**ПЕРЕЛІК СКОРОЧЕНЬ**

ПК – Персональний комп’ютер;

КПК – кишеньковий персональний комп’ютер;

ЕОМ – електронно-обчислювальна машина;

ПЕОМ – персональна електронно-обчислювальна машина;

ВДТ – відеодисплейний термінал;

ТК – Технічний коледж;

ТНТУ – Тернопільський Національний Технічний Університет;

ОС – операційна система;

ОО

НДР – науково-дослідна робота;

МОЗ – Міністерство Охорони Здоров’я;

HTML – HyperText Markup Language;

SGML – Standard General Markup Language;

XHTML – Extensible Hypertext Markup Language;

XML – Extensible Markup Language;

PHP – HyperText Preprocessor;

CSS – Cascading Style Sheet;

ANSI – American National Standards Institute;

FTP – File Transfer Protocol;

RTF – Rich Text Format;

URL – Uniform Resource Locator;

ASCII – American Standard Code for Information Interchange;

WWW – World Wide Web;

W3C – World Wide Web Consortium.

**ВСТУП**

В даний час бурхливого розвитку комп'ютерних технологій усе більш актуальним завданням є розробка і використання в навчальному процесі електронних навчальних систем, що розробляються з застосуванням гіпертекстових і мультимедійних технологій. Такі системи називаються інтерактивними навчальними веб-матеріалами та можуть використовуватися не тільки для денної, заочної та самостійної форм навчання, але і знайти широке застосування під час самостійної роботи студентів.

Мета виконання дипломного проекту полягає у проведенні досліджень та створенні інформаційно-документальної бази даних для розробки електронного посібника з дисципліни «Комп’ютерна логіка».

Актуальність проведення досліджень по створенню інформаційно-документальної бази електронного посібника з дисципліни «Комп’ютерна логіка» обумовлена необхідністю впровадження сучасних мультимедійних електронних посібників з метою удосконалення навчального процесу для різних форм навчання.

Завдання дипломного проекту – створення електронного посібника з дисципліни «Комп’ютерна логіка», як універсальної методичної розробки для студентів спеціальності 5.091504 «Обслуговування комп’ютерних та інтелектуальних систем та мереж», яка може використовуватися при вивченні дисципліни «Комп’ютерна логіка» в фахових навчальних курсах з комп’ютерного забезпечення.

Об’єктом дослідження є навчально-методичний комплекс спеціальності «Обслуговування комп’ютерних та інтелектуальних систем та мереж». Під час дипломного дослідження був здійсненний аналіз існуючих електронних та мультимедійних розробок, які використовуються в освітній діяльності та можливості застосування сучасних інформаційних технологій при створенні інформаційно-документальної бази для спеціалізованого електронного посібника.

На сучасному етапі розвитку інформаційних технологій все більш актуальним стає використання мультимедійних та Інтернет-технологій, які усе більше проникають в різні сфери нашого життя. Більшість користувачів персональних комп’ютерів починають освоєння мультимедіа з ігрових програм, але цією областю не вичерпуються численні можливості його використання.

Серед основних областей застосування засобів мультимедіа виділяють: ділову сферу; сферу розваги й освіти. В даний час значна частина освітніх, розважальних, інформаційно-довідкових програм на споживчому ринку відноситься до категорії мультимедіа.

В дусі мультимедіа створюється і малотиражна продукція рекламно-інформаційного характеру – каталоги, електронні посібники, довідники, презентації. Приклади використання мультимедіа можна найти в мультимедіа-енциклопедіях «Eyewitness History of the World», «Cinemania» «Encarta», «Большая энциклопедия Кирилла и Мефодия» та багатьох інших.  
Енциклопедії, альманахи, збори довідників, діалогові ігри, освітні програми і кінофільми із супровідним сценарієм, біографіями акторів, примітками режисера й оглядами роблять мультимедіа однієї з найбільш цікавих і творчих областей комп'ютерного світу.

