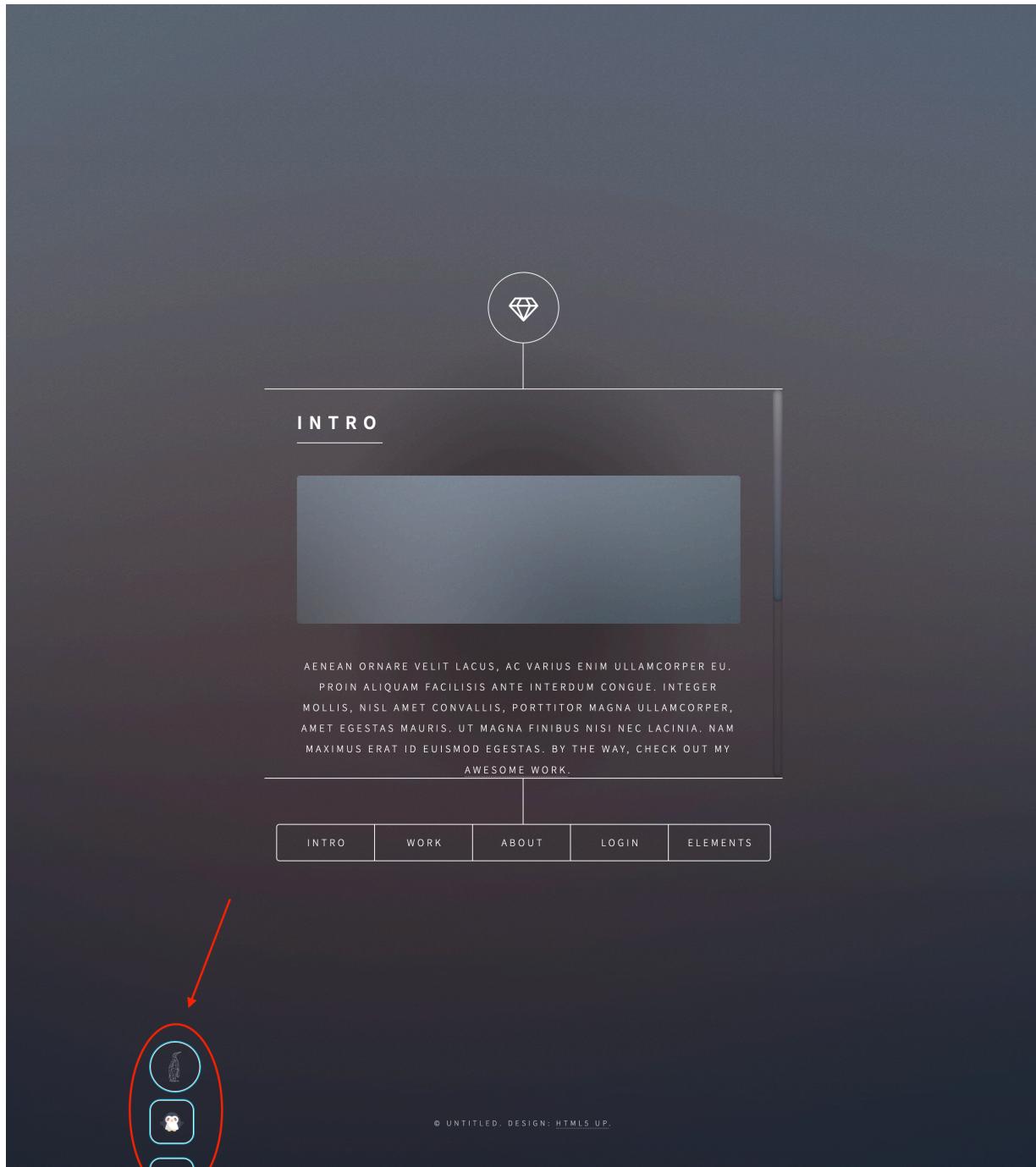


- **General :**
 - List the Requirements of the project
 - Database schemas should be created
 - Websites mapping should be created
 - List the Features
 - Function Trigger in Postgre
 - ◆ stored procedure
 - ◆ https://www.w3schools.com/sql/sql_stored_procedures.asp
 - <https://dbdiagram.io/home>
 - ◆ <https://sqlitebrowser.org/> üsttekinin masaüstü programı hali, postgre hali vardır onu bulalım
 - <https://miro.com/tr/>
 - How to handle users relation with each other on DB
 - Google -> how to map your website
 - Öncelik sıralaması : live'a çıkmak, eva'yı vermek, prod'u salmak için en temel en öncelikli şeylerin geri kalan "detaylar"dan ayrılması -> öncelikleri ve "detaylar"ı kendi listeleri içerisinde önceliğe göre sıralamak (bknz).
Yukarıda : database şeması, site mapleme, proje planlama grafiği -> öncelikli liste için Gerekli şeylerin araştırılması -> öncelikli buildi -> bug testing -> detay liste için gerekli şeylerin araştırılması -> detay buildi -> bug testing
 - Matplotlib için ; Museker , mucakmak
 - Fullstack Destek için : emirhan(front ağırlıklı), acan, bilal, saksız
 - Front için : piltan
 - SQL için : acan
 - Server için : soksak
 - Sql injection protection(Django includes this most likely but have to be sure that its enough)
 - Website name : Peng?(?), Pengpong (?), Ponguen(?), ... , ...
 - Indexing, query optimization(!), database normalization(?)
 - Containerize Microservices(?)
 - ◆ Kubernetes(?)
 - Message Brokers(?) : RabbitMQ, Apache Kafka
 - ◆ Web socket or message brokers?
 - Documentation
 - CORS - **Cross-Origin Resource Sharing(?)**
 - OAuth(?)
 - ◆ OpenID Connect(?)
- **Game**

- **What is Pong's most used AI?**
