

УДК 004

М.Єрємєєв, магістр гр. КІ-22М-1

Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ БРАУЗЕРУ ДЛЯ ПЕРЕГЛЯДУ ВЕБ-САЙТІВ

У статті розроблено програмне забезпечення, яке призначено для системи браузеру для перегляду веб-сайтів. Метою розробки є дослідження та програмна реалізація системи браузеру для перегляду веб-сайтів. Об'єктом дослідження є процес браузеру для перегляду веб-сайтів. Предметом дослідження є методи браузеру для перегляду веб-сайтів. Методи дослідження базуються на методах теорії телетрафіку, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи браузеру для перегляду веб-сайтів. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Постановка проблеми. Веб-браузери – це програми, які допомагають переглядати веб-сайти в Інтернеті. Ось що вони роблять:

- Отримання файлів: коли ви вводите веб-адресу або натискаєте посилання, браузер запитує у сервера файли веб-сайту. Це включає HTML, CSS, JavaScript, зображення тощо.
- Інтерпретація коду: браузер читає код HTML, CSS і JavaScript із веб-сайту та визначає, як має виглядати та працювати сторінка.
- Візуалізація вмісту: браузер об'єднує сторінку на вашому екрані за допомогою прочитаного ним коду. Це стосується всього тексту, зображень і способу компоновання сторінки. Сторінка може змінюватися, коли ви натискаєте або взаємодієте з нею.

Такі веб-переглядачі, як Chrome, Firefox, Safari та Edge, усі це роблять, але вони намагаються бути швидшими, простішими у використанні та мають більше функцій, ніж їхні конкуренти.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-20] було виявлено певні прогалини у забезпеченні системи браузеру для перегляду веб-сайтів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи браузеру для перегляду веб-сайтів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем браузеру для перегляду веб-сайтів.
- Дослідження системи браузеру для перегляду веб-сайтів.
- Програмна реалізація системи браузеру для перегляду веб-сайтів.

Об'єктом дослідження є процес браузеру для перегляду веб-сайтів.

Предметом дослідження є методи браузеру для перегляду веб-сайтів.

Методи дослідження базуються на методах теорії телетрафіку, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Щоб створити базовий веб-браузер, вам потрібно знати кілька основних мов програмування:

- HTML: це основна мова для створення веб-сторінок. Це допомагає вам об'єднати основні частини сторінки, наприклад заголовки, абзаци та зображення.

– CSS: ця мова робить веб-сторінки гарними. Це дозволяє змінювати кольори, шрифти та спосіб розташування речей на сторінці.

– JavaScript: він змушує веб-сайти щось робити. Наприклад, він може показувати повідомлення, перевіряти, чи має сенс те, що ви ввели у форму, або створювати меню, які відкриваються та закриваються, коли ви клацаєте.

Ви також можете почути про C++, який може допомогти у складніших завданнях. Але для початку достатньо зосередитися на HTML, CSS і JavaScript.

Інструменти та програмне забезпечення

Ось кілька основних інструментів, які вам знадобляться:

– Текстовий редактор: проста програма, у якій можна писати та редагувати свій код. Серед популярних – VS Code, Sublime Text і Atom.

– Механізм веб-браузера: це серце вашого браузера. Це те, що бере код із веб-сайту та показує його на вашому екрані. Деякі добре відомі двигуни включають Chromium, Gecko та WebKit.

– Бібліотека інтерфейсу користувача: це як будівельні блоки для зовнішнього вигляду вашого браузера, включаючи вікна та кнопки. Два приклади – GTK+ і Qt.

– Інструменти налагодження: вони допомагають знаходити та виправляти помилки у коді. Зазвичай вони включені в текстові редактори та веб-браузери.

– Контроль версій: це як функція збереження вашого коду, яка також дозволяє повернутися до старіших версій. Git і GitHub є популярними варіантами.

З цими мовами та інструментами ви готові розпочати роботу над власним веб-переглядачем. Далі ми крок за кроком розглянемо планування вашого проекту.

Вибір механізму візуалізації

У цій частині йдеться про механізми рендерингу, такі як Blink, Gecko та WebKit. Ми порівняємо їх, щоб допомогти вам вибрати один для свого браузера.

Огляд механізмів візуалізації

Механізми візуалізації – це мозок веб-браузерів. Вони беруть код із веб-сайтів (наприклад, HTML, CSS і JavaScript) і перетворюють його на веб-сторінки, які ви бачите та з якими взаємодієте.

Ось деякі з них, про які ви могли чути:

– Blink – Chrome, Opera та новіша версія Edge використовують це. Це прийшло з WebKit.

– Gecko - Firefox використовує цей. Це зроблено Mozilla.

– WebKit - двигун Safari, створений Apple.

Усі ці механізми працюють, щоб швидко й точно перетворювати код веб-сайту на сторінки. Вони змагаються в тому, наскільки вони швидкі, наскільки добре вони працюють з різними веб-матеріалами та які додаткові речі вони можуть робити.

Blink надзвичайно швидкий і дозволяє вам багато чого змінити, тому що це те, на чому побудовано Chrome та інші. Gecko, який використовується Firefox, не використовує так багато пам'яті вашого комп'ютера та має чудові інструменти для людей, які створюють веб-сайти. WebKit, який використовує Safari, також не потребує великої потужності.

