**Documentation(Whole Group Contributed)  
  
Must-Have (Big Features):  
(**Note: We don't need to incorporate all of the must-have features; we can select three to focus on. I’ve included the full list here for reference)  
**Our 3 big features/functions:**I think all of the must-have features are quick and simple to implement, except for 1. Customization for Front End, 2. Levels for Back End, and 3. Unique elements that determine interactions for Back End. (Farron’s group comment)

**Front End:**

1. Customization (profiles, gameplay duration tracking, achievements, login functionality).
2. Game Description (instructions on how to play the game and reasons to enjoy it).
3. A logo (a visually appealing one).
4. Community Section (a space for communication, similar to Discord, with both chat and interaction).

**Back End:**

1. Levels (must include 1-3 distinct levels).
2. A cool menu with customizable settings (options to retry or skip levels).
3. Unique elements that determine interactions (create our own elements).

**Nice-to-Haves:**

**Front End:**

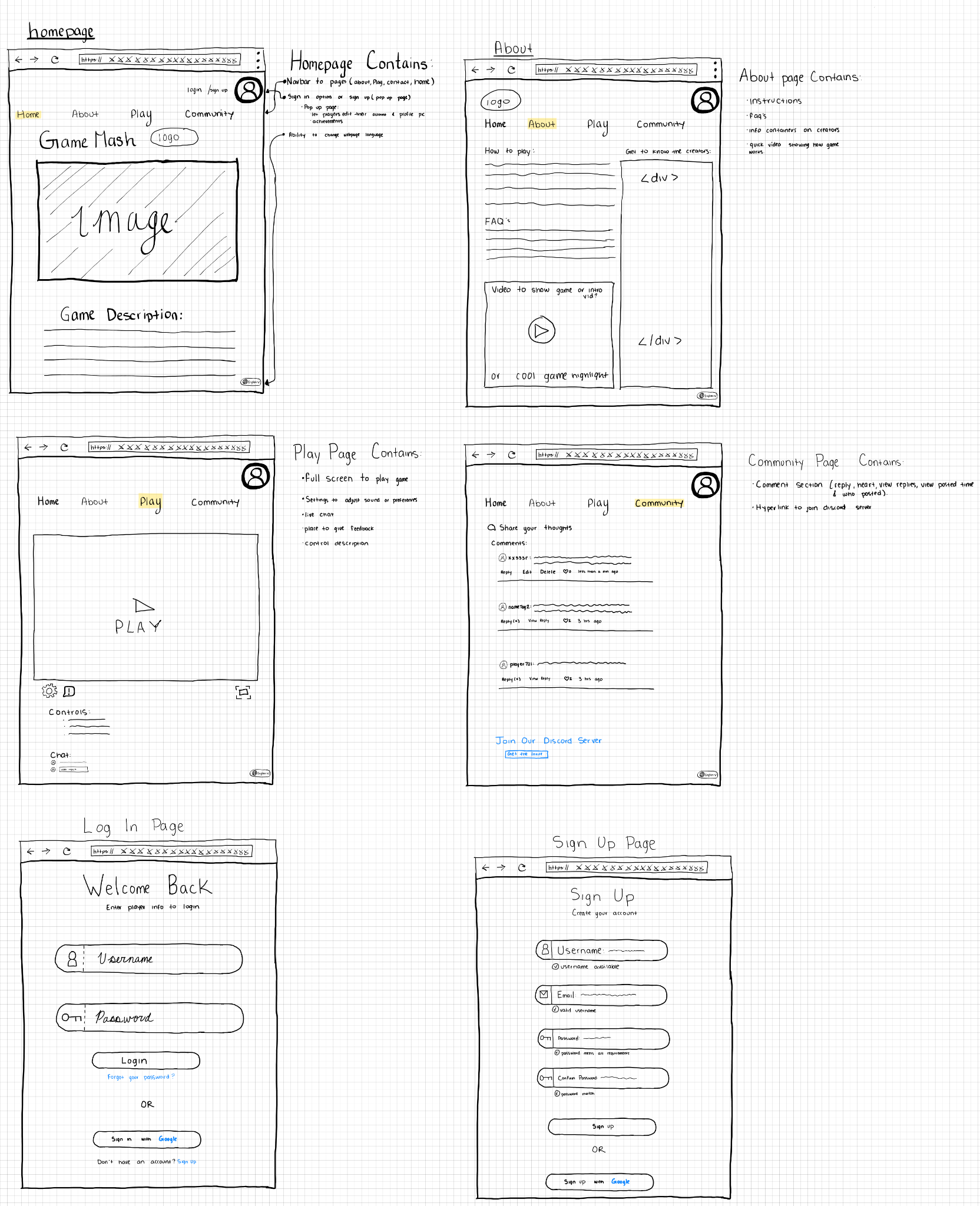
* An attractive UI.
* The ability to play any game regardless of the user’s background (e.g., language preferences).
* Additional services (emotes and online chat functionality).