- **What is a* algo(if we gonna implement the AI ALGO module. Cause we should avoid using it.)**
- **A pause mechanic?**
- **Django**
 - **Env and requirements Automation**
 - **Django Signals**
 - Django testing tools(?)
 - Django Authentication System for user account
 - WebSockets for Real-Time Communication
 - ◆ Channels - Django extension that supports WebSockets
- **API**
 - **Fetch API**
 - **Swagger, Postman API automation**
 - **API Versioning**
 - **Rate Limiting of API's**
 - tools like Prometheus or ELK Stack for monitoring and logging to gain insights into the performance and behavior of your microservices.
 - Token Based authentication for API requests
 - What is service decomposition, API Gateways, inter-service communication
 - RESTful API design, Endpoints, HTTP methods, status codes, request/response formats(JSON)
- **Security**
 - **Two Factor Authentication(2FA)(Google?)**
 - **WAF(Do we need it ? Django looks like handles it)**
 - **Bad request Time out**
 - Common security practices

FrontEnd :

- İşaretli kısım en aşağı geldiğinde pop'upların yukarı doğru çıkması, chat baloncukları aktif olunca baloncukların chat en soldaysa sağa en sağdaysa sola doğru açılması



- Geçişlerde yazıların taşması ve gözükmesi



INTRO



AENEAN ORNARE VELIT LACUS, AC VARIUS ENIM ULLAMCORPER EU.

PROIN ALIQUAM FACILISIS ANTE INTERDUM CONGUE. INTEGER

MOLLIS, NISL AMET CONVALLIS, PORTTITOR MAGNA ULLAMCORPER,

AMET EGESTAS MAURIS. UT MAGNA FINIBUS NISI NEC LACINIA. NAM

MAXIMUS ERAT ID EUISMOD EGESTAS. BY THE WAY, CHECK OUT MY

[AWESOME WORK](#)

LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. DUIS



DAPIBUS RUTRUM FACILISIS. CLASS APTENT TACITI SOCIOSQU AD

LITORA TORQUENT PER CONUBIA NOSTRA, PER INCEPTOS HIMENAEOS.