Вибираючи механізм візуалізації, подумайте про те, на яких пристроях ваш веб-переглядач має працювати, наскільки він має бути швидким, наскільки ви хочете його змінити та яку допомогу та поради ви можете отримати.

Налаштування середовища розробки

Перш ніж почати створювати свій веб-браузер, вам потрібно підготувати комп'ютер із потрібними інструментами. Ось як це зробити простими кроками:

Встановіть текстовий редактор

По-перше, вам потрібно місце для написання коду. Ось декілька простих у використанні варіантів:

– Код Visual Studio – популярний і має багато корисних функцій для кодування.

– Atom – створений GitHub, він дозволяє багато налаштовувати та додавати додаткові функції.

– Sublime Text – це швидко та просто, але після безкоштовної спроби ви можете купити його.

Додавання таких речей, як підсвічування коду та автозавершення, може полегшити кодування, тому шукайте ці розширення.

Виберіть і налаштуйте механізм візуалізації

Механізм візуалізації перетворює ваш код на веб-сторінки, які бачать люди. Для початківців механізм Blink (як у Chrome) є хорошим місцем для початку. Вам потрібно буде завантажити деякі файли та налаштувати речі, дотримуючись інструкцій для Blink.

Встановіть засоби налагодження та тестування

Інструменти налагодження допомагають знаходити та виправляти помилки. Ваш текстовий редактор може мати деякі, але також перевірте:

– Valgrind – він знаходить такі проблеми, як витік пам'яті.

– Selenium – цей інструмент автоматизує веб-переглядачі для тестування ваших сторінок.

Тестування часто допомагає виявити проблеми на ранній стадії.

Використовуйте контроль версій

Контроль версій схожий на функцію збереження, яка відстежує всі ваші зміни. Якщо щось піде не так, ви можете повернутися до версії, яка працювала.

GitHub – це зручне місце для безпечного зберігання вашого коду в Інтернеті. Він використовує Git, систему відстеження змін.

Завдяки цим крокам ваш комп'ютер готовий до того, щоб почати створювати веб-браузер. Пам'ятайте, що починайте з простого, тестуйте по ходу роботи та використовуйте такі інструменти, як контроль версій і налагодження, які допоможуть вам на цьому шляху.

Побудова базової структури

Створення нового проекту

Щоб розпочати проект веб-браузера, створіть нову папку на своєму комп'ютері, щоб упорядкувати всі файли коду. Відкрийте код Visual Studio, натисніть «Відкрити папку» та виберіть нову папку.

Тепер давайте налаштуємо основні файли, які вам знадобляться:

– index.html- Цей файл міститиме основний HTML-код, який є скелетом вашого браузера.

– script.js- Тут ви будете писати код JavaScript, який змусить ваш браузер виконувати певні дії.

По ходу роботи ми додамо більше деталей до частин голови та тіла.

Проектування інтерфейсу користувача

Тепер давайте створимо простий інтерфейс користувача для нашого браузера, який включає:

– Місце для введення веб-адрес (рядок URL-адреси)

– Кнопки для повернення назад, вперед і оновлення сторінки

– Вкладки, щоб тримати відкритими кілька сторінок одночасно

Пізніше ми використаємо JavaScript, щоб ці кнопки працювали з браузером.

Нарешті, створіть місця для показу відкритих вкладок і вмісту веб-сторінок. Використання додаткових елементів HTML, таких як <div> і <iframe> може допомогти вам упорядкувати ці розділи.

Під час створення тестуйте свій інтерфейс за допомогою Live Server, щоб побачити зміни в реальному часі.

Реалізація основних функцій

Елементи керування навігацією

Щоб наш веб-браузер працював як справжній, нам потрібно, щоб кнопки «Назад», «Вперед» і «Оновити» виконували свою роботу. Ось простий спосіб зробити це за допомогою JavaScript:

Ми можемо використовувати спеціальні інструменти браузера, які називаються historyAPI, щоб переміщатися назад і вперед по сторінках. Для оновлення ми просто перезавантажуємо поточну сторінку.

Завантаження URL-адреси

Щоб дозволити людям відвідувати веб-сайт, ввівши його адресу, ми робимо наступне:

- Знайдіть місце для введення адреси.
- Змусити його щось робити, коли ми натискаємо Enter.
- І ось як ми завантажуємо веб-сайт.

Зробивши це, ми додали ключові функції, необхідні нашому браузеру для переходу на веб-сайти та навігації сторінками!

Використання механізму візуалізації

Використання механізму візуалізації є ключовим, коли ви хочете створити веб-браузер. Цей механізм перетворює весь код веб-сайту на веб-сторінки, з якими ви можете взаємодіяти. Давайте крок за кроком розберемо, як додати такий механізм, як Blink, до вашого проекту браузера.

Вибір двигуна

Раніше ми розглядали такі движки, як Blink, Gecko та WebKit. Для тих, хто тільки починає, Blink – чудовий вибір, оскільки Chrome використовує його та містить багато корисних посібників. Gecko і WebKit також є хорошими варіантами. Виберіть той, який найкраще відповідає вашим потребам щодо швидкості, на яких пристроях він працює та наскільки ви можете його налаштувати.

Завантаження необхідних файлів

Для роботи кожного механізму візуалізації потрібні певні файли. Якщо ви використовуєте Blink, ви можете завантажити файли, готові до використання. Ці файли допомагають вашому браузеру зрозуміти та показати код веб-сайту.