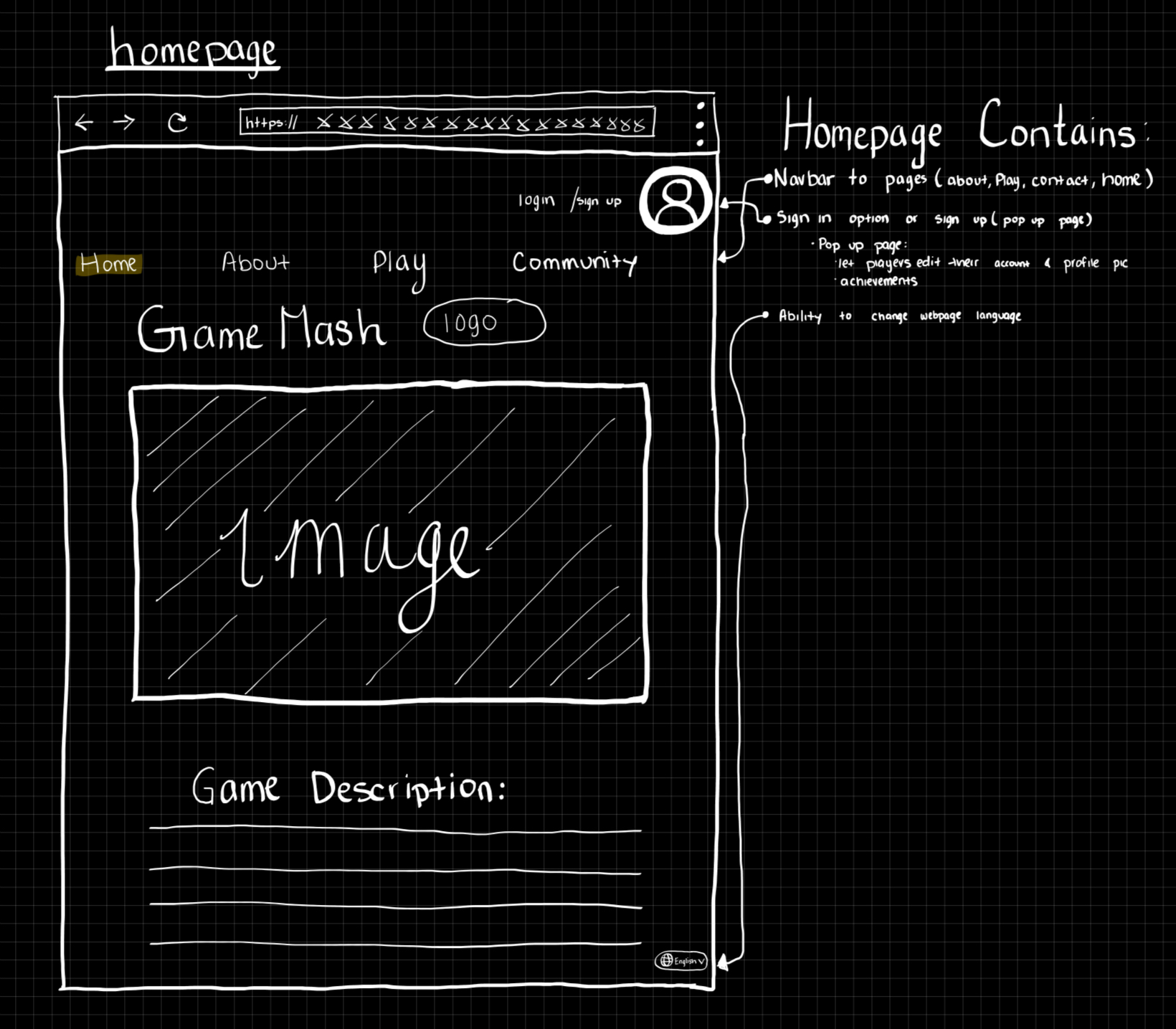
**Back End:**

* A smooth player experience.
* Puzzles that deliver a mind-blowing experience.
* Multiplayer capabilities (some way to play with others).