Однак, використання мультимедіа, і персональних комп'ютерів узагалі, не обмежується сферою ділового застосування. Комп'ютер усе ширше входить й у сферу розваг і персональної освіти. Створюються продукти мультимедіа, розраховані на широке коло споживачів. Споживчою сферою застосування мультимедіа є інтерактивні CD-ROM, інтерактивні ігри та он-лайнові послуги.

Розвиток глобальної комп'ютерної мережі показав перспективність і необхідність віддаленого навчання студентів за допомогою інтерактивних електронних посібників, встановлюваних на серверах, підключених до локальної комп'ютерної мережі чи мережі Інтернет. Більш ширше розповсюдження таких інформаційних технологій в освіті дозволяє не тільки підвищити інтенсивність і ефективність процесу навчання, але й істотно розширити аудиторію потенційних слухачів провідних навчальних закладів країни.

Електронний посібник є однією з нових інформаційних технологій, які використовують в навчальному процесі. Тому дослідження та розробка електронного посібника з дисципліни «Комп’ютерна логіка», в даному дипломному проекті, дозволить удосконалити якість навчання студентів ТК ТНТУ ім. Івана Пулюя та дозволить знаходити необхідну їм інформацію серед величезної кількості інформації, яка вивчається студентами комп’ютерних спеціальностей протягом всього навчального процесу.

Даний дипломний проект містить вступ, 3 основних розділи, 2 додаткових розділи висновки та перелік посилань.

У першому розділі розглядається характеристика сучасних мультимедійних та електронних продуктів, зокрема, електронних посібників, які використовуються в навчальних закладах, у другому розділі формулюються вимоги до розробки електронних посібників та розглядаються інструментальні засоби їх створення, у третьому розділі наведено опис інформаційно-документального забезпечення електронного посібника, а також наведені деякі екранні форми та довідкові матеріали для електронного посібника.

**1 ЗАГАЛЬНИЙ РОЗДІЛ**

* 1. **Аналітичний огляд існуючих рішень**

Електронні посібники створюються з використання гіпертекстових технологій та технологій мультимедіа. Дамо стислу характеристику цих технологій.

**Від SGML до HTML та XHTML**

Стандартна узагальнена мова розмітки SGML — нащадок розробленої в 1960 році в IBM мови GML (від англ. Generalized Markup Language).

Спочатку SGML була розроблена для можливості спільного використання документів, що мають читатися машинами, у великих урядових та аерокосмічних проектах. Також вона широко використовується в печатній та видавничій сферах, але складність документа ускладнила його широке розповсюдження для повсякденного використання.

SGML надає множину варіантів синтаксичної розмітки для використання різними прикладними програмами. Змінюючи декларацію SGML можна навіть відмовитись від використання кутових дужок, хоча, цей синтаксис вважається стандартним.

В 1980 році фізик Тім Бернерс-Лі, який був підрядником у ЦЕРНі, запропонував і розробив прототип ENQUIRE, систему для вчених та дослідників ЦЕРНу для використання документів та обміну даними. В кінці 1990 року Бернерс-Лі узагальнив та написав програмне забезпечення HTML для браузерів та серверів.

Першою публічною згадкою про HTML був документ «HTML Tags» вперше викладений в Інтернеті Бернерсом-Лі наприкінці 1991 року. В ньому описувалось 20 елементів включаючи початковий, відносно просте оформлення. За винятком гіперпосилання будова HTML була дуже схожа з SGML. Тринадцять з тих елементів до сих пір використовуються в HTML 4.

HTML – це мова розмітки, яку браузери використовують для відображення та опису тексту, зображень та даних іншого типу. Базова характеристика роботи з HTML описана в самому браузері, але також може бути змінена чи розширена за допомогою CSS – каскадних таблиць стилів.