Налаштування інструментів збірки

Вам знадобляться деякі інструменти розробника, щоб підготувати механізм візуалізації для вашого проекту. Це можуть бути такі інструменти, як Make, Ninja або Visual Studio. Кожен механізм має власний посібник із налаштування, тому дотримуйтеся посібника для свого двигуна. Щоб підготувати ці інструменти, може знадобитися трохи зусиль.

Інтеграція з вашим кодом

Тепер вам потрібно підключити інтерфейс користувача та функції вашого браузера до механізму рендерингу. Якщо ви використовуєте Blink, ви можете використовувати його код C++, щоб робити такі дії, як відкриття веб-сторінок. Можливо, вам знадобиться використовувати спеціальні інструменти, щоб змусити код C++ працювати з вашим кодом JavaScript.

Тестування та налагодження

Обов'язково перевірте свій браузер, щоб виявити будь-які проблеми на ранній стадії. Використовуйте інструменти налагодження та тестуйте на різних веб-сайтах і пристроях. Таким чином ви можете переконатися, що ваш браузер добре працює для реальних користувачів.

Виконуючи ці кроки, ви можете додати механізм візуалізації до свого проекту браузера. Це дозволить вашому браузеру відображати веб-сторінки. Пам'ятайте, що вам може знадобитися налаштувати деякі речі залежно від вашої системи та того, як ви створюєте свій браузер.

Додавання основних веб-функцій

Закладки

Щоб полегшити користувачам відстеження своїх улюблених веб-сторінок, ви можете додати функцію створення закладок. Ось простий спосіб зробити це:

Зберегти закладки:

– Додайте кнопку «Зберегти закладку», яка зберігає веб-адресу, заголовок та інші відомості поточної сторінки.

– Зберігайте інформацію про закладку в сховищі браузера або простому файлі.

Показати збережені закладки:

– Майте бічну область або меню зі списком усіх збережених закладок.

– Показувати назву кожної закладки, веб-адресу та час її збереження.

– Дайте можливість видалити закладки, якщо потрібно.

Відкрийте Закладки:

– Зробіть кожну закладку посиланням, яке можна натиснути.

– Натискання закладки має відкрити цю веб-сторінку в поточному вікні.

Упорядкуйте закладки

– Дозвольте користувачам створювати папки для сортування своїх закладок.

– Дозволити сортування закладок, наприклад упорядкування їх по порядку.

Синхронізація закладок:

– Запропонуйте спосіб зберігати однакові закладки на різних пристроях.

– Використовуйте онлайн-сервіс для оновлення закладок, де б ви не користувалися своїм браузером.

Починаючи з цих кроків, ви отримаєте базовий спосіб керування закладками. Ви завжди можете додати інші функції пізніше.

Перегляд із вкладками

Ось як налаштувати систему для відкриття кількох веб-сторінок одночасно:

Панель вкладок:

– Створіть угорі розділ для вкладок, які відображатимуть заголовки відкритих веб-сторінок.

– Виділіть вкладку, яку ви зараз переглядаєте, щоб виділити її.

Відкрити вкладки:

– Відкривати посилання в новій вкладці після натискання.

– Додайте можливість відкривати посилання в новій вкладці за допомогою меню, що відкривається правою кнопкою миші.

Перемикання вкладок:

– Натисніть вкладку, щоб переглянути цю веб-сторінку.

– Змініть вигляд активної вкладки, щоб показати, що вона використовується.

Закрити вкладки:

– Розмістіть невелику кнопку закриття на кожній вкладці, щоб користувачі могли її легко закрити.

– Дозволити закривати поточну вкладку середньою кнопкою миші.

Керування вкладками:

– Тримайте під контролем кількість відкритих вкладок.

– Повторно відкривайте веб-сторінки в наявних вкладках замість того, щоб завжди відкривати нові.

Завдяки цим крокам у вас буде простий спосіб використання кількох вкладок. Пізніше ви зможете додати розширені функції, як-от упорядкування вкладок у групи, показ попереднього перегляду та синхронізацію вкладок на різних пристроях.

Тестування вашого браузера

Перш ніж ділитися ним з іншими, дуже важливо перевірити свій браузер. Це допомагає вам виявляти та виправляти проблеми, гарантуючи, що все працює гладко для всіх, хто ним користується. Ось інструкція щодо ефективного тестування браузера:

Тестування функціональності

– Переконайтеся, що всі кнопки та функції виконують те, що вони повинні робити:

○ Кнопки навігації дозволяють легко переміщатися вперед і назад між сторінками.

○ Закладки дозволяють без проблем зберігати та відкривати веб-сторінки.

- Вкладки дозволяють відкривати, перемикатися та закривати різні сторінки.
- Спробуйте ввести адреси веб-сайтів і натиснути посилання, щоб перевірити, чи правильно завантажуються сторінки.

- Подивіться, як швидко ваш браузер завантажує сторінки, і переконайтеся:

- Зображення та відео відображаються правильно.
- Текст і посилання виглядають так, як повинні.
- Макет сторінки виглядає добре та відповідає дизайну веб-сайту.

Кросбраузерне тестування:

- Спробуйте свій веб-переглядач на різних веб-переглядачах, наприклад Firefox, Edge і Safari.

