Нефункціональні вимоги до програми

1)– операційні вимоги:

Система – персональний комп’ютер. Так як це буде консоль, вона буде працювати зі всіма платформами.

Повинні бути встановлені складові частини для роботи з Python інтерпретатором, засіб для роботи з запуском програми (може використовуватися і звичайний командний рядок, зокрема) та бібліотека-директорія з встановленими засобами праці над програмою (Їх я отримую з GitHub репозиторію).

2)-Практичні вимоги

Час виконання програми не повинен займати більше 2 хвилин;

Система завантажує список всієї потрібної користувачу інформації;

Програма має бути доступною круглий рік;

Програма підтримує користування кожним юзером особисто;

Програма повинна бути адаптованою під будь-який ввід користувача, навіть помилковий;

Система повинна бути зрозумілою користувачеві, у ній має буди хоча б 2 мови інтерфейсу користувача(Українська та Англійська). Інтерфейс програми має мати чіткі кроки для виконання і для оновлення.

3)-Безпека

Так як інформацію я буду брати з сайту Укрзалізниці, а даний АРІ вже використовується телеграм-ботом, з загальновідомим token-ом, то цією програмою зможе користуватися кожен без авторизації.

Споживачі отримують інформацію як окрему сторінку. Історія не зберігається.

Система не є захищеною від вірусів, але вона не використовуватиметься в промислових цілях, тому реальної цілі для «забруднення» вірусами немає.

4)-Ліцензії

Формально, я використовую ліцензію для доступу до інформації, тому все дозволено. Сам АРІ використовує інформацію з booking.uz.gov.ua. Ключ – token є в html – описі сторінки, тому проблем з доступом до інформації немає. Моєю основною думкою є просто можливість провести дослідження, яке можна буде виконувати будь- коли, а дослідження не потребує якихось окремих дозволів.

Загалом, мій проект не має багато вимог, які потрібні для нормального функціонування. Звичайний комп’ютер, який може завантажити якусь інформацію з сайту у вигляді json-файлу.

В АРІ використовуються лише POST запити. Для захисту від використовування АРІ майже в усіх викликах в тіло цього виклику включається token-ключ. Без цього ключа можна лише робити пошук станцій.

Варто також відзначити деякі нюанси роботи з датами. По-перше, формат дати змінюється в залежності від місцезнаходження юзера. Наприклад, для EN формат дати буде: місяць/день/рік, тоді як для UA, RU – це буде формат день/місяць/рік. По-друге, для деяких запитів дата представляється як timesamp, проте він залежить від стану літнього/зимового часу.

Функціональні вимоги

**1)**Що має виконуватися

*Обробка інформації щодо вашого місцезнаходження*

Користувач буде вводити своє розташування або місце, з якого йому б хотілося почати мандрівку. Ця інформація буде використана у формуванні маршруту доїзду.

*Виведення всіх станцій*

Виведення основної їнформації користувачу про кожен окревий маршрут

*Отримання від користувача дати і години початку подорожі*.

Початок буде залежати від розкладу руху потягів. Якщо година, яка буде введена, перевищує час відправлення якогось поїзда до 1 години, користувачу буде виведена інформація, чи не хоче він почати подорож на годину(або менше) швидше.

Програма працює з часом, використаним з модуля time – currtime.

*Пошук станцій*

Моя програма повністю автоматизувала POST-запити і тепер напряму доступається до сайту

‘’https://booking.uz.gov.ua/train\_search/’

*Пошук поїздів*

Для пошуку необхідно виконати запит на http://booking.uz.gov.ua/train\_search/ , де потрібно буде ввести ІD станції відправлення, ID станції призначення, дата відправлення в форматі dd.mm.yyyy. Ми отримаємо список поїздів, які йдуть по вказаному маршруту та їхню основну інформацію про дату прибуття, дату відправлення, час подорожі і вільні місця.

*Пошук вільних місць( хоча б одного вільного місця) в вибраному вагоні*

Для перегляду вільних місць потрібно виконати запит <http://booking.uz.gov.ua/en/purchase/coach/>, з інформацією про ID станції відправлення, ID станції призначення, номер потяга, кількістю потрібних ліжок та іншою потрібною інформацією. У відповідь ми маємо отримати список вільних місць.

**2)**Яка інформація буде використовуватися і виводитися користувачу

Ресурс Укрзалізниці буде основним запасом даних, яким ми будемо користуватися. Користувач буде отримувати дані:

Маршрут, яким ми будемо рухатися; Інформація про розклад руху поїздів, якими ми подорожуватимемо;

Інформація про час відправлення-прибуття кожного потяга;

.

