

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Кафедра «Обчислювальна техніка та програмування»

ПОЯСНЮВАЛЬНА ЗАПИСКА
до курсового проекту

з навчальної дисципліни
«Веб-програмування»

Виконав студент 3 курсу, групи КІТ-118В
Кліщов Богдан Романович
Керівник: Лимаренко Вячеслав Володимирович

Харків 2020

АНОТАЦІЯ

В даному документі описується процес створення веб-сайту, використовуючи HTML, CSS та JavaScript. В даному проєкті будуть використовуватися допоміжні бібліотеки, такі як jQuery та стилі cdnjs.

АННОТАЦИЯ

В данном документе описывается процесс создания веб-сайта, используя HTML, CSS и JavaScript. В данном проекте будут использоваться вспомогательные библиотеки, такие как jQuery и стили cdnjs.

ANOTATION

This document describes the process of creating a website using HTML, CSS and JavaScript. This project will use auxiliary libraries such as jQuery and cdnjs styles.

ЗМІСТ

ЗАГАЛЬНІ ПОНЯТТЯ ПРО ВЕБ-ПРОГРАМУВАННЯ.....	4
ВЕРСТКА САЙТУ.....	6
МОВА JAVASCRIPT ТА ЇЇ МОЖЛИВОСТІ.....	8
ВИКОРИСТАННЯ QUERY В ВЕБ-ПРОГРАМУВАННІ.....	9
РОЗРОБКА ВЛАСНОГО ВЕБ-САЙТУ.....	11
РЕЗУЛЬТАТИ.....	17
ВИСНОВКИ.....	19
СПИСОК ЛІТЕРАТУРИ.....	20

ЗАГАЛЬНІ ПОНЯТТЯ ПРО ВЕБ-ПРОГРАМУВАННЯ

Стрімкий розвиток інтернету за останні пару років призвів до збільшення кількості нових онлайн-сервісів. Починаючи від первісних характеристик і основних сервісів, доступних користувачам (пошта і веб), інтернет розвинувся у велику всесвітню павутину з великою кількістю різних сервісів, технологічних рішень і став тим місцем, де зустрічаються практично бізнес-організації всього світу. Спостерігається зростання кількості компаній, які переходять на електронний бізнес. Вони відкривають свої корпоративні домени, створюють свої веб-сайти, рекламують свої товари і послуги в інтернеті, запускають продажі онлайн.

Основна роль в технічній частині цієї роботи буде належати саме веб-програмістам і розробникам. Їх завдання:

- створювати веб-сайти
- html верстка
- з'єднувати їх з відповідною базою даних
- писати веб-сервлети
- створювати веб-сервіси
- встановлювати з'єднання в інтернеті
- створювати веб-додатки для різних цілей
- створювати технічні передумови для продажів онлайн і просування товарів і послуг
- дбати про безпеку веб-сайтів і веб-серверів
- оптимізувати існуючі веб-додатки клієнтів
- робити можливою автоматизацію публікації контенту

Веб-програмування — це створення сайтів і програм, які працюють у мережі. Створюються спеціальні комп'ютерні програми — скрипти, які діляться на два типи: серверні і клієнтські. Сервер являє собою комп'ютер, на якому знаходяться файли сайту. Клієнт — це сам користувач, а якщо бути

точніше, то браузер, встановлений на ПК. Подаючи запит до сервера, він виступає в ролі клієнта.

Сполучною ланкою між веб-розробником і сайтами є мова веб-програмування. Існує багато мов на яких пишуться і серверні, і клієнтські скрипти. Але деякі мови, навпаки, використовуються виключно для серверних чи клієнтських скриптів.

ВЕРСТКА САЙТУ

Для людей, зайнятих в ІТ сфері, поняття верстки здається очевидним. Тим більше складно пояснити людині, що з сайтами стикається тільки в якості користувача.

Коли створений дизайн сайту, коли готові всі функціональні модулі й компоненти, необхідно з'єднати всі елементи в єдине ціле, тобто зверстати сайт. Верстка сайту – це кінцевий етап розробки сайту, створення структури сайту, що буде визначати відображення тексту й графіки на сайті в різних браузерах.