- Переконайтеся, що ваш браузер працює однаково на всіх них.

- Порівняйте, як веб-сторінки виглядають у різних браузерах.

- Виправте будь-які відмінності, щоб переконатися, що всі отримують однаковий досвід.

Тестування на реакцію:

- Перевірте свій веб-переглядач на пристроях із різними розмірами екрана, наприклад на телефонах, планшетах і комп'ютерах.

- Переконайтеся, що веб-сторінки налаштовані так, щоб їх було легко читати та використовувати на менших екранах.

- Перевірте, чи легко торкатися кнопок і посилань на сенсорних екранах.

Тестування доступності:

- Використовуйте інструменти для людей з обмеженими можливостями, як-от програми зчитування з екрана, щоб перевірити свій веб-переглядач.

- Переконайтеся, що ви можете використовувати браузер лише за допомогою клавіатури.

- Переконайтеся, що текст легко читається на його фоні.

Тестування безпеки:

- Шукайте ризики безпеки, як-от XSS і впровадження коду.

- Перевірте, чи безпечні сайти, захищені паролем.

- Переконайтеся, що ваш браузер захищає конфіденційну інформацію.

Тестування продуктивності:

- Подивіться, як працює ваш браузер із великою кількістю відкритих вкладок і вікон.

- Переконайтеся, що з часом він не використовує забагато пам'яті.

- Знайдіть і виправте будь-які місця повільного завантаження сторінок.

Багато тестування на ранній стадії допомагає виявити та виправити невеликі проблеми, перш ніж вони стануть великими. Це покращує ваш браузер і залишає користувачів задоволеними.

Розповсюдження та наступні кроки

Після того, як ви створили власний веб-браузер і переконалися, що він працює добре, ви можете дозволити іншим людям спробувати його. Ось як зробити свій веб-переглядач доступним для інших і постійно покращувати його:

Упаковка вашого браузера:

Щоб надати спільний доступ до свого браузера, вам потрібно зібрати всі необхідні файли:

- Створюйте програми налаштування для різних операційних систем, таких як Windows, Mac і Linux.

- Включіть усе, наприклад механізм (Blink, Gecko тощо), JavaScript, CSS і зображення.

- Заздалегідь визначте параметри браузера та параметри за замовчуванням.

- Використовуйте унікальні піктограми та логотипи, щоб ваш браузер виділявся.

Ви також можете зробити:

- Посібники, які показують, як встановити ваш браузер.

– Файли "Read me", які повідомляють користувачам, що включено та як повідомити про проблеми.

Подивившись на те, наскільки великі браузери це роблять, може бути дуже корисно. Спільне використання та просування вашого браузера.

Тепер, коли він готовий, повідомте людям про ваш браузер:

- Розмістіть його для завантаження на своєму веб-сайті або GitHub.
- Розкажіть про це на форумах і в соціальних мережах.
- Створіть відео, які демонструють, що може робити ваш браузер і як його налаштувати.
- Зверніть увагу на те, що говорять користувачі, щоб зробити ваш браузер ще кращим.

Якщо ви очікуєте, що ваш браузер завантажить багато людей, подумайте про використання таких служб, як AWS або Azure.

Створення веб-браузера – велике досягнення. Використовуйте цей досвід, щоб продовжувати навчатися:

- Подивіться на браузер з відкритим кодом, щоб дізнатися більше про їх роботу.
- Спробуйте додати складніші функції, наприклад розширення.
- Пориньте в такі теми, як зробити ваш веб-переглядач швидшим, доступнішим і безпечнішим.
- Подумайте про те, щоб вивчити більше мов програмування, які використовуються в браузерах.
- Ресурси з таких місць, як Mozilla та проект Google Chromium, можуть бути справді корисними.

Потенційні додаткові функції

Коли ваш браузер працює нормально, подумайте про додавання додаткових функцій:

- Синхронізація – надайте користувачам доступ до своїх закладок, історії та відкритих вкладок на будь-якому пристрої.
- Розширення. Дайте можливість розробникам створювати додаткові компоненти, які додають нові функції у ваш браузер.
- Синхронізація налаштувань. Зберігайте налаштування користувача однаковими на різних пристроях.
- Керування вкладками. Полегшіть роботу з великою кількістю відкритих вкладок.
- Пошук в історії – допоможіть користувачам знайти веб-сайти, які вони відвідували раніше.
- Інструменти веб-розробника – додайте функції для людей, які створюють веб-сайти, щоб тестувати та виправляти свої сторінки.

Є багато інших речей, які ви можете додати, щоб зробити свій браузер ще кращим!

Створення власного веб-браузера – чудовий спосіб зануритися в роботу Інтернету. Ознайомившись із цим посібником, ви навчилися основам веб-браузерів, а також дізналися, як їх створити.

Ми говорили про основні частини, такі як HTML, CSS і JavaScript, які є будівельними блоками для веб-сайтів. Ми також обговорили інструменти для кодування та ключові частини веб-переглядача, такі як інтерфейс користувача та механізм, завдяки якому веб-сайти відображаються на вашому екрані.

Вам вдалося створити простий браузер, який може показувати веб-сайти, переходити між сторінками та відстежувати ваші улюблені сайти. Тестування браузера в різних ситуаціях є ключовим для пошуку та вирішення проблем.