ETIAM TRISTIQUE LIBERO EU NIBH PORTTITOR FERMENTUM. NULLAM

VENENATIS ERAT ID VEHICULA VIVERRA. NUNC UTRICES EROS UT

ULTRICIES CONDIMENTUM. MAURIS RISUS LACUS, BLANDIT SIT AMET

VENENATIS NON, BIBENDUM VITAE DOLOR. NUNC LOREM MAURIS,

FRINGILLA IN ALIQUAM AT, EUISMOD IN LECTUS. PELLentesQUE

HABITANT MORBI TRISTIQUE SENECTUS ET NETUS ET MALESUADA

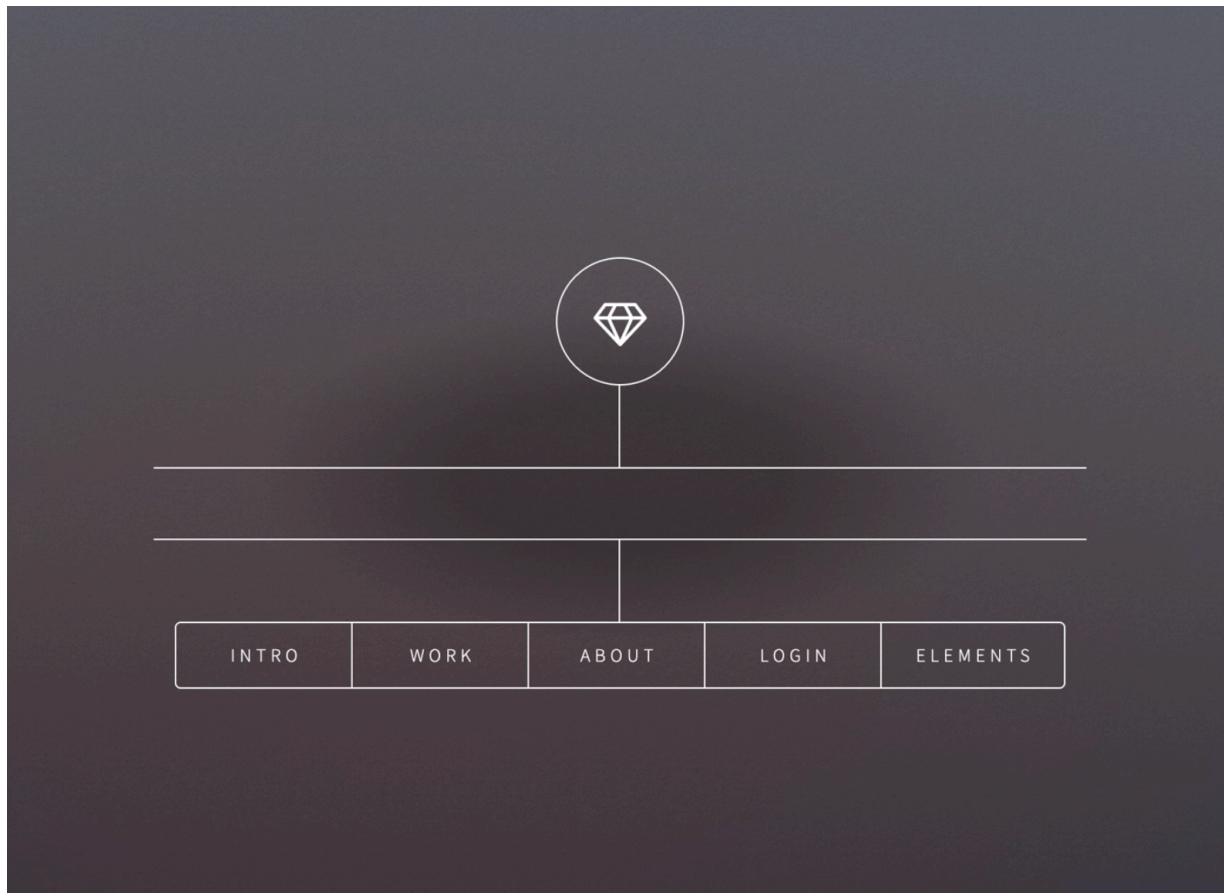
FAMES AC TURPIS EGESTAS. IN NON LOREM SIT AMET ELIT PLACERAT

MAXIMUS. PELLentesQUE ALIQUAM MAXIMUS RISUS, VEL SED

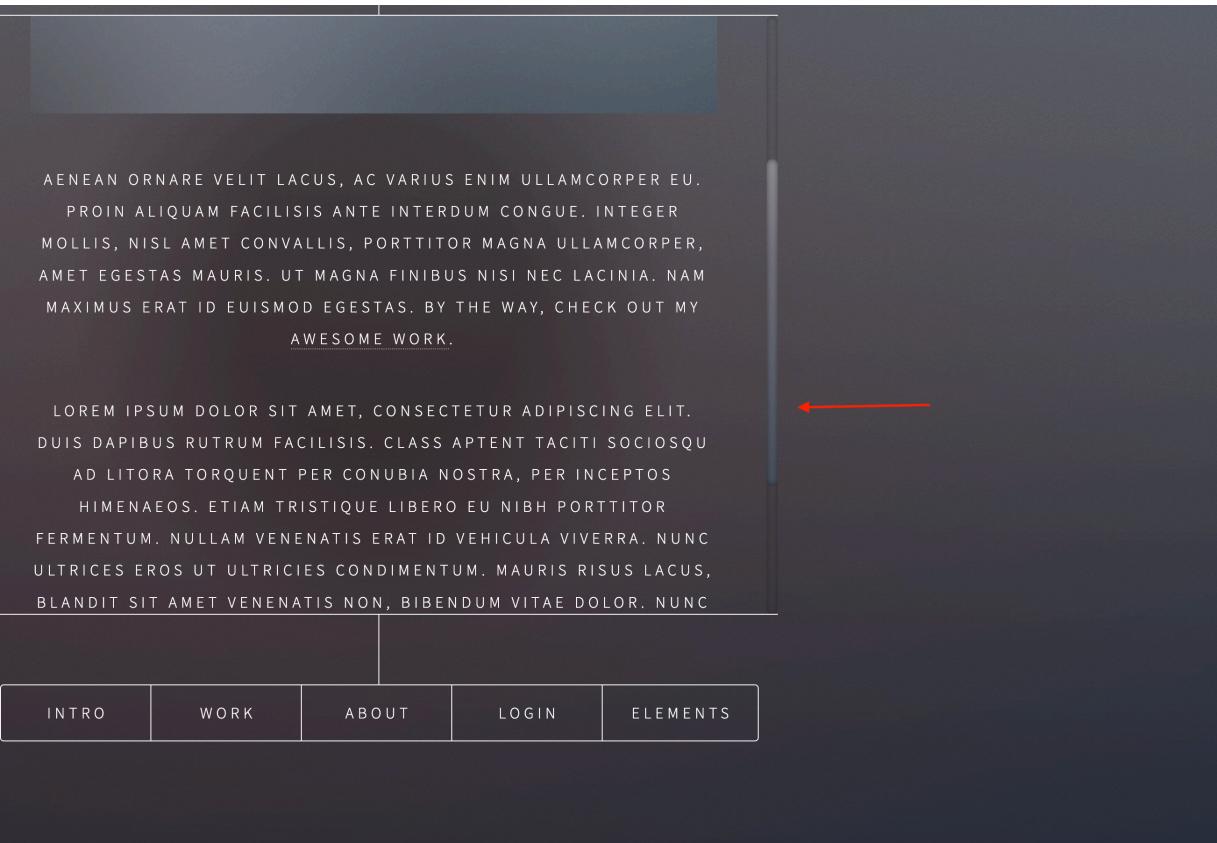
VEHICULA.