**JSON** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова)*JavaScript Object Notation*, [укр.](https://uk.wikipedia.org/wiki/%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Українська мова) *запис об'єктів [JavaScript](https://uk.wikipedia.org/wiki/JavaScript" \o "JavaScript)*, вимовляється *джейсон*) — це текстовий **JSON** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова)*JavaScript Object Notation*, [укр.](https://uk.wikipedia.org/wiki/%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Українська мова) *запис об'єктів [JavaScript](https://uk.wikipedia.org/wiki/JavaScript" \o "JavaScript)*, вимовляється *джейсон*) — це текстовий [формат](https://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82) обміну [даними](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96) між [комп'ютерами](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80). JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати [об'єкти](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та інші [структури даних](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85). Цей формат головним чином використовується для передачі структурованої інформації через [мережу](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0) (завдяки процесу, що називають [серіалізацією](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D1%96%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F)). За рахунок своєї лаконічності в порівнянні з [XML](https://uk.wikipedia.org/wiki/XML), формат JSON може бути більш придатним для [серіалізації](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D1%96%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F) складних структур.

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером ([AJAX](https://uk.wikipedia.org/wiki/AJAX)), так і між самими серверами (програмні HTTP-інтерфейси). Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші.

Оскільки JSON представляється синтаксично правильним фрагментом коду JavaScript, природним способом розбору JSON-даних в JavaScript-програмі є використання вбудованої в JavaScript функції eval(), яка призначена для обчислення JavaScript-виразів. При цьому підході відпадає необхідність у використанні додаткових [парсерів](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D1%81%D0%B5%D1%80" \o "Парсер).

Техніка використання eval() робить систему вразливою, якщо джерело JSON-даних, що використовуються, не відноситься до надійних. Такими даними може виступати шкідливий JavaScript-код для атак за допомогою ін'єкції коду. За допомогою цієї вразливості можливо здійснювати крадіжку даних, підробку [автентифікації](https://uk.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F). Проте, вразливість можна усунути за рахунок використання додаткових засобів перевірки даних на коректність. Наприклад, до виконання eval() отримані від зовнішнього джерела дані можуть перевірятися за допомогою [регулярних виразів](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D0%B8%D0%B9_%D0%B2%D0%B8%D1%80%D0%B0%D0%B7). У RFC, що визначає JSON[[1]](https://uk.wikipedia.org/wiki/JSON#cite_note-1) пропонується використовувати такий код для перевірки його відповідності формату JSON.

Останні версії [веб-браузерів](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) мають вбудовану підтримку JSON і здатні його обробляти без виклику функції eval(), що призводить до описаної проблеми. Обробка JSON у такому разі зазвичай здійснюється швидше. Так у червні 2009 року вбудовану підтримку JSON мали такі браузери:

* [Mozilla Firefox](https://uk.wikipedia.org/wiki/Mozilla_Firefox) 3.5+, [SeaMonkey](https://uk.wikipedia.org/wiki/SeaMonkey" \o "SeaMonkey) 2, та [Thunderbird](https://uk.wikipedia.org/wiki/Thunderbird" \o "Thunderbird) 3[[2]](https://uk.wikipedia.org/wiki/JSON#cite_note-2)
* [Microsoft Internet Explorer](https://uk.wikipedia.org/wiki/Microsoft_Internet_Explorer) 8[Native JSON in IE8](http://www.webcitation.org/65OHu3fNT). Архів [оригіналу](http://blogs.msdn.com/ie/archive/2008/09/10/native-json-in-ie8.aspx) за 2012-02-12. (англ.)
* [Opera 10.5](https://uk.wikipedia.org/wiki/Opera)+[[3]](https://uk.wikipedia.org/wiki/JSON#cite_note-3)
* Браузери, засновані на [WebKit](https://uk.wikipedia.org/wiki/WebKit" \o "WebKit) (наприклад, [Google Chrome](https://uk.wikipedia.org/wiki/Google_Chrome" \o "Google Chrome), [Apple Safari](https://uk.wikipedia.org/wiki/Apple_Safari" \o "Apple Safari))[[4]](https://uk.wikipedia.org/wiki/JSON#cite_note-4)

**Розши́рювана мо́ва розмі́тки** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) Extensible Markup Language, скорочено XML) — запропонований консорціумом World Wide Web ([W3C](https://uk.wikipedia.org/wiki/W3C)) стандарт побудови [мов розмітки](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D1%80%D0%BE%D0%B7%D0%BC%D1%96%D1%82%D0%BA%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) ієрархічно структурованих [даних](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96) для обміну між різними [застосунками](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA), зокрема, через [Інтернет](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82).[[1]](https://uk.wikipedia.org/wiki/XML#cite_note-XmlOriginsGoals-1) Є спрощеною підмножиною мови розмітки [SGML](https://uk.wikipedia.org/wiki/SGML). XML-документ складається із текстових знаків, і придатний до читання людиною.