Створення веб-переглядача з усіма фантастичними функціями потребує більше зусиль, але почати з чогось базового – чудовий спосіб почати навчання. Подивіться на такі популярні браузери, як Firefox і Chrome, щоб побачити, як вони справляються зі складнішими проблемами, такими як швидкість, безпека, додавання додаткових функцій і надання користувачам більшого контролю.

Об'єднання зусиль з іншими розробниками проектів браузера може допомогти вам швидше навчатися. Продовжуючи навчатися та стикаючись із новими викликами, ви станете кращими у веб-розробці. З кожною новою версією ваш браузер буде захоплюючим покращенням. Розпочати цей простий проект – це перший крок до створення веб-переглядачів, які забезпечать чудовий досвід для користувачів у всьому світі.

Браузер, що розробляється, дозволяє працювати з декількома документами усередині одного робочого вікна додатка. Перемикання між документами відбувається за допомогою вкладок. На відміну від конкурентів, де вкладки використовуються тільки для перемикання між web-сторінками, у браузері, що розробляється, подібним способом ви можете перемикатися між всіма інструментами пакета.

У багатьох браузерах панель вкладок розташовується нижче панелі інструментів. І це логічно, адже незалежно від того, яка інформація буде відображатися у вкладках, набір інструментів єдиний для будь-якого випадку. Говорячи про браузер, що розробляється, ми повинні пам'ятати про те, що до складу пакета включено кілька утиліт, і web-сторінка – лише окремий випадок застосування вкладки. Як наслідок, залежно від типу документа міняється панель інструментів. Вкладка – батьківський елемент стосовно панелі інструментів, а виходить, необхідний такий підхід, який би наочно відображав всю ієрархію елементів інтерфейсу. Із цієї причини панель вкладок розташовується вище інших елементів.

Стандартна панель інструментів складається із шести кнопок. Крім звичних елементів навігації "Назад" і "Уперед", у браузер, що розробляється, присутні кнопки швидкого переходу, за допомогою яких ви можете перейти на перший або останній елемент ланцюжка сторінок, що відкривалися по внутрішніх посиланнях. З іншого боку, у браузері, що розробляється, відсутній список останніх сторінок, які були відвідані, викликуваний стандартними навігаційними кнопками. Кнопки "Стоп" і "Обновити" об'єднані в один елемент, що міняє свій стан залежно від стану завантаження поточної сторінки. Далі треба кнопка "Жезл", що дозволяє швидко вводити користувальницькі дані при повторному відвідуванні сайтів.

Праворуч від основних інструментів розташовується рядок уведення адреси. Перехід до уведення адреси відкриває додаткове меню, з якого ви можете перейти на домашню сторінку, відкрити дерево закладок, а також відвідати кожної з десяти самих популярних сайтів. Гаряча десятка формується на основі збору статистики про раніше сайти, які були раніше відвідані.

Праворуч від рядка адреси розташовується вікно пошукових запитів. Ви можете ввести будь-яке ключове слово, і браузер автоматично буде здійснювати пошук, використовуючи обраний пошуковий сервер. За замовчуванням використовується Google, українських пошукових серверів немає.

У правому кутку розташовується кнопка "Вид", за допомогою якої викликається додаткова панель властивостей. Ви можете робити пошук на сторінці, здійснювати голосове керування, вибирати режим відображення сторінки (стилі), міняти режими завантаження графіки, стискати будь-який сайт до ширини робочого вікна й, нарешті, масштабувати сторінку. Зміна масштабу в браузері, що розробляється, застосовується до всієї сторінки відразу, а не тільки до текстової інформації.

Ви можете управляти показом тих або інших панелей, а також додавати або видаляти на них кнопки. Клацання правою кнопкою миші на будь-якій панелі відкриває меню, у якому два пункти – видалити поточний елемент і виклик загальних налаштувань. У дистрибутиві браузера, що розробляється, включена величезна кількість кнопок, за допомогою яких можна реалізувати безліч різноманітних дій.

Інтерфейс браузера, що розробляється, гнучко настроюється, дозволяючи надавати браузеру неповторні риси. Ви можете забрати практично всі панелі, залишивши у своєму розпорядженні тільки саме необхідне, або, навпроти, улаштувати в робочому вікні справжній маджонг.

Браузер, що розробляється, допускає зміну тем оформлення, яких у мережі величезна безліч. Завантаження й установка тим оформлення в браузері, що розробляється, реалізована за допомогою невеликого браузера, що обробляє дані, одержувані з офіційного сайту. Ви вибираєте тему, що сподобалася, оформлення, завантажуєте її й можете відразу застосувати налаштування без перезавантаження програми.

Низькі системні вимоги браузера, що розробляється, дозволяють успішно працювати навіть на морально застарілому обладнанні.

Принцип завантаження й відображення web-сторінок у браузері відбувається по іншому принципі, відмінному від Microsoft Edge. Як тільки починається завантаження сторінки, браузер практично відразу намагається відобразити отримані дані. Широкополосне підключення до мережі, завдяки високій швидкості передачі даних, нівелює різницю, однак при використанні з'єднання, що комутирується, створюється ілюзія більше високої швидкості завантаження сторінки. Ви можете перервати процес у будь-який момент і почати завантаження іншої сторінки.