Документи HTML складаються повністю з елементів HTML, що зазвичай складаються з трьох елементів: відкриваючого та закриваючого тега, атрибутів всередині відкриваючого тега а також текстової чи графічної інформації між відкриваючим та закриваючим тегом, нерідко інших елементів HTML. Елементом HTML вважається все, що знаходиться всередині тега, включаючи сам тег. Кожен тег поміщується в трикутні дужки.

HTML підтримує декілька типів даних, наприклад: сценарії та стилі, а також безліч типів даних для атрибутів елементів, таких як: унікальний номер, ім’я, посилання, цифри, мови, кольори, кодування дата та час та ін.

Розширювана мова розмітки гіпертексту XHTML  — мова розмітки, що має таку саму виразну силу як і HTML але відповідає синтаксичним правилам XML.

XHTML 1.0 є «реформулюванням трьох типів документів стандарту HTML 4 засобами XML 1.0». W3C також продовжує підтримку Рекомендації HTML 4.01 та активну роботу над специфікаціями стандартів [HTML 5](http://uk.wikipedia.org/wiki/HTML_5) і XHTML 5. В поточному документі Рекомендацій XHTML 1.0, який було опубліковано та переглянуто до серпня 2002 року, W3C зазначив, що, «Сімейство XHTML є наступним кроком в еволюції Інтернету». Шляхом переходу на XHTML сьогодні, розробники контенту можуть увійти в світ XML з усіма його перевагами, залишаючись впевненими в зворотній та майбутній сумісності їхнього контенту.

Проте в 2004 році, незалежно від W3C було створено [Робочу групу з web-додатків та гіпертекстових технологій](http://uk.wikipedia.org/w/index.php?title=%D0%A0%D0%BE%D0%B1%D0%BE%D1%87%D1%83_%D0%B3%D1%80%D1%83%D0%BF%D1%83_%D0%B7_%D0%B2%D0%B5%D0%B1-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BA%D1%96%D0%B2_%D1%82%D0%B0_%D0%B3%D1%96%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B8%D1%85_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D0%B9&action=edit&redlink=1) ([WHATWG](http://uk.wikipedia.org/w/index.php?title=WHATWG&action=edit&redlink=1)), для роботи по вдосконаленню звичайного HTML не заснованого на XHTML. Більшість великих виробників [браузерів](http://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) не бажали реалізовувати функції з нових проектів стандартів W3C XHTML оскільки вважали, що вони не відповідають сучасним потребам розвитку Інтернету, а W3C захопився [формалізмом](http://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%96%D0%B7%D0%BC) XML і не реагував на реальні вимоги виробників. Apple, Mozilla та Opera сформували робочу групу [WHATWG](http://uk.wikipedia.org/w/index.php?title=WHATWG&action=edit&redlink=1), яка почала працювати над стандартом [HTML 5](http://uk.wikipedia.org/wiki/HTML5), який допускав, але не вимагав застосування XML. У 2007 році, Робоча група W3C HTML проголосувала за офіційне визнання HTML 5 і роботу над ним як наступне покоління стандарту HTML.

Завдяки динамічному HTML перед web-дизайнерами відкриваються нові можливості для підготовки змісту, щоб передати його в Internet. Якщо до HTML 3.2 web-сторінки були статистичними, то за новим стандартом з'явилися необмежені можливості у оформленні документів. Завдяки Java-аплетам, Java- та VB-сценаріям, а також Stylesheets, web-сторінки стали не тільки візуально привабливими, але й інтерактивними. Так, наприклад, HTML-документ може безпосередньо реагувати на команди користувача і самостійно оновлюватися.

Щоправда, використання динамічного HTML в web-сторінках потребує принципового планування і чіткої концепції. Наприклад, web-автор повинен заздалегідь визначитися, в якій послідовності елементи сторінки будуть представлені її відвідувачу. Додатково повинен бути розроблений план діалогу, який вкаже користувачу на різні інтерактивні ефекти (анімацію, випливаючі логотипи чи прихований зміст).