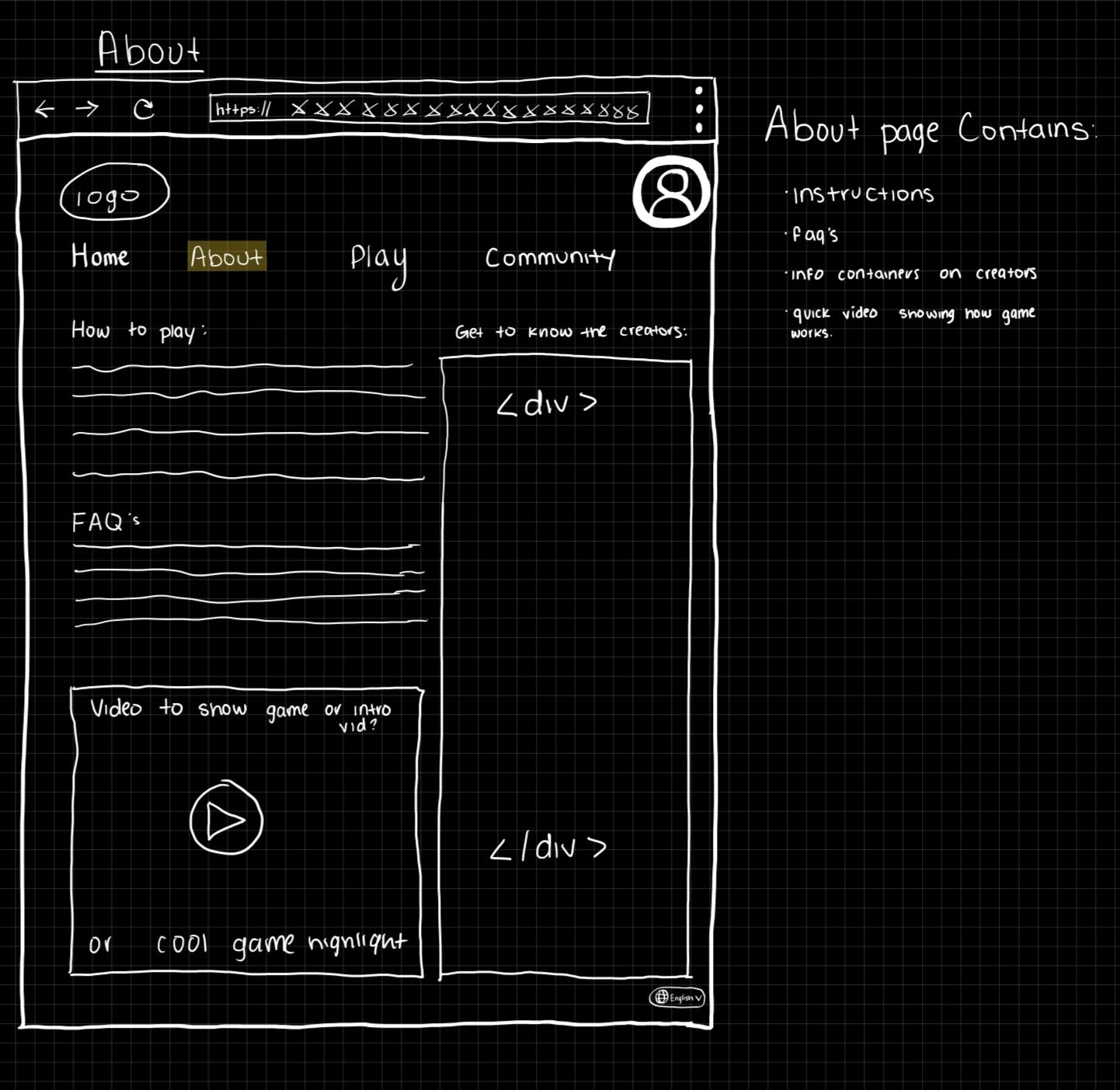
**Front-End(Perla Contributed)  
Wireframe diagrams:**