Особливої уваги заслуговує кеш браузера, що розробляється. Ви можете включати й відключати показ зображень, а також перемикатися в спеціальний режим, при якому допускається показ зображень тільки з кеша. Даний режим дозволяє заощаджувати трафік за рахунок зниження обсягів передачі графічних даних.

Переходи "Уперед" і "Назад" у браузері, що розробляється, виконуються миттєво. Браузер не обробляє web-сторінку заново, а просто бере дані з кеша в оперативній пам'яті. При цьому підвищується швидкість навігації по Мережі, однак страждає актуальність. За те час, поки ви не відвідували сторінку, вона може змінитися. Браузер, що розробляється, не відслідковує зміни сторінок у кеші. При навігації по сторінках зі статичною інформацією ви практично нічим не ризикуєте, однак робота зі швидко мінливими даними (наприклад, форуми) сильно утрудняється. Втім, ви можете завжди перевірити актуальність сторінки, перезавантаживши її з мережі.

При завантаженні великих файлів з мережі, ви можете перервати процес, а пізніше продовжити завантаження з перерваного місця. Якщо ваш зв'язок з мережею часто рветься, то подібна властивість браузера, що розробляється, дозволить заощадити час і трафік, тому що відповідає необхідності у повторному завантаженні файлів.

Допускається застосування "дзеркальних" налаштувань клавіш миші для лівш. Налаштування мишачих жестів здійснюється лише шляхом ручного введення імен всіх елементів. Крім того, не можна включити відображення сліду під час малювання жесту для контролю над правильністю виконання операції.

Для прискорення введення адрес сайтів у браузері, що розробляється, реалізований механізм завдання скорочень. З панелі закладок, за допомогою правої кнопки миші можна викликати властивості папки, а також властивості будь-якої вхідної в неї закладки. Ви можете задати коротке ім'я для елемента, що редагується.

Після цього, якщо в адресному рядку браузера ввести задане раніше ім'я, то відкриються сторінки, що належать даному елементу. Крім того, будь-яку закладку можна відкривати усередині вікна особистої панелі. Дана функція дозволяє зручно переглядати PDA-версії сайтів.

При інтенсивній роботі із браузером ви відкриваєте й закриваєте безліч вкладок. Якщо ви бажаєте знову потрапити на який-небудь із раніше відвіданих сайтів, то можете скористатися спеціальним кошиком, де втримується список закритих вкладок.

Крім закритих вкладок, у списку кошика перебувають заблоковані спливаючі вікна. Включення блокування спливаючих вікон і деяких інших операцій здійснюються за допомогою виклику вікна швидких налаштувань (F12).

За допомогою швидких налаштувань можна міняти деякі властивості поточної вкладки, список яких відображений на останньому малюнку.

Працюючи в браузері, що розробляється, ви можете підключати користувальницькі стилі для зміни правил відображення сторінок. Застосування користувальницьких стилів

може бути доречно в різних областях. Блокування небажаної інформації, відображення числа тегів на поточній сторінці, перетворення форматування з HTML в BBCode, робота з язовими атрибутами й багато чого іншого можуть бути здійснені за допомогою технології UserCSS.

Крім браузера, у дистрибутиві браузера, що розробляється, включено кілька компонентів, що мають лише непряме відношення до web-серфінгу.

Поштовий клієнт M2, що входить до складу браузері, що розробляється, у момент першої появи був досить прогресивним інструментом у своєму класі. Згодом конкуренти підтяглися, але й сьогодні M2 має достатню кількість можливостей, щоб з успіхом застосовуватися для електронної переписки.

M2 працює тільки з єдиною поштовою базою, як це прийнято в Outlook Express, не дозволяючи розбивати облікові записи на окремі групи папок. З іншого боку, єдина база дозволяє здійснювати більше швидкий пошук інформації. Результати пошуку можна зберігати в окремі папки, які по зовнішньому вигляді нічим не відрізняються від фізичних аналогів. Всі вкладені файли групуються по типах, і ви можете швидко переміщатися між повідомленнями, що містять вкладення.

Поштовий клієнт дозволяє працювати також і із групами новин. Для цього необхідно створити обліковий запис відповідного типу.

Фільтрація реклами відбувається на основі того, якого навчають, алгоритму. Ви можете вибрати кілька ступенів захисту, тим самим, регулюючи агресивність фільтра.

У поштовому клієнті дотепер відсутній HTML-редактор, що унеможливує додавання якого-небудь форматування в повідомлення.

До складу браузера, що розробляється, входить також простий IRC-клієнт.

Ви створюєте обліковий запис за допомогою тих же засобів, що й у випадку з електронною поштою, указуючи відповідний тип. Надалі потрібно лише вказати конкретний обліковий запис, що варто використовувати для підключення до необхідного сервера. IRC-клієнт дозволяє працювати на каналах з будь-яким кодуванням, що використовує кирилицю. Програма підтримує відображення графічних смайликів і уведення керуючих кодів. Ви можете повністю міняти зовнішній вигляд робочого вікна IRC-клієнта за допомогою користувальницьких стилів, які необхідно завантажувати з мережі.

Поштову програму й IRC-клієнт поєднує загальний менеджер контактів.

Ви можете додавати нові записи, вибираючи контакти як серед учасників бесід на IRC-каналах, так і серед тих, хто надсилає вам електронні листи. При цьому кожному контакту можна привласнити власний значок. Після цього даний значок буде відображатися напроти ім'я учасника бесіди на IRC-каналі. Якщо серед інформації, занесеної в контакт, є його електронна адреса, то клацання на даному елементі списку відкриває вікно створення нового повідомлення.