INTRO WORK ABOUT LOGIN ELEMENTS

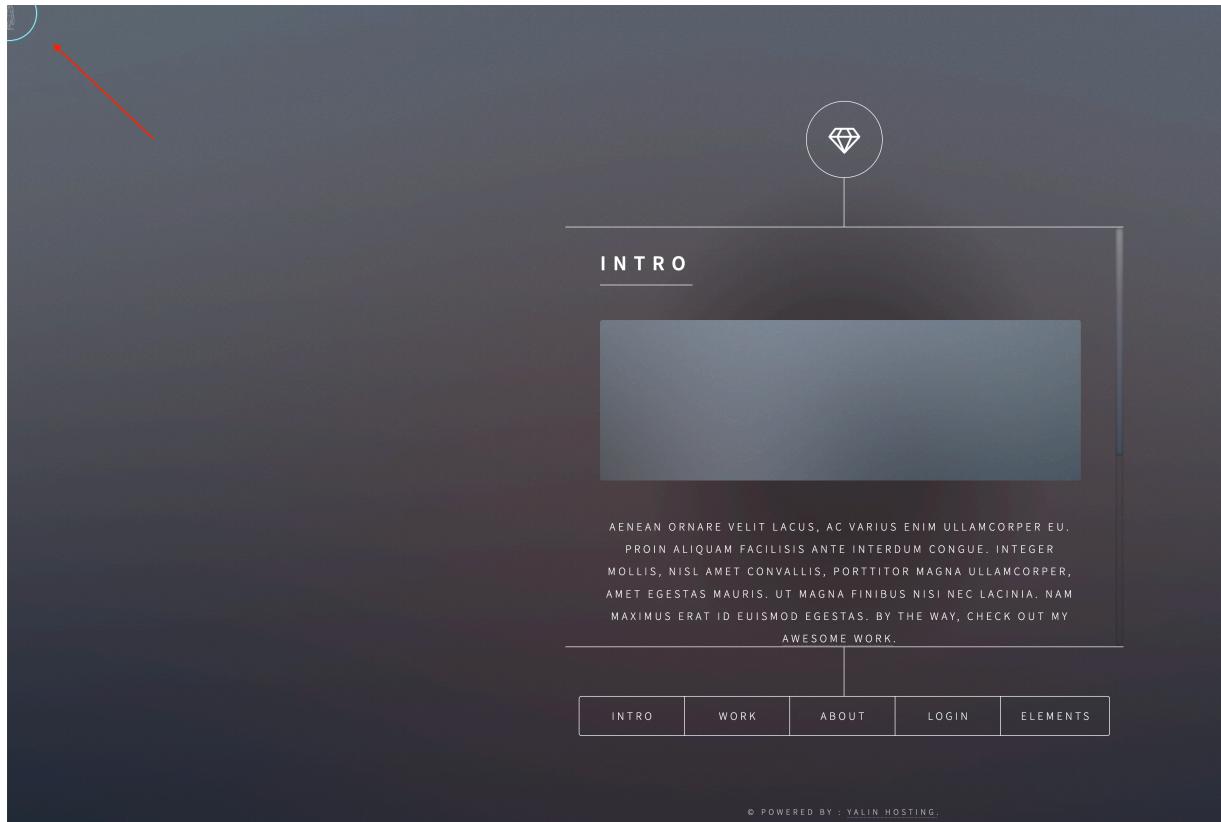
- Geçişlerde barların kapanırken takılması, ve tam kapanmıyor olması.



- Yazı geçişlerinde kare gradient yerine yazı
- Scroll bar hayalet scroll bar olsun, ya aşırı silik ya da gözükmescin, ya da küçük bir indikatör buton



- Chat balonunun responsivity için küçüldüğünde sol üstte statik bir şekilde durması ve tıklandığında bütün ekranı kaplaması(eski facebook app'inin chat gibi bir görüntü olabilir)



- Mouse üzerinden çıkış geri gelmediği sürece chat box tıklandığında açılıp kapanmıyor.
- Map
 - Intro
 - Games
 - Login/Logout
 - Profile
 - About Us
 - Ranking
- Websites Features
 - Search Bar
 - Chat Button
 - Accessibility button
 - Top button will redirect to pong game
- Accessibility Features
 - Dyslexic Font
 - Theater Mod
 - Font size up/down
 - Saturation - Contrast change
 - The contrast range should be in the standards of W3
 - Light Dark Mod
 - standards are here
 - ◆ <https://www.w3.org/TR/2008/REC-WCAG20-20081211/>

- ◆ [https://www.w3.org/WAI/WCAG22/quickref/?
versions=2.0&showtechniques=125](https://www.w3.org/WAI/WCAG22/quickref/?versions=2.0&showtechniques=125)
- ◆ <https://webaim.org/standards/wcag/WCAG2Checklist.pdf>

Details About Modules ;