**Entire Web Design:**

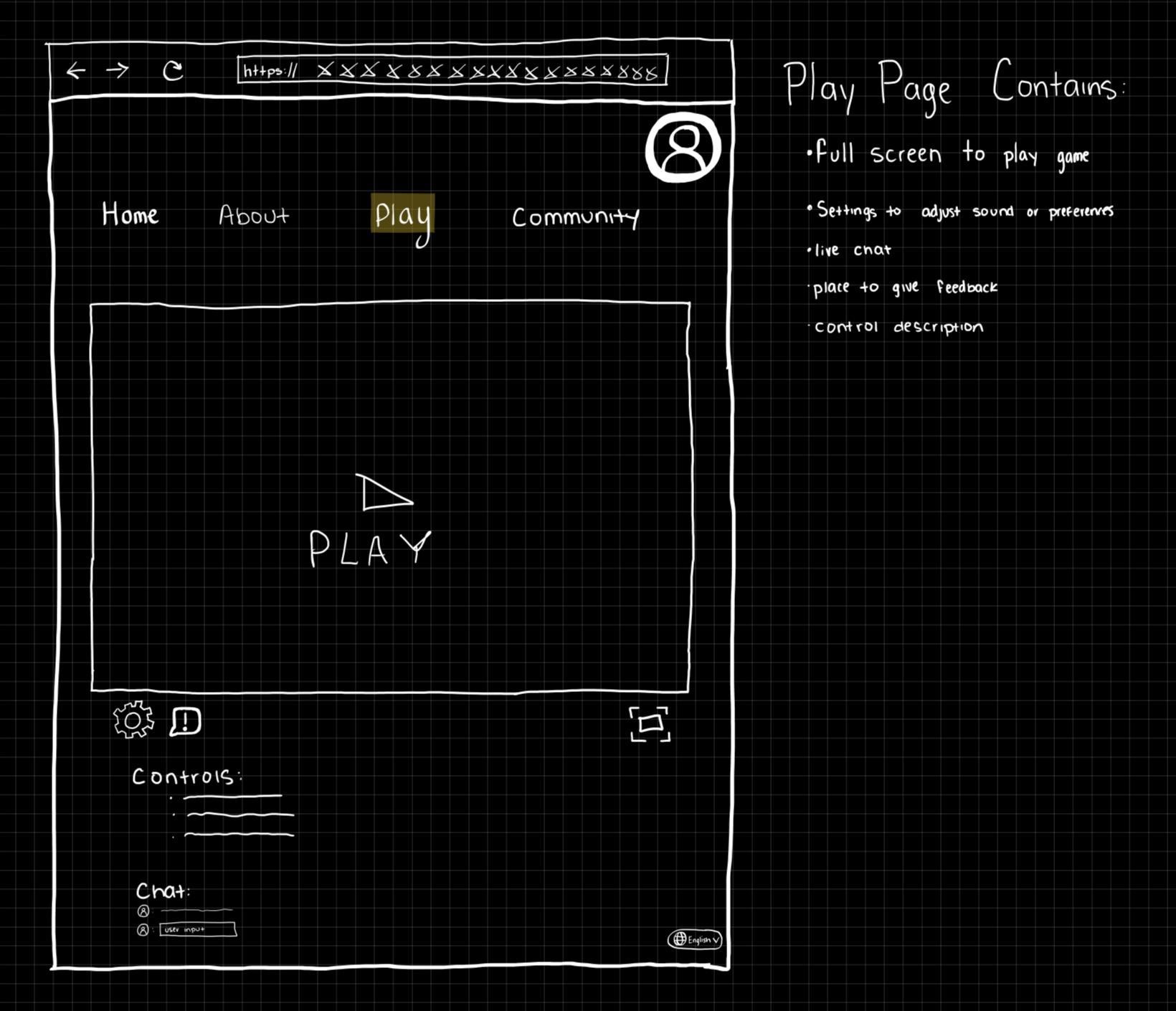
****

**Home Page:  
**

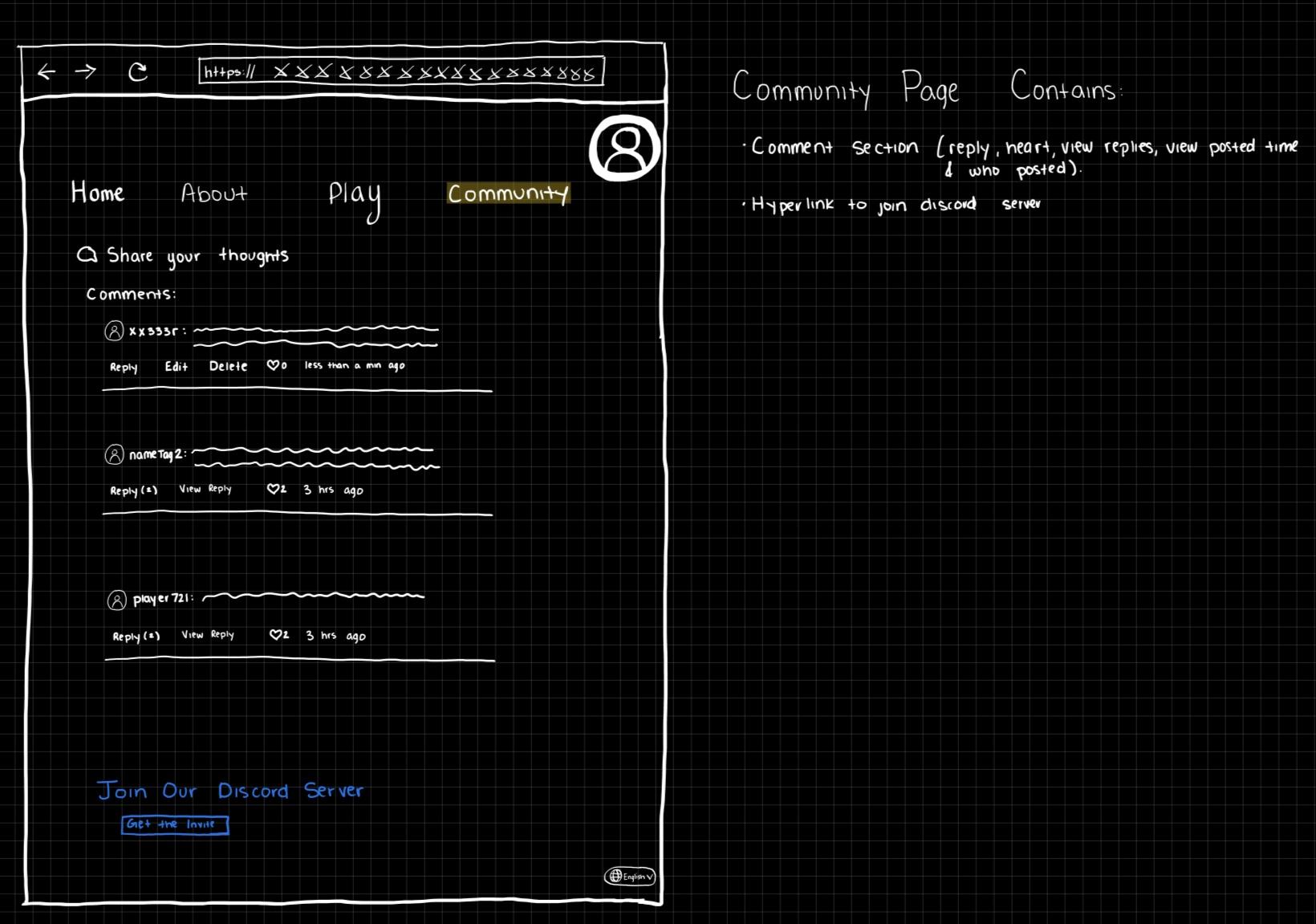
**About Page:**

****

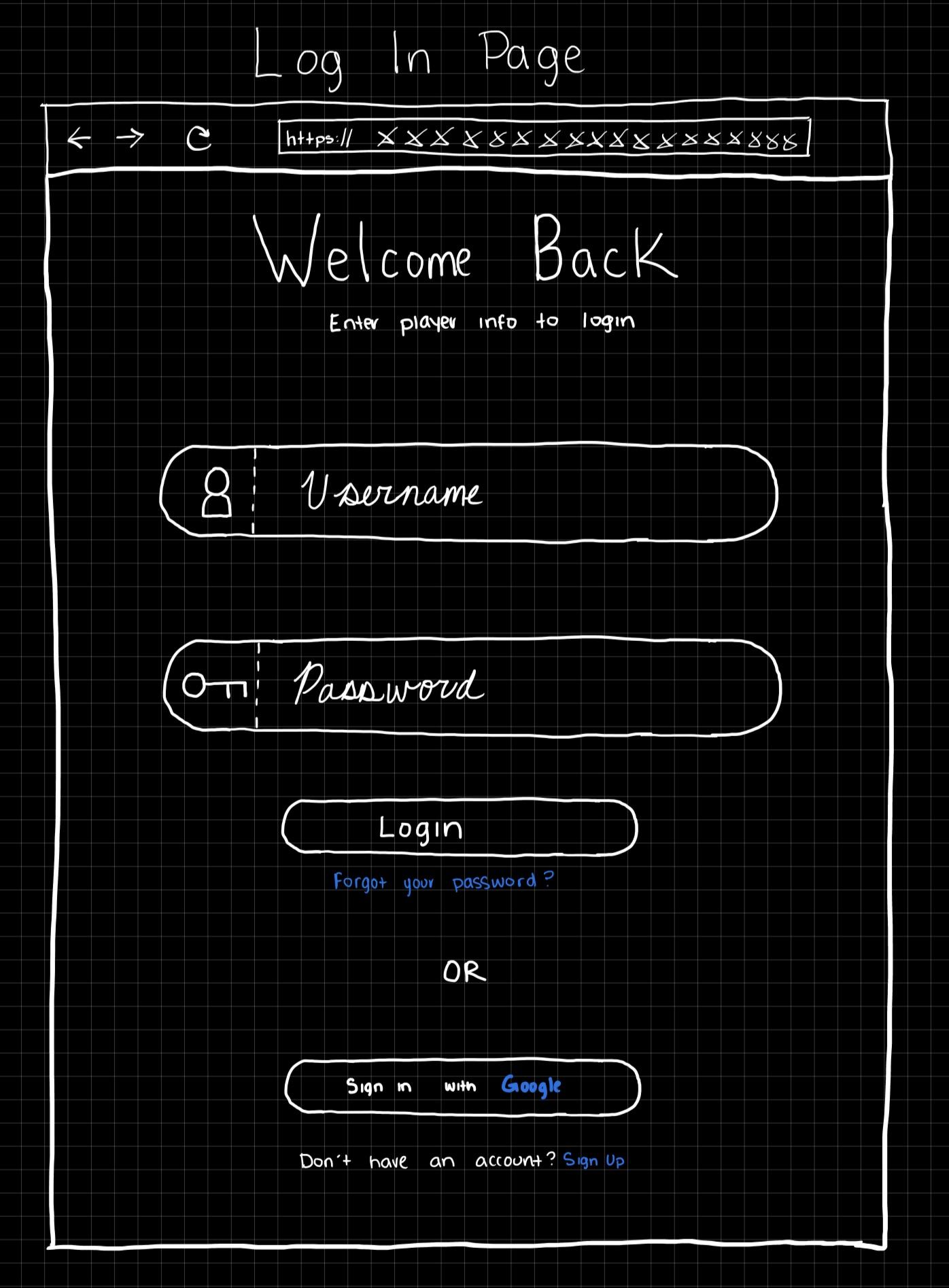
**Play Page:**

****

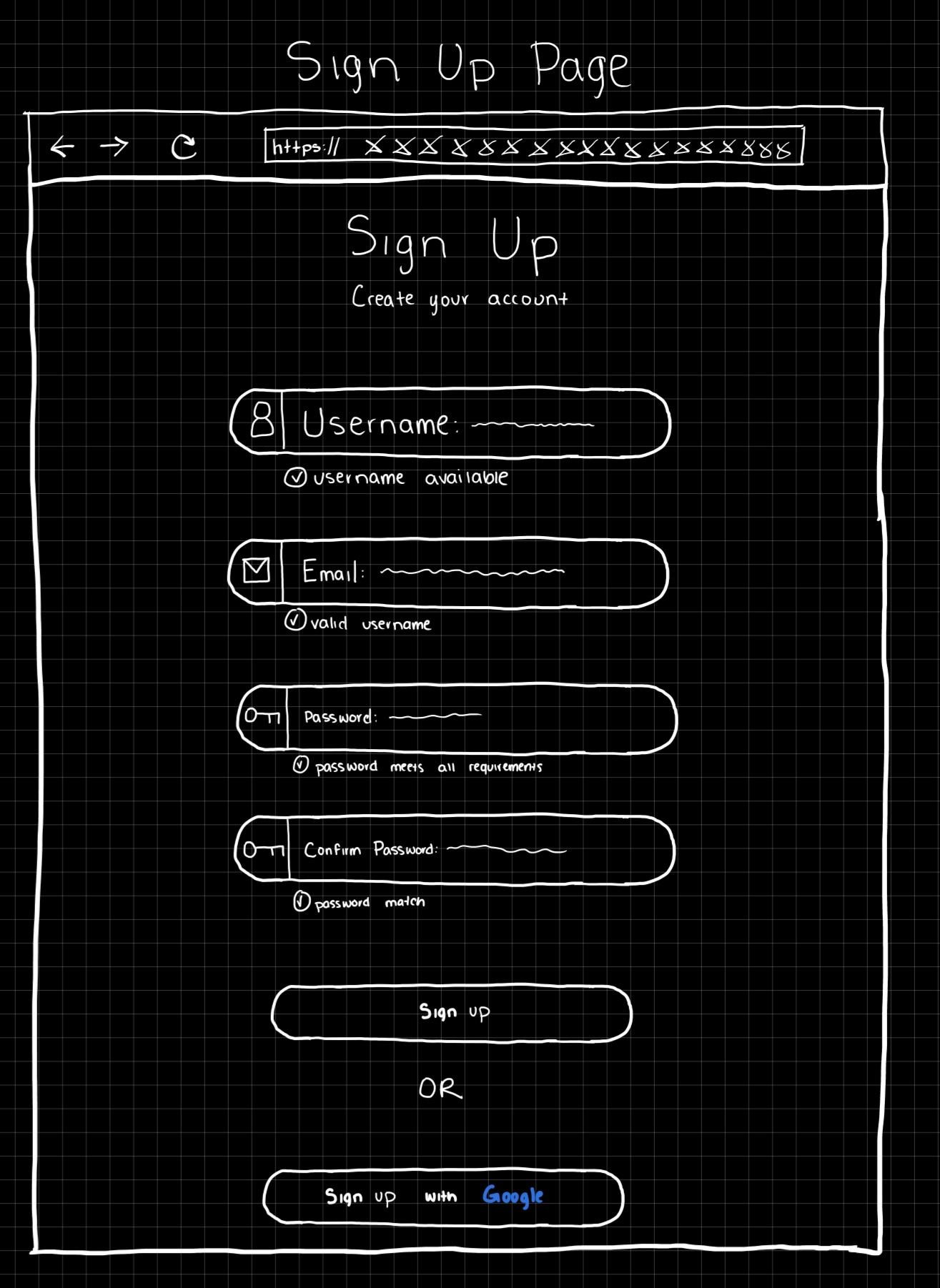
**Community Page:**

****

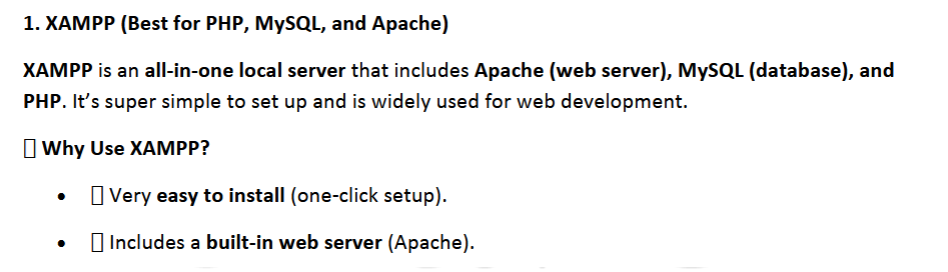
**Log in page:**

****

**Sign up page:**

****

**Back-End(Farron Contributed)  
  
 *How we will host our backend:***

We plan to use XAMPP for hosting our backend. XAMPP integrates seamlessly with development environments