Ще одним цікавим інструментом є можливість складання невеликих заміток, які можна сортувати по папках.

Ви відкриваєте бічну панель заміток і створюєте нові елементи. У верхній частині панелі втримується список заміток, а в нижньої – вікно редагування тексту. За замовчуванням подвійне клацання по будь-якій замітці відкриває вікно створення електронного повідомлення із впровадженим текстом з поточного елемента списку. Якщо поточна замітка була створена з матеріалів якого-небудь сайту, то подібна дія відкриває web-сторінку – джерело інформації.

Як ми бачимо, у дистрибутиві браузера, що розробляється, включається безліч дрібних утиліт, тісно інтегрованих один з одним. Функціональні можливості компонентів помітно уступають популярним незалежним додаткам, однак, завдяки вдалим механізмам взаємодії, становлять єдине ціле, по своїх можливостях не має рівних в усьому світі програмного забезпечення. Як часто буває в подібних ситуаціях, відповідь залежить від того, з якої сторони розглядається питання.

Жорстока конкуренція змушує шукати розроблявачам нові шляхи для підтримки інтересу до своїх продуктів. Браузер, що розробляється, обзавівся цікавою новинкою, що дозволяє розширити її можливості – віджетами. Ви можете встановлювати із браузера невеликі додатки, що виконуються на робочому столі.

Більшість віджетів не мають ніякого відношення до роботи браузера, хоча іноді зустрічаються й приємні виключення.

Під час роботи з великою кількістю відкритих документів, заголовки вкладок стають що не читаються. Для того щоб не перегортати щораз весь список відкритих вкладок у пошуках потрібного документа, у браузері, що розробляється, реалізований механізм створення й відображення мініатюр відкритих web-сторінок. Ви підводите мишу до довільної вкладки, якийсь час проводите чекаючи, після чого з'являється невелике вікно, що містить зменшену копію документа зі службовою інформацією (MIME-Тип, тег Title, web-адреса).

Мініатюра створюється після першого завантаження сторінки й не міняється у випадку втрати актуальності (зміни) її вмісту.

У браузері, що розробляється, з'явилися засоби вибіркового блокування завантаження зображень і flash-анімації. Для активізації процедури блокування необхідно вибрати відповідний пункт із контекстного меню браузера.

Якщо ви клацніть мишею по будь-якому зображенню, то це буде означати повну заборону завантаження всіх графічних елементів на даній сторінці. При втриманні клавіші Shift відбувається блокування тільки конкретного зображення. Після того, як процедура блокування кінчена, можна переглянути список заборонених для завантаження елементів, нажавши кнопку "Додатково". браузер, що розробляється, не підтримує регулярні вираження для завдання правил блокування. Знак "*" означає блокування всіх елементів.

Ви можете встановлювати для кожного відвідуваного сайту власні налаштування поведінки браузера. Вікно налаштувань складається з декількох вкладок.

Таблиця 1 – Індивідуальні налаштування для сайтів

Ім'я вкладки	Список налаштувань
Загальні	Адреса сайту, правила блокування спливаючих вікон
Cookies	Правила прийому й видалення Cookies, ручне виправлення елементів списку
Уміст	Включення/вимикання анімації GIF і SVG, звуків, Java і плагінів
Відобразити	Включення/вимикання фреймів, стилів форм, оформлення смуг прокручування. Можливість підключення власної таблиці стилів
Сценарії	Включення/вимикання JavaScript, зміни розмірів вікон і рядка стану, обробки клацань правою кнопкою миші, приховання рядка адреси, відкриття консолі при помилці. Можливість підключення власного файлу JavaScript
Мережа	Включення/вимикання кодування міжнародних web-адрес в UTF-8, вказівки джерела переходу, автоматичного перенапряму. Можливість зміни рядка ідентифікації браузера

Можна згадати чимало прикладів, коли виникає необхідність у нестандартних налаштуваннях. Спливаючі вікна, що найчастіше носять рекламний характер, іноді бувають дуже корисні. На деяких форумах таким способом робиться акцент на особистих повідомленнях.

У браузері, що розробляється, є підтримка BitTorrent. Завантаження інформації з файлообмінної мережі нічим не відрізняється від роботи зі звичайним сервером.

Ви можете одержати доступ до повного списку налаштувань браузера, набравши в адресному рядку команду `mybrowser:config`. Для простих користувачів дана функція практично марна, але під час бета-тестування фахівці могли перевіряти роботу тих або інших прихованих можливостей.

Будь-яку вкладку можна заблокувати щоб уникнути випадкового закриття. У браузері, що розробляється, є інструмент перегляду вихідного коду сторінок. При повному збереженні сторінки на жорсткому диску створюється окрема папка для зображень.

Розробка структурної схеми

Структурна схема розробленої системи зображена на рисунку 1. При написанні різних браузерів спостерігається таке явище: основна кількість браузерів роблять на якихось певних движках.

Разом з тим продукти, зроблені з використанням движків, успадковують їхні помилки й проблеми безпеки.