Розрізняють «жорстку» (фіксовану) і «гумову» (змінювану) верстку сайту. При «жорсткій» верстці всі елементи web-сторінки мають завжди фіксовані розміри, незалежно від розміру монітора користувача й установленого дозволу екрана. «Гумова» верстка сайту дозволяє змінювати розміри елементів сторінки, підлаштовуючись під різні розміри й дозволи моніторів.

Верстка сайту на сьогоднішній день може бути виконана безліччю різних способів. Розглянемо найпоширеніші з них.

HTML-верстка сайту – це верстка сайту на основі мови гіпертекстової розмітки HTML. Залежно від основних елементів html, що використовуються при верстці сайту, виділяють табличну html-верстку й блокову html-верстку.

Таблична верстка сайту – це верстка, при якій структура сторінки сайту представлена у вигляді таблиці. Кожний елемент сторінки – це одна або кілька комірок таблиці. Таблична верстка сайту зручна й широко застосовується верстальниками, однак вона не завжди може задовольнити потреби сайту зі швидкості завантаження, наприклад.

Блокова верстка сайту, або як її ще називають верстка div'ами, – це верстка сайту на основі елементів <div>. Така верстка сайту має ряд переваг: стислість коду, висока швидкість завантаження коду й т.д.

CSS-верстку іноді виділяють в окремий вид верстки сайту, хоча в сучасному веб-дизайні каскадні таблиці стилів (css) використовуються практично при будь-якій верстці. Каскадні таблиці стилів описують зовнішній вигляд сторінок сайту, написаних мовою розмітки: HTML, XHTML, XML.

Верстка з використанням css може використовувати зв'язані стилі, глобальні стилі, внутрішні стилі. Зв'язані стилі (таблиці зв'язаних стилів) дозволяють «відокремити» код від зовнішнього оформлення: всі параметри форматування тегів розміщаються в окремому файлі css. Достатньо лише підключити цей файл у потрібному місці web-сторінки. Один файл css може бути використаний для безлічі різних web-сторінок.

Глобальні стилі описуються в тегу `<style>`, що розташовується в заголовку (`<head>`) web-сторінки. Описані стилі будуть застосовані тільки до однієї сторінки. Внутрішні стилі, у свою чергу, призначаються для кожного конкретного тегу й застосовуються тільки стосовно нього.

У сучасному веб-дизайні всі частіше використовується комбінована верстка сайту – це дозволяє набагато збільшити можливості верстки, створити більш складний і привабливий сайт.

Верстка повинна відповідати певним стандартам, щоб сайт коректно відображався в будь-якому браузері, що підтримує ці стандарти. Здатність сайту коректно «працювати» незалежно від браузера, називається кросбраузерністю. Кросбраузерна верстка сайту – важлива вимога до якості сучасних сайтів.

МОВА JAVASCRIPT ТА ЇЇ МОЖЛИВОСТІ

JavaScript – це мова програмування, що дозволяє зробити Web -сторінку інтерактивною, тобто такою що реагує на дії користувача.

Послідовність інструкцій (що називається програмою, скриптом або сценарієм) виконується інтерпретатором, вбудованим в звичайний Web -браузер. Іншими словами, код програми вбудовується в HTML – документ і виконується на боці клієнта. Для виконання програми не потрібно навіть перезавантажувати Web -сторінку, всі програми виконуються в відповідь на будь-яку подію. Наприклад, перед відправленням даних форми можна перевірити їх на допустимі значення і, якщо значення не відповідають очікуванім, заборонити відправлення даних.

JavaScript – об'єктно-орієнтована скриптова мова програмування і є діалектом мови ECMAScript. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить у браузерах як мова сценаріїв для надання інтерактивності веб-сторінкам.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання непрограмістами. JavaScript має низку властивостей об'єктно-орієнтованої мови, але реалізоване в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними об'єктно-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, - функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання – що додає мові додаткову гнучкість.