like Eclipse IDE and Visual Studio Code, ensuring a smooth and efficient workflow. **(Farron’s Group Comment)**  
  
**Link to steps:**[**https://sxu.instructure.com/courses/34988/files/4879377?module\_item\_id=1600564**](https://sxu.instructure.com/courses/34988/files/4879377?module_item_id=1600564) ***Pseudo-code for our Must-Haves:  
(Note: We don't need to incorporate all of the must-have features; we can select three to focus on. I’ve included the full list here for reference)*Our 3 big features/functions:**I think all of the must-have features are quick and simple to implement, except for 1. Customization for Front End, 2. Levels for Back End, and 3. Unique elements that determine interactions for Back End. (Farron’s group comment) ***Must-Have:*Front End:1. Customization**  
Function Login():

Display "Enter Username and Password"

User enters credentials

IF UserCredentials are valid:

Fetch user data from **database**

Set session to logged-in user

Display user profile

ELSE:

Display error message

Function CreateProfile():

Display "Enter Profile Information"

User enters details (Username, Avatar, Preferences)

Save profile details to **database**

Display success message and profile overview

Function TrackGameplayDuration():

Start timer when game session begins

Stop timer when game session ends

Save gameplay duration in user profile in **database**

Function RecordAchievement():

IF user reaches milestone (e.g., level completed):

Save achievement to user profile in **database**

Display achievement pop-up  
**2. Game Description(self-explanatory)**  
Function DisplayGameInstructions():

