deadline and price offered
- **Search\_Writer():**
  - Search the content writer who fulfill the demand according to category of content and past writer ratings



# Database Schema

In case RDBMS is used

authentication\_table

## -Columns :

- email\_id [Key] (varchar(50))
- name (varchar(20))
- phone\_no (number) [unique]
- user\_type [Writer/ Reader] (text(20))

## -Description:

This table will be used to store the details for the authentication system which will be used to login the users to our platform and on basis of the usertype the users will be redirected to their respective pages

# Database Schema

In case RDBMS is used

content\_table

## -Columns :

- content\_id [key] (varchar(20))
- email\_id (varchar(50)) [foreign-key]  
references authentication\_table[email\_id]
- content [text]
- date [date]
- price [decimal(10,2)]
- upvotes [number]

**-Description:** This table will store the contents published by the writers on the platform and on the basis of those various gists will be shown on the main page so that readers can buy it

# Database Schema

In case RDBMS is used

purchases\_table

## -Columns :

- email\_id (varchar(50)) [foreign key]  
references authentication\_table[email\_id]
- date [date]
- type [soft/hard] varchar(20)
- delivered\_or\_not [yes/no] varchar(20)

**-Description:** This table will be used to keep a track of the purchases made by readers on the platform after reading the gist. It will store the email\_id of the buyers which will be taken by logged in details, date of purchase will also be stored along with hard/ soft copy purchased information.. and if hard then delivered\_or\_not to keep a track that whether the content has been successfully emailed/ sent to the buyer

# Database Schema

In case RDBMS is used

alerts\_subscriber\_list

## -Columns :

- email\_id (varchar(20))
- phone\_number (number) [NULL]
- subscribed (varchar(20)) [yes/no]

**-Description:** This table will keep a list of the users who have subscribed to the platform for the alerts of the new content published on the platform

# Thank You



Github : <https://github.com/bhomic-GJ/StorySelloGJ>