ВИКОРИСТАННЯ QQUERY В ВЕБ-ПРОГРАМУВАННІ

Творцем фреймворка jQuery є відомий програміст Джон Резіг, який з дитинства захоплювався кодингом. Спочатку він вивчав QBasic, Visual Basic і HTML. У 2004 році він пристрасився до JavaScript і згодом створив на цій мові цілий ряд додатків. Розробнику не подобалося, що для роботи доводиться використовувати фрагменти коду інших девелоперів. При створенні jQuery, Резіг побачив у цьому можливість впровадити в нього модулі, які будуть часто використовуватися згодом. Це б допомогло спростити JavaScript у плані кросбраузерних рішень.

Готову бібліотеку рішень презентували на нішевій події BarCamp, яка пройшла у місті Нью-Йорк в 2006 році.

Бібліотека включає наступний функціонал:

- CSS-селектори Sizzle, які вже встигли оформитися в окремий проект;
- Робота з DOM-елементами;
- Події;
- Візуальні ефекти;
- Доповнення на AJAX;
- Різноманітні JavaScript-розширення.

Query намагається відокремити поведінку від загальної структури HTML-документа. Наприклад, якщо мова йде про роботу обробника подій натискання кнопки, то замість безпосереднього його зазначення, управління переходить до Джейквері. Бібліотека спочатку визначає кнопку, після чого трансформує сигнал в обробник подій кліку.

Відділення поведінки і структурних компонентів дозволяє зробити Javascript більш гнучким.

Серед модулів бібліотеки jQuery є багато вигідних компонентів, які можна застосувати в широкому спектрі завдань розробника. Разом з тим, команда розробників бібліотеки не намагалася створити набір інструментів, які дозволяли б поєднувати велику кількість jQuery-функцій, оскільки це може призвести до створення великих фрагментів коду. Саме тому вибрали

архітектуру універсального ядра, яка містить бібліотеку і плагіни. Це дає можливість об'єднати тільки ті JavaScript-функції, які по-справжньому потрібні.

Основні переваги:

- **Кросбраузерність** – модулі бібліотеки однаково добре працюють у будь-якому браузері;
- **Швидкість введення коду** – якщо в Javascript найпростіші команди використовують кілька рядків коду, то в JQuery виходить заощадити ;
- **Зрозумілість** та зручність читання команд
- **Доступно підключення різних плагінів** та можливість завдання їх стилів;
- Дуже спрощує використання Ajax;

А найголовніше JQuery вдало і просто поєднує між собою Javascript, HTML і файли CSS-стилів.

РОЗРОБКА ВЛАСНОГО ВЕБ-САЙТУ

Для розробки власного сайту буде використовуватися текстовий редактор Visual Studio Code. Для комфортної роботи завантажені допоміжні плагіни (рис. 1)