Найбільше браузерів зроблено на движку Microsoft Edge, це й не дивно, тому що браузер древній як динозавр, і ще не швидко він вимре, тому що його розвитком і підтримкою займається Майкрософт. На розробку браузерного движку потрібні чималі ресурси, тому програмістам легше робити різні оригінальні примочки й доповнення до движка, іменованому Microsoft Edge, і називати цей браузер уже своїм, оригінальним ім'ям, хоча це все той же Microsoft Edge тільки модифікований як у кращу, так і в гіршу (що буває досить часто) сторону.

У магістерському проекті за основу браузера взято движок Trident. Trident (також відомий як MSHTML) – назва браузерного движка для Windows-версії Microsoft Microsoft Edge. Уперше Trident був реалізований у четвертій версії Microsoft Edge, і з тих пор піддавався постійним поліпшенням і переробкам.

Trident був розроблений як програмний компонент, що дозволяв розроблювачам програмного забезпечення легко додавати можливість перегляду веб-сторінок у їхні власні додатки. Він використовує технологію COM (компонентну модель об'єктів) для перегляду й виправлення веб-сторінок у будь-якому оточенні, що підтримує інтерфейс цієї моделі – наприклад, в Delphi, C++ або .NET. Відповідний елемент керування може бути доданий, скажемо, у програму, написану на Delphi – і Trident буде використовуватися для доступу до задалегідь заданого веб-сторінці, для читання або зміни інформації, що перебуває на ній. Події елемента керування будуть перехоплюватися й передаватися в основну програму.

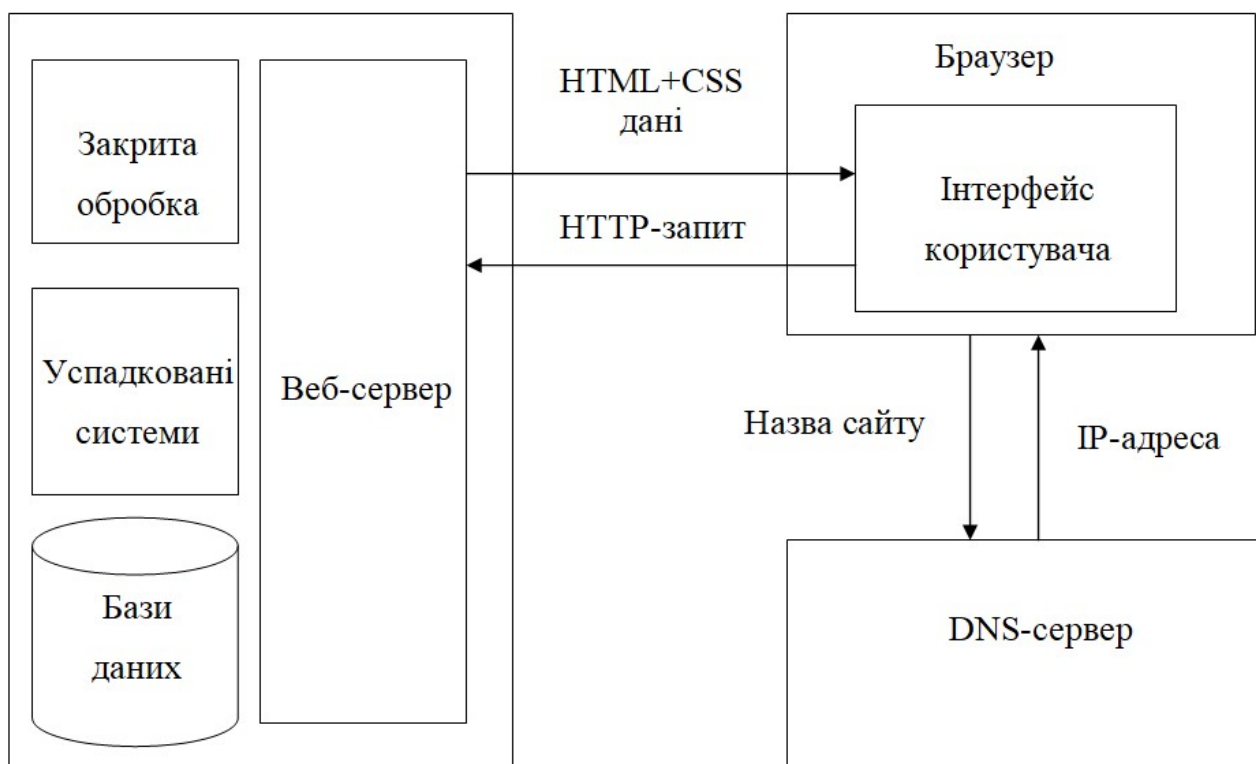


Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів браузеру для перегляду веб-сайтів. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем браузеру для перегляду веб-сайтів; Досліджена система браузеру для перегляду веб-сайтів; На основі отриманих результатів досліджень створена програмна реалізація системи браузеру для перегляду веб-сайтів. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання браузеру для перегляду веб-сайтів. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

- Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyzy, M., «QoE optimization technique for media delivery in 5G networks». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.
- Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.
- Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.
- Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.
- Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.
- Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.
- Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.
- Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». Підводні технології, 2024, № 13, с. 28-35.
- Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.
- Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». Проблеми інформатизації та управління, № 2(70). 2022. С. 28-37.
- Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98.
- Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
- Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.
- Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
- Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.
- О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

17. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.
18. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
19. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.
20. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.
21. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
22. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139
23. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.