Стандарт XML ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) Recommendation, перше видання від [10 лютого](https://uk.wikipedia.org/wiki/10_%D0%BB%D1%8E%D1%82%D0%BE%D0%B3%D0%BE) [1998](https://uk.wikipedia.org/wiki/1998), останнє, четверте видання [29 вересня](https://uk.wikipedia.org/wiki/29_%D0%B2%D0%B5%D1%80%D0%B5%D1%81%D0%BD%D1%8F) [2006](https://uk.wikipedia.org/wiki/2006)) визначає набір базових [лексичних](https://uk.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BA%D1%81%D0%B8%D0%BA%D0%B0) та [синтаксичних](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81) правил для побудови мови описання інформації шляхом застосування простих тегів. Цей формат достатньо гнучкий для того, аби бути придатним для застосування в різних галузях[[2]](https://uk.wikipedia.org/wiki/XML#cite_note-XMLNut-2). Іншими словами, запропонований стандарт визначає [метамову](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D0%BC%D0%BE%D0%B2%D0%B0), на основі якої шляхом запровадження обмежень на структуру та зміст документів визначаються специфічні, [предметно-орієнтовані](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) мови розмітки даних. Ці обмеження описуються [мовами схем](https://uk.wikipedia.org/wiki/XML_schema) ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) Schema), такими як [XML Schema (XSD)](https://uk.wikipedia.org/wiki/XML_Schema_(W3C)), [DTD](https://uk.wikipedia.org/wiki/Document_Type_Definition) або [RELAX NG](https://uk.wikipedia.org/wiki/RELAX_NG). Прикладами мов, заснованих на XML, є: [XSLT](https://uk.wikipedia.org/wiki/XSLT), [XAML](https://uk.wikipedia.org/wiki/XAML), [XUL](https://uk.wikipedia.org/wiki/XUL), [RSS](https://uk.wikipedia.org/wiki/RSS), [MathML](https://uk.wikipedia.org/wiki/MathML" \o "MathML), [GraphML](https://uk.wikipedia.org/wiki/GraphML" \o "GraphML), [XHTML](https://uk.wikipedia.org/wiki/XHTML), [SVG](https://uk.wikipedia.org/wiki/Scalable_Vector_Graphics), а також [XML Schema](https://uk.wikipedia.org/wiki/XML_Schema_(W3C)).

**Коректність**

Коректний документ ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) well-formed document) відповідає всім синтаксичним правилам XML. Документ, що не є коректним, не може називатись XML-документом. Сумісний синтаксичний аналізатор ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) Conforming parser) не повинен обробляти такі документи. Зокрема, коректний XML-документ має:

* Лише один елемент у корені.
* Непорожні елементи розмічено початковим та кінцевим тегами (наприклад, <пункт>Пункт 1</пункт>). Порожні елементи можуть позначатися «закритим» тегом, наприклад <IAmEmpty />. Така пара еквівалентна <IAmEmpty></IAmEmpty>.
* Один елемент не може мати декілька атрибутів з однаковою назвою. Значення атрибутів перебувають або в одинарних ('), або у подвійних (") лапках.
* Теги можуть бути вкладені, але не можуть перекриватись. Кожен некореневий елемент мусить повністю перебувати в іншому елементі.
* Документ має складатися тільки з правильно закодованих дозволених символів Юнікоду. Єдиними кодуваннями, які обов'язково має розуміти XML-процесор, є UTF-16 та UTF-8. Фактичне та задеклароване кодування ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0" \o "Англійська мова) character encoding) документа мають збігатись. Кодування може бути задекларовано ззовні, як у заголовку «Content-Type» при передачі по протоколу [HTTP](https://uk.wikipedia.org/wiki/HTTP), або в самому документі використанням явної розмітки на самому початку документа. У разі відсутності інформації про кодування, документ має бути в кодуванні UTF-8 (або його підмножині ASCII).

HTML

HTML є похідною мовою від [SGML](https://uk.wikipedia.org/wiki/SGML), успадкувавши від неї [визначення типу документа](https://uk.wikipedia.org/wiki/Document_Type_Definition) та ідеологію структурної розмітки тексту

Попри те, що HTML — штучна комп'ютерна мова, вона не є [мовою програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F).

HTML разом із [каскадними таблицями стилів](https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D0%BA%D0%B0%D0%B4%D0%BD%D1%96_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D1%96_%D1%81%D1%82%D0%B8%D0%BB%D1%96%D0%B2) та [вбудованими скриптами](https://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) — це три основні технології побудови [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0).[[1]](https://uk.wikipedia.org/wiki/HTML#cite_note-w3-1)

HTML впроваджує засоби для:[[1]](https://uk.wikipedia.org/wiki/HTML#cite_note-w3-1)

* створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
* отримання інформації із Всесвітньої мережі через [гіперпосилання](https://uk.wikipedia.org/wiki/%D0%93%D1%96%D0%BF%D0%B5%D1%80%D0%BF%D0%BE%D1%81%D0%B8%D0%BB%D0%B0%D0%BD%D0%BD%D1%8F);
* створення інтерактивних форм;
* включення зображень, звуку, відео, та інших об'єктів до тексту.