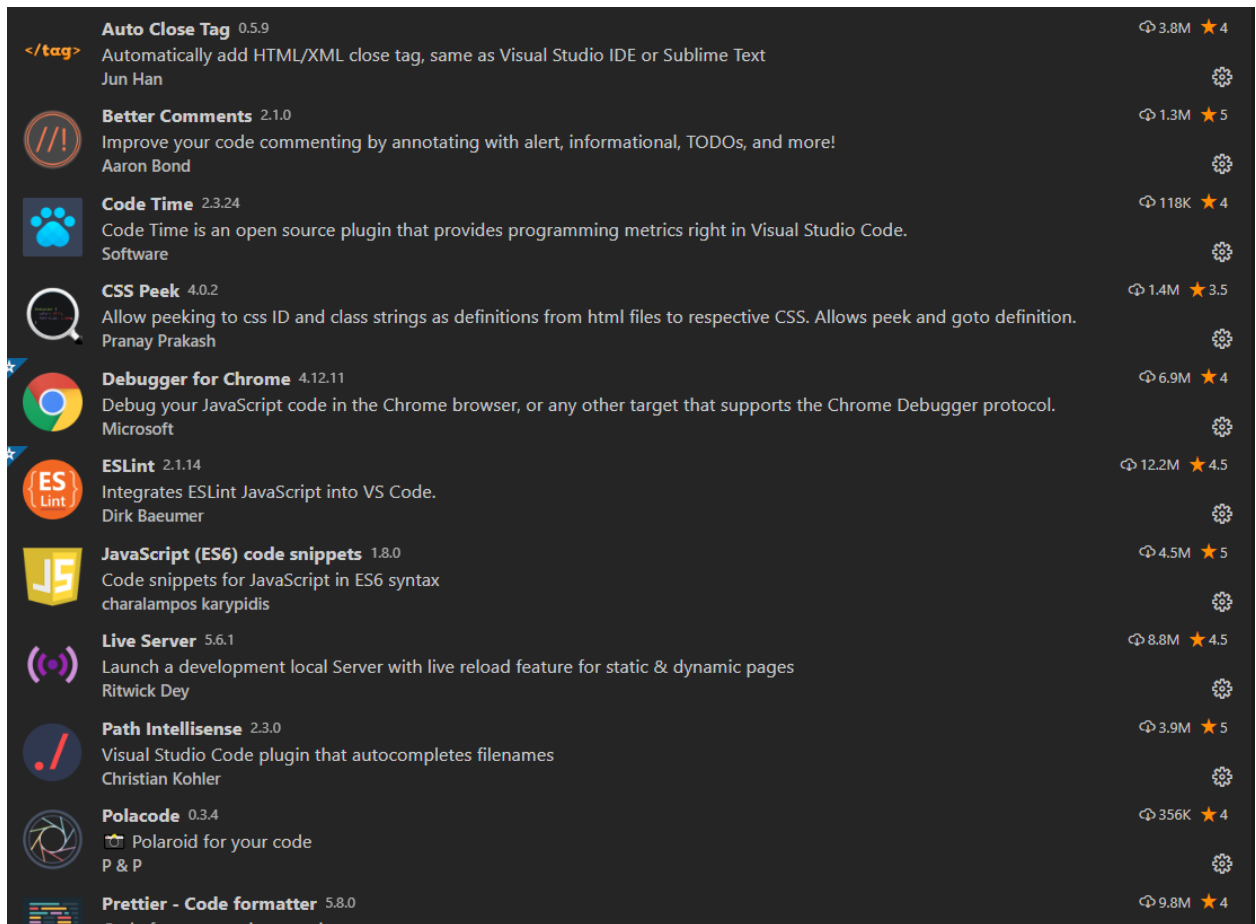


Рисунок 1 – Плагіни для Visual Studio Code

Темою сайту вибрана гра Cyberpunk 2077. **Основною задачею** є розробка простого, зручного та привабливого сайту, який буде гарно працювати як на комп'ютерах, так і на мобільних пристроях.

Сайт буде складатися з декількох частин: анімація завантаження, хедер, меню, особливості гри, основні NPC та футер.

В <head> підключені додаткові бібліотеки та кастомні шрифти від Google (рис. 2)

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>CyberPunk</title>
  <link rel="stylesheet" href="style.css" />
  <link
    rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/css/all.min.css"
  />
  <link
    href="https://fonts.googleapis.com/css2?family=Baloo+Da+2:wght@400;500;600;700;800&family=Josefin
    rel="stylesheet"
  />
</head>
```

Рисунок 2 – Вміст тегу <head>

Створимо блок, який буде відповідати за анімацію завантаження веб-сторінки (рис. 3)

```
<div class="spinner-container">
  <div class="circles">
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
  </div>
</div>
```

Рисунок 3 – Створення блоку завантаження

Далі додаємо блоки виклику меню, а також хедер (рис. 4)

```
<div class="hamburger-menu">
  <div class="line line-1"></div>
  <div class="line line-2"></div>
  <div class="line line-3"></div>
  <span>Close</span>
</div>
<header class="header">
  <div class="img-wrapper">
    
  </div>
  <div class="banner">
    
  </div>
</header>
```

Рисунок 4 – Виклик меню та хедер

Додамо функціоналу в меню (рис. 5)