Крім того, web-автор повинен знати, що з використанням динамічного HTML, його потенційна клієнтура може зменшитися. Тому, що не кожен мандрівник мережі Інтернет зможе побачити в своєму web-браузері результати цієї технології. Для цього йому потрібен або Netscape Communicator, або Internet Explorer, і вони повинні бути не нижче 4-ої версії. Використання старішої версії призведе до помилки чи зміщення символів на екрані монітора.

Для запобігання цього на практиці, рекомендується розробити паралельно декілька версій, орієнтованих на конкретний web-браузер. Як правило, буває недостатньо простої версії, розробленої у відповідності зі старим стандартом 3.2.

**HTML-редактори**

Можна створювати найскладніший HTML-документ, користуючись лише найпростішим текстовим редактором. Проте застосування засобів, спеціально призначених для цієї цілі, дозволить заощадити чимало часу і зусиль. Для створення нових web-документів найбільш ефективними є HTML-редактори. У зв'язку з поширенням Internet в оформленні web-документів зростає кількість дизайнерів, які не хочуть навчатися мові HTML. Тому і створені були такі HTML-редактори, у яких web-дизайнеру більше не треба працювати з HTML-кодами. Цілком достатньо вибрати відповідну команду меню.

Практично для всіх платформ створені окремі незалежні програми, а також програмні надбудови до поширених додатків для роботи з HTML. В основному це умовно-безкоштовне програмне забезпечення. У наступній частині будуть описані найбільше поширені HTML-редактори. Це допоможе зробити обґрунтований вибір того або іншого редактора.

Для створення будь-якого HTML-документа достатньо простого текстового редактора, проте більшість розробників вважають, що краще використовувати спеціалізований HTML-редактор, який забезпечує контроль помилок і виконує інші корисні функції. Стандартні HTML-редактори є самостійними прикладними програмами, призначеними для підготовки HTML-документів, що не потребують додаткових програмних засобів. Існує два основних типи редакторів: текстові та графічні.

При роботі з текстовими редакторами HTML-код виводиться у вигляді звичайного тексту, а HTML-дескриптори виділяються іншим кольором.  
Графічні редактори дозволяють створити макет сторінки за допомогою миші. При цьому на екрані буде чудовий результат і такий, яким побачать створену сторінку відвідувачі.

У багатьох редакторах використовуються два типи інтерфейсу, між якими можна легко переключатися (яскравим прикладом служить HotDog, FrontPage98).

Графічні редактори забезпечують більш естетичний підхід до створення документів, дозволяючи зосередитися на їхньому макетуванні і форматуванні.  
Редактори HTML, що поставляються у вигляді надбудов, є додатковими програмами, що функціонують у якості розширень існуючих додатків (як правило, текстових процесорів). У будь-якому випадку використання надбудов має як сильні, так і слабкі сторони.

Коли стандарт HTML уперше став використовуватися в Internet, існували тільки HTML-редактори, які підтримували лише стандартні HTML-коди (їх називали тестовими або ASCII-редакторами). Текстові редактори і в даний час залишаються самими популярними інструментами створення web-документів. Як правило, такі редактори можуть оперувати тільки деяким набором регулярних кодів. Список цих кодів для більшості редакторів приведений у меню дескрипторів. Як приклади текстових редакторів можна навести Блокнот, Notepad++ і WordPad (Windows), vi та paco (UNIX) і SimpleText(Macintosh).

Одна із причин, чому розробники віддають перевагу текстовим редакторам, полягає в тому, що за їх допомогою набагато простіше вносити в документ будь-які зміни, що доводиться робити практично постійно. Ті, хто використовують текстові редактори, дозволяють набагато легше маніпулювати текстом документа і забезпечують більш гнучкий і творчий підхід при розміщенні в документ програмного тексту.

**Розширювана мова розмітки XML**

Розширювана мова XML – запропонований консорціумом World Wide Web (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними додатками, зокрема, через Інтернет. Є спрощеною підмножиною мови розмітки SGML. XML документ складається із текстових знаків, і придатний до читання людиною.