Всесвітня павутина складається в основному з HTML-документів, переданих з [веб-серверів](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) для [браузерів](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), використовуючи протокол [HTTP](https://uk.wikipedia.org/wiki/HTTP). До того ж HTTP використовується для передачі зображень, звуків, відео та іншого супутнього контенту. Для правильного відтворення документа браузером окрім нього самого передається ще й інша інформація ([метадані](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D1%96)), у якій зазвичай міститься визначення [MIME-типу](https://uk.wikipedia.org/wiki/MIME) (наприклад, text/html або application/xhtml+xml) та [кодової таблиці](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D1%96%D0%B2) документа.

Пошта

Більшість графічних [поштових клієнтів](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BF%D0%BE%D1%88%D1%82%D0%B0#%D0%9F%D0%BE%D0%BF%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D1%96_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8-%D0%BA%D0%BB%D1%96%D1%94%D0%BD%D1%82%D0%B8_%D0%B4%D0%BB%D1%8F_%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B8_%D0%B7_E-mail) дозволяють використовувати підмножину елементів HTML (часто визначених стандартом як застарілі та не рекомендовані у розмітці), щоб забезпечити візуальне форматування тексту, роблячи неможливим [семантичну](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D0%BF%D0%B0%D0%B2%D1%83%D1%82%D0%B8%D0%BD%D0%B0) розмітку тексту засобами поштового клієнта. Багато таких клієнтів включають в себе [WYSIWYG](https://uk.wikipedia.org/wiki/WYSIWYG)-редактор HTML для складання повідомлень. Використання HTML у повідомленнях електронної пошти є спірним питанням через проблеми з сумісністю у різних поштових клієнтів; також HTML може допомогти приховати [фішинг](https://uk.wikipedia.org/wiki/%D0%A4%D1%96%D1%88%D0%B8%D0%BD%D0%B3" \o "Фішинг).

З моменту свого створення HTML і пов'язані з нею протоколи порівняно швидко отримали визнання. Однак у перші роки існування цієї мови розмітки не було жодних чітких стандартів. Хоча її творці спочатку і задумували HTML як семантичну мову, позбавлену презентаційних можливостей[[12]](https://uk.wikipedia.org/wiki/HTML#cite_note-12), її практичне використання із різними браузерами призвело до додавання багатьох презентаційних елементів і атрибутів в HTML. Останні стандарти, пов'язані з HTML, відображають зусилля з подолання хаотичного розвитку мови і створення раціональної основи для розробки як змістовних, так і виразних документів.[[13]](https://uk.wikipedia.org/wiki/HTML#cite_note-13) Щоб повернути HTML її роль семантичної мови, [Консорціум Всесвітньої павутини](https://uk.wikipedia.org/wiki/W3C) розробив мови стилізування, такі як [Каскадні таблиці стилів](https://uk.wikipedia.org/wiki/CSS) та [Розширена мова таблиць стилів](https://uk.wikipedia.org/wiki/XSL), аби перенести на них відповідальність за вигляд документа. У зв'язку з цим специфікація HTML повільно почала повертатися виключно до семантичних елементів.

Список поїздів: Перегляд вагонів:

Номер Місця

Особистий номер потяга Номер класу,

Модель Клас, чи дозволяють мати бонуси

Модель потяга Кількість місць, що залишилися

Категорія Чи ліжко чи крісло

Пасажирський/Товарний Ціна резервування

Час поїздки Сервіси, які підтримуються

Очікуваний час подорожі Номер місця

Звідки: Номер класу

ID станції

Кожна станція має свій власний

ІД у цій АРІ системі

Назва станції

Дата (особистий номер рейсу)

Дана система реєструє всі рейси

за спеціальними номерами

Дата відправлення

Точний час відправлення і дата

(рік-місяць-день,

година-хвилина-секунда) Перегляд вільних місць:

Куди: Словник вільних місць

ID станції

Назва станції

Дата (особистий номер рейсу)

Дата приїзду

Типи доступних класів:

Розподіл вагонів по більш і менш

затишних місцях

(відповідно за більшу і меншу плату)

Перший клас

К-сть місць

Другий клас

К-сть місць

АРІ дає нам можливість отримати повну інформацію про все, що потрібно. Дані отримуються у форматі .json, який дає можливість отримати цю інформацію. В курсовій роботі я буду застосовувати засоби Python для обробки даних. Програма отримає дані по списку поїздів, часу прибуття і відправлення. Після цього вона отримує інформацію про кількість доступних місць (нам потрібно принаймні одне).