```
<section class="sidebar">
  <ul class="menu">
    <li class="menu-item">
      <a href="#" class="menu-link" data-content="Home">Home</a>
    </li>
    <li class="menu-item">
      <a href="https://cyberpunk.fandom.com/wiki/Main_Page" class="menu-link" data-content="Features">Features</a>
    </li>
    <li class="menu-item">
      <a href="https://cyberpunk.fandom.com/wiki/Category:Cyberpunk_2077_Characters" class="menu-link" data-content="Characters">Char
    </li>
    <li class="menu-item">
      <a href="https://cyberpunk.fandom.com/wiki/Category:Cyberpunk_2077_Corporations" class="menu-link" data-content="Corporations">
    </li>
    <li class="menu-item">
      <a href="https://cyberpunk.fandom.com/wiki/Category:Cyberpunk_2077_Locations" class="menu-link" data-content="Locations">Locati
    </li>
  </ul>
  <div class="social-media">
    <a href="https://www.facebook.com/getfandom"><i class="fab fa-facebook-f"></i></a>
    <a href="https://www.instagram.com/getfandom/"><i class="fab fa-instagram"></i></a>
    <a href="https://twitter.com/getfandom"><i class="fab fa-twitter"></i></a>
  </div>
</section>
```

Рисунок 5 – Функціонал меню

Створимо першу сторінку, на якій буде відображатися інформація про гру, її переваги, а також значки, які були підключені в теги <head>, а також додамо зображення (рис. 6)

```
<div class="service">
  <div class="service-header">
    <i class="fa fa-gamepad"></i>
    <h3>Ultimate gameplay</h3>
  </div>
  <p class="service-text">
    In the definition process, you can choose backstory
    elements that will influence the ultimate gameplay.
  </p>
</div>
```

```
<div class="features-img-wrapper">
  
</div>
```

Рисунок 6 – Створення першої сторінки

Таким же чином добавляємо ще шість параметрів.

Створимо другу сторінку сайту, на якій буде відображатися список головних NPC з короткою інформацією про них та анімованим блоком зображення, який реагує на положення курсору (рис. 7)

```
<div class="card" data-tilt>
  <div class="card-img-wrapper">
    
  </div>
  <div class="card-info">
    <h2>Judy Alvarez</h2>
    <h3>Techie</h3>
    <p>
      Judy Alvarez is a skilled braindance technician and
      member of the Moxes in Cyberpunk 2077.
    </p>
    <button><a href="https://cyberpunk.fandom.com/wiki/Judy_Alvarez">Read More</a></button>
  </div>
</div>
```

Рисунок 7 – Створення інформації про NPC

Весь функціонал анімації буде описаний в .js файлах.

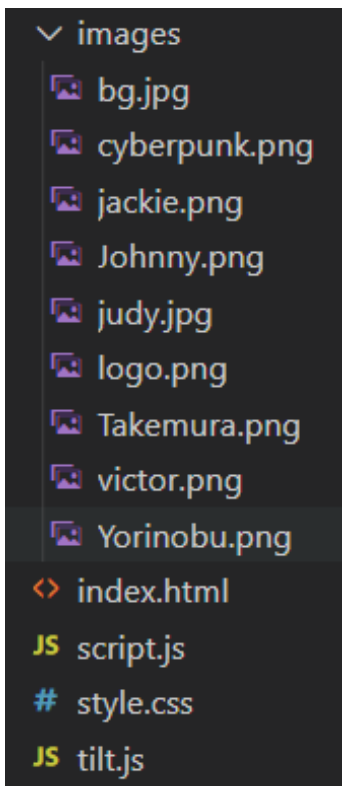
Додамо футер в проект з посиланнями на соціальні мережі (рис. 8)

```
<footer class="site-footer">
  <div class="container">
    <div class="row">
      <div class="col-md-8 col-sm-6 col-xs-12">
        <p class="copyright-text">Copyright &copy; 2020 All rights reserved by Santa Claus.</p>
      </div>