Стандарт XML визначає набір базових лексичних та синтаксичних правил для побудови мови описання інформації шляхом застосування простих тегів. Цей формат достатньо гнучкий для того, аби бути придатним для застосування в різних галузях. Іншими словами, запропонований стандарт визначає метамову, на основі якої, шляхом запровадження обмежень на структуру та зміст документів визначаються специфічні, предметно-орієнтовані мови розмітки даних.

Залишивши назви, дозволену ієрархію, та значення елементів і атрибутів відкритою та можливою бути визначеною в спеціалізованих схемах або визначеннях типу документа (DTD), XML утворює синтаксичну основу для створення спеціалізованих, основаних на XML мовах розмітки даних. Загальний синтаксис таких документів стабільний і наперед визначений – документи мають відповідати базовим вимогам XML, гарантуючи те, що довільне програмне забезпечення з підтримкою XML буде здатне щонайменше зчитати і відтворити відносну структуру інформації що міститься в них. Схема лише доповнює синтаксичні правила множиною обмежень. Зазвичай, схеми обмежують назви елементів та атрибутів, дозволені типи значень і допустиму ієрархію елементів, наприклад, дозволяючи лише елементу з назвою «народження» містити під-елемент з назвою «місяць» та з назвою «день», і кожен із них мусить містити лише літери. Обмеження, вказані в схемі, можуть також включати присвоєння певних типів даних для впливу на те, як обробляється інформація; наприклад, дані елемента «місяць» можна визначити як такі, що містять лише місяць, як це визначено відповідно до використаної мови схем.

**Гіпертекстовий препроцесор PHP**

PHP – скриптова мова програмування, що була створена для генерації HTML-сторінок на стороні web-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері web-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP – проект відкритого програмного забезпечення.

PHP інтерпретується web-сервером в HTML-код, який передається на сторону клієнта.

На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий HTML-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Тим не менш це не забороняє використовувати РНР для генерування JavaScript-кодів які виконаються вже на стороні клієнта.

Історія PHP починається з 1995 року, коли Расмус Лердорф створив простий застосунок мовою Perl, що аналізував відвідування користувачами його резюме на web-сайті. Потім, коли цим застосунком вже користувалися кілька чоловік, а число охочих одержати його постійно збільшувалося, Лердорф назвав своє творіння Personal Home Page Tools версія 1 і виставив для вільного завантаження. З цієї миті почався небувалий зліт популярності PHP.

Як це завжди буває, терміново було потрібно доопрацювання і нові доповнення. Для їхньої реалізації Расмус створює нову версію пакету, тепер уже написану С. Отриманий таким чином інструмент набуває робочої назви PHP/FI (Personal Home Page / Forms Interpreter), надалі він також буде відомий під назвою PHP 2. Ця версія вже більшою мірою схожа на сьогоднішній PHP. Вона мала синтаксис і спосіб іменування змінних в стилі мови Perl, можливість вбудовування PHP операторів в HTML-код сторінки, автоматичну інтерпретацію форм, інтеграцію з базами даних. При цьому все працювало досить швидко, оскільки PHP приєднувалось до web-серверу Apache. До 1997 року PHP використовувався вже на 50,000 доменах (не більше 1% всіх web-серверів).

У тому ж 1997 році до проекту PHP підключилися Зеєв Сураський і Енді Гутманс. Ці студенти Техніону, одного з ізраїльських університетів, намагалися використовувати PHP/FI для одного з комерційних університетських проектів. При цьому їм довелося зіткнутися з багатьма труднощами і обмеженнями цієї технології. Вивчаючи початковий код PHP 2, Зеєв і Енді дійшли висновку про необхідність доопрацювання, а точніше істотної переробки PHP, особливо в плані синтаксису мови. Протягом декількох місяців вони блискуче впоралися з цим завданням.