Fetch game instructions from **database** or static content

Display instructions on the screen

Display reasons to enjoy the game (e.g., unique gameplay features)

Function DisplayGameInstructionsPopup():

When user clicks "How to Play" button:

Fetch and display instructions

Include gameplay tips and hidden features

Provide a "Close" button for exiting the popup  
3. **A logo(Don’t need Pseudo-code for this, self-explanatory)**  
4. **A Community Section(We can use discord for this or whatever, self-explanatory)**

Function InitializeCommunitySection():

Display community interface with chat window and options

Initialize WebSocket for real-time communication

Function SendMessage():

User types message in chat input field

WHEN user clicks "Send":

Send message to WebSocket server

Save message to **database** (chat history)

Display message in community section

Function DisplayChatHistory():

Fetch chat history from **database** (limit by recent X messages)

Display previous messages in chat window  
**Back End:  
1. Levels**Function InitializeGame():

Display "Select Level"

Levels = ["Level 1", "Level 2", "Level 3"]

Display level options

Function StartLevel(level):

IF level == "Level 1":

Initialize Level 1 (easy difficulty, basic mechanics)

ELSE IF level == "Level 2":

Initialize Level 2 (medium difficulty, new mechanics)

ELSE IF level == "Level 3":

Initialize Level 3 (hard difficulty, complex mechanics)

Function LoadLevelData(level):

Connect to Database

Fetch level details (difficulty, objectives) from level **database**

Load level data (e.g., environment, obstacles, enemies)

Display level content

Function CompleteLevel(level):

IF level == "Level 1":

Set next level to "Level 2"

ELSE IF level == "Level 2":

Set next level to "Level 3"

ELSE:

Display "Game Over" or "You Win"

Save progress to **database** (level completed, score, etc.)

Update player profile with new level unlocked

**2. A cool menu with customizable settings(self-explanatory)**Function DisplayMainMenu():

Display "Start Game", "Settings", "Exit"

User selects "Settings"

Function ShowSettings():

Display "Sound", "Difficulty", "Display Preferences"

Display current settings for each option

Display "Save Changes" button

Function AdjustSoundSetting():

Display "Volume: Low, Medium, High"

User selects sound preference

Save new sound setting to **database**

Function AdjustDifficultySetting():

Display "Easy", "Medium", "Hard"

User selects difficulty level

Save new difficulty setting to **database**

Function AdjustDisplaySetting():

Display "Resolution: 1080p, 1440p, 4K"

User selects display preference

Save new display setting to **database**

Function SaveSettings():

Save adjusted settings to **database** (sound, difficulty, display)

Display "Settings Saved" message

**3. Unique elements that determine interactions**Function InitializeGameElements():

Display "Pick your Element" (Power-up, Item, Environment Interaction)

List available unique elements (e.g., Speed Boost, Shield, etc.)

Function CreateElement(elementType):

IF elementType == "Power-up":

Create a power-up (e.g., "Speed Boost", "Health Pack")

Display power-up on the game screen

ELSE IF elementType == "Item":

Create an item (e.g., "Key", "Weapon")

Display item on the game screen

ELSE IF elementType == "Environment Interaction":

Create an interactive environment element (e.g., "Moving Platform", "Hidden Path")

Display element in the game world

Function ElementInteraction(player, element):

IF element is "Speed Boost":

Increase player speed temporarily

Play sound effect for power-up

Save the event to **database** (element collected)

ELSE IF element is "Health Pack":

Increase player health

Play sound effect for power-up

Save the event to **database** (health increased)

ELSE IF element is "Moving Platform":

Trigger platform movement

Update player’s position on the platform

Save the event to **database** (platform interacted with)

ELSE IF element is "Hidden Path":

Reveal the hidden path

Display new gameplay area

Save event to **database** (path unlocked) **(Note: Our nice-to-haves don't require pseudo-code as they are self-explanatory.)**  
 **Database(Javier Contributed)** We will be using MySQL/SQL. SQL is the language we are using, and MySQL is the platform that enables us to interact with a database that primarily operates using the SQL language. We can also use PhpMyAdmin if we want to work with SQL, depending on what the group is most comfortable with. PhpMyAdmin also works with XAMPP. **(Farron’s Group Comment)**Link to websites:  
<https://www.phpmyadmin.net/>  
<https://www.mysql.com/>  
  
**Er Dia gram base off 1. Customization for Front End, 2. Levels for Back End, and 3. Unique elements that determine interactions for Back End:**  