● User Management

- **Standard User Management (Major Mod. +1) :**
 - ◆ Users should subscribe to the website in a secure way
 - ◆ Hash and salt the info of the users, backend doesn't have to know the direct credentials
 - ◆ Logging in in a secure way
 - ◆ Same rules applies with ^
 - ◆ Users Can select a unique display name to play the tournaments
 - ◆ Could be different for each tournament or not? If not it should be possible to change it from some settings
 - ◆ Users can update their info
 - ◆ Users can upload an avatar (Or can choose from our Peng selection)with a default option if none is provided
 - ◆ Users can add each other and see each others online status
 - ◆ Users has profiles and it displays stats, such as wins and loses
 - ◆ Each user has a Match History including 1v1 games, dates , and relevant details, accessible to logged-in users only.
 - ◆ Should manage the duplicate usernames/emails.
- **Implementing a remote authentication (Major Mod. +1):**
 - ◆ OAuth 2.0 with 42
 - ◆ What are the best practices and security standards for an Authentication Login?
 - ◆ Ensure the secure exchange of authentication tokens and user information between the web application and the authentication provider.
- **Live Chat(Major Mod. +1):**
 - ◆ Direct message to each other
 - ◆ Users should be able to block each other. When this is done, they shouldn't see each others messages no more.
 - ◆ Tournament system should ping user via messages for the next game(we might add a little sound as well, mutable ofc.)
 - ◆ Users should be able to go to other users profiles via messages.

● AI-Algo

- **AI Opponent(Major Mod. +1):**
 - ◆ A* algorithm(?)
 - ◆ Should simulate keyboard impact, ai can only refresh its view of

the game 1/sec

- User and Game Stats Dashboards(Minor Mod. +0.5):
 - ◆ Where people can see their own gaming statistics.
 - ◆ A separate dashboard for game sessions, showing detailed statistics, outcomes, and historical data for each match
 - ◆ Data visualization is needed as charts and graphs(matplotlib, pandas etc. maybe?)
- Cybersecurity
 - 2FA & JWT(Major Mod. +1):
 - ◆ With options for 2FA sms, authenticator apps or email based
- DevOps
 - Monitoring System (Minor Mod. +0.5):
 - ◆ Use Prometheus(?) and Grafana(?)
 - ◆ Prometheus is a monitoring and alerting toolkit(probably for api's)
 - ◆ It's too much to do but using just prometheus and not doing all of the module could work for us.
 - Backend as Microservices(Major Mod. +1):
 - ◆ Basically each microservice should be responsible for a single task.
- Game
 - Remote Players (Major Mod. +1):
 - ◆ Two pc's, two player, easy peasy lemon squeezy(WebSockets for Real-Time Comm., check the Channels Django Extension and [socket.io](#))
 - Multiple Players(Major Mod. +1):
 - ◆ The name is pretty self explanatory ^
 - Add Another Game \w User History & Matchmaking(Major mod. +1):
 - ◆ It's all up to Eda's decision to what to make
 - ◆ It could be Beer pong
 - ◆ The matchmaking must be fair and balanced, so it needs some kind of a "ranking" as well.
 - ◆ The data should be stored.
 - ◆ It should be responsive etc.
 - Game Customizations(Minor Mod. +0.5):
 - ◆ It should be applied to each and every game
 - ◆ if we add another game then do this it should be done for the both
 - ◆ The games should have user-friendly setting menu's or interfaces.
 - ◆ Customization features should be consistent for an unified experience across all games
- Accessibility
 - Support on All devices(Minor Mod. +0.5):
 - ◆ It should be responsive and should work on all devices.

- ◆ Touchscreen's etc. should work.
- Browser Compatibility(Minor Mod. +0.5):
 - ◆ Extend browser support to include an additional web browser.
- Multiple Languages(Minor Mod. +0.5):
 - ◆ At least 3 languages, Fr, Eng, Tr are the easy ones to do. Might add Spanish, Italian, Russian, Chinese etc. as well. It shouldn't be that hard once we handle more than 1 language.
 - ◆ Language selector
 - ◆ We can use language packs or localization libraries
 - ◆ Should allow user to set their preferred language when they log-in
- Server-Side Pong
 - Basically Pong \w API
 - It should provide enough resource and information to be played on web or CLI with the information has been given by the API.
 - CLI olarak çalışması için ne gerekli