Закінчивши роботу Зеєв і Енді домовились з Расмусом про співпрацю в галузі розвитку та вдосконалення мови. З цієї миті з'являється PHP Group – група однодумців, що працюють над розвитком технології PHP. Одержаний продукт з'явився на світ у 1998 році під назвою PHP 3.

При цьому головною особливістю PHP 3 була можливість розширення ядра, що привернуло до роботи над PHP безліч сторонніх розробників, що створюють спеціалізовані модулі. Їх наявність дала PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. До кінця 1998 кількість користувачів PHP перевалила за сто тисяч, а PHP був вже встановлений на не менше ніж 10% серверах Інтернету. У той ж час значному поширенню даної мови сприяли публікації в електронній пресі та вихід книжок для вивчення PHP.

Відразу ж після виходу PHP 3, Енді Гутманс і Зеєв Сураський почали переробку ядра PHP. В першу чергу належало вирішити проблему підвищення продуктивності. Новий продукт, названий Zend Engine, успішно справлявся з поставленим завданням і був реалізований в 1999 році. Основною реалізованою ідеєю була можливість компіляції сценарію у виконуваний модуль, за рахунок чого продуктивність можна було підняти на порядок.

PHP 4, що працює на цьому ядрі, вийшов в 2000 році. На додаток до збільшення продуктивності, PHP 4 мав нові можливості щодо підтримки сесій, буферизацію виводу, безпечні способи обробки інформації, що вводиться користувачем, і нові мовні конструкції. З виходом 4 версії PHP став використовуватися вже на більш ніж 20% доменів Інтернету.

За час з 2000 по 2004 рік продовжувалися активні роботи з покращення 4 версії, але майже відразу PHP Group приступила до продумування можливостей нової версії. В першу чергу було вирішено підсилити об'єктні можливості мови, що дозволяло використовувати його для реалізації масштабних проектів. Роботи по створенню 5-тої версії велися тривалий час, в них брало участь рекордна кількість фахівців, зокрема Стерлінг Хьюз і Маркус Бергера.

У липні 2004 року виходить офіційний реліз PHP 5. В першу чергу, як і планувалося, було перероблено весь механізм роботи з об'єктами. І якщо в попередніх версіях об'єктно-орієнтоване програмування на PHP було можливе в мінімальному ступені, а тому і використовувалося на практиці не часто, то PHP 5 володіє прекрасним потенціалом реалізації об'єктного програмування. Окрім цього, PHP збагатився рядом цінних розширень для роботи з XML, різними джерелами даних, генерації графіки і інше.

Серед інших украй корисних доповнень в PHP 5 слід зазначити нову схему обробки виключень. Конструкція try/catch/throw дозволяє весь код обробки помилок локалізувати в одному місці сценарію.

Всі основні бібліотеки для роботи з XML, запозичені в PHP 4, були піддані серйозній переробці. Такі популярні розширення, як SAX, DOM і XSLT, тепер використовують інструмент libxml2, що робить їх ще ефективнішими.

У PHP 5 також включені два нові модулі для роботи з протоколами — SimpleXML і SOAP. SimpleXML дозволяє значно спростити роботу з XML-даними, представляючи вміст XML-документа у вигляді PHP-об'єкта. Розширення SOAP дозволяє будувати на PHP сценарії, що обмінюються інформацією з іншими застосунками за допомогою XML-повідомлень поверх існуючих web-протоколів, наприклад HTTP. Модуль для роботи з SOAP для PHP 5 надає розробникам засіб для достатньо швидкого створення ефективних SOAP-клієнтів і SOAP-серверів.

Новий модуль PHP 5 MySQLi (MySQL Improved) призначений для роботи з MySQL-сервером версій 4.1.2 та вищих, реалізовуючи не тільки процедурний, але і об'єктно-орієнтований інтерфейс до MySQL. Додаткові можливості цього модуля включають – SSL, контроль транзакцій, підтримка реплікації та ін.