      <div class="col-md-4 col-sm-6 col-xs-12">
        <ul class="social-icons">
          <li><a class="facebook" href="https://www.facebook.com/getfandom"><i class="fab fa-facebook"></i></a></li>
          <li><a class="twitter" href="https://twitter.com/getfandom"><i class="fab fa-twitter"></i></a></li>
          <li><a class="youtube" href="https://www.youtube.com/fandomentertainment"><i class="fab fa-youtube"></i></a></li>
          <li><a class="linkedin" href="https://www.linkedin.com/company/157252"><i class="fab fa-linkedin"></i></a></li>
        </ul>
      </div>
    </div>
  </div>
</footer>
```

Рисунок 8 – Футер

В результаті отримано таку структуру сайту:



В файлі tilt.js описано поведінку блока з зображеннями другої сторінки на зміну положення курсору.

Файл style.css містить в собі стилі для всіх тегів, а також інструкції зміни положення елементів при зміні ширини екрану.

В файлі script.js знаходиться текст програми, яка описує поведінку елементів при анімації, для додання динамічності веб-сайту.

На рисунку 9 знаходиться код файлу script.js

```
window.onload = () => {  
  setTimeout(() => {  
    document.querySelector("body").classList.add("display");  
  }, 2000);  
};  
  
document.querySelector(".hamburger-menu").addEventListener("click", () => {  
  document.querySelector(".container").classList.toggle("change");  
});  
  
document.querySelector(".scroll-btn").addEventListener("click", () => {  
  document.querySelector("html").style.scrollBehavior = "smooth";  
  setTimeout(() => {  
    document.querySelector("html").style.scrollBehavior = "unset";  
  }, 1000);  
});
```

Рисунок 9 – Код файлу script.js

Проект збережений на GitHub за наступним посиланням:
<https://github.com/klishchov-bohdan/klishchov-bohdan.github.io>

Веб-сайт розміщено на сервісі GitHub Pages, продивитися його можна, перейшовши за наступним посиланням посиланням:

<https://klishchov-bohdan.github.io/>

РЕЗУЛЬТАТИ



Ultimate gameplay

In the definition process, you can choose backstory elements that will influence the ultimate gameplay.

Character

You can customize your character's abilities, and the process for customizing appearance is extremely detailed, including body type, skin tone, tattoos, makeup, and more.

Aspects of your character

The aspects of your character's appearance will influence the way that other people interact with you.



CYBERPUNK
2077

Body modification

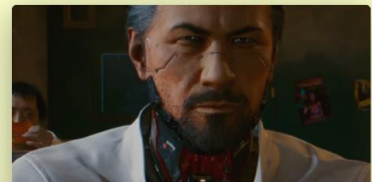
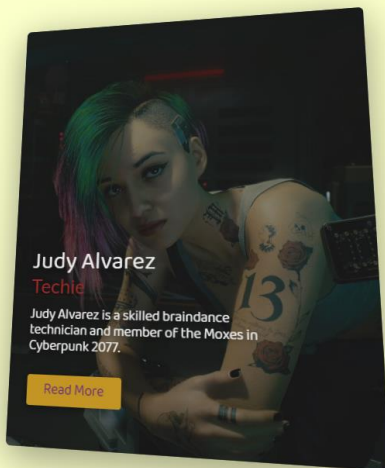
In Cyberpunk 2077, getting body modifications seems to be more than just picking upgrades from a list.

Backpack

The backpack provides an overview of all the different items currently in your inventory.

Minimap

The minimap could use more zoom levels so that you can see upcoming turns better while driving.



Copyright © 2020 All rights reserved by Santa Claus.



ВИСНОВКИ

Розроблено власний сайт на тему CyberPunk, з використанням HTML, CSS та JavaScript, а також допоміжних бібліотек. Сайт є адаптивним, тобто змінюється, залежно від ширини екрана, для зручності використання на будь-яких пристроях. Отримав досвід додавання анімації на сайт, а також досвід роботи з jQuery.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Дженнифер Нидерст Роббинс "HTML5, CSS3 и JavaScript. Исчерпывающее руководство". 4-ое издание (2014)
2. Дэвид Макфарланд "Большая книга CSS3" (2014)
3. Брайан Хоган "HTML5 и CSS3. Веб-разработка по стандартам нового поколения" (2011)
4. Терри Фельке-Моррис «Большая книга веб-дизайна» (2017)
5. https://cyberpunk.fandom.com/wiki/Main_Page