**Каскадні таблиці стилів CSS**

Каскадні таблиці стилів – спеціальна мова, що використовується для відображення сторінок, написаних мовами розмітки даних. Найбільш часто CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої павутини.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже існуючи функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці web-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їх візуальної презентації.

CSS використовується авторами та відвідувачами web-сторінок для того щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори та ін. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою, КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерах та ін.).

Один і той же HTML або XML документ може бути відображений по-різному в залежності від використаного CSS. Стилі для відображення сторінки можуть бути:

* стилі автора (інформація надана автором сторінки):
* зовнішні таблиці стилів частіше за все записані в один чи декілька окремих файлів з розширенням \*.css;
* внутрішні таблиці стилів, включені як частина документу;
* стилі для окремого елементу;
* стилі користувача;
* локальний \*.css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад Opera);
* стилі браузера;
* стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, те, в якій послідовності і для яких елементів застосовуються стилі. Таким чином використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена більш загальними стилями.

Звичайно використання CSS стилів має свої переваги та недоліки. До переваг можна віднести наступні якості:

* Інформація про стиль для цілого сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;
* Різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;
* Сторінки зменшуються в об'ємі та стають більш структурованими, за рахунок того що інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з їх урахуванням.

Прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

**Java та сценарії JavaScript**

Java– об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Синтаксис мови багато в чому походить від C та C++. У офіційній реалізації, Java програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Sun Microsystems надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель С++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в С/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім, Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

Java вплинула на розвиток J++, що розроблялась компанією Microsoft. Роботу над J++ було зупинено через судовий позов компанії Sun Microsystems, оскільки ця мова програмування була модифікацією Java. Пізніше в новій платформі Microsoft.NET випустило J#, щоб полегшити міграцію програмістів J++ або Java на нову платформу. З часом нова мова програмування С#, стала основною мовою платформи, перейнявши багато чого з Java. J# востаннє включався в версію Microsoft Visual Studio 2005. Мова сценаріїв JavaScript має схожу із Java назву і синтаксис, але не пов'язана із Java.

Розробку проекту започаткував Джеймс Ґослінґ, сам проект мав назву «Green» (Зелений). Створення першої робочої версії, яка мала назву «Oak» зайняло 18 місяців. Оскільки виявилось, що ім'я Oak уже використовувалось іншою фірмою, то в результаті тривалих суперечок навколо назви нової мови з поміж ряду запропонованих було вибрано назву Java, у 1995р. мову було офіційно перейменовано.

Головним мотивом створення Java була потреба в мові програмування, яка б не залежала від платформи (тобто від архітектури) і яку можна було б використовувати для створення програмного забезпечення, яке вбудовується в різноманітні побутові електронні прилади, такі як мобільні засоби зв'язку, пристрої дистанційного керування тощо.

Досить скоро майже всі найпопулярніші тогочасні браузери отримали можливість запускати «безпечні» для системи Java аплети всередині web-сторінок. У грудні 1998 р. Sun Microsystems випустила Java 2 (спершу під назвою J2SE 1.2), де було реалізовано декілька конфігурацій для різних типів платформ. Наприклад, J2EE призначалася для створення корпоративних застосунків, а значно урізана J2ME для приладів з обмеженими ресурсами, таких як мобільні телефони. У 2006 році в маркетингових цілях, Версії J2 було перейменовано у Java EE, Java ME та Java SE, відповідно.

13 листопада 2006 року Sun випустили більшу частину Java в якості вільного та відкритого програмного забезпечення згідно з умовами GNU General Public License (GPL). 8 травня 2007 корпорація закінчила процес, в результаті якого всі початкові коди Java були випущенні під GPL, за винятком невеликої частини коду, на який Sun не мала авторського права.

Період становлення Java збігся у часі з розквітом міжнародної інформаційної служби World Wide Web. Ця обставина відіграла вирішальну роль у майбутньому Java, оскільки Web теж вимагала платформо-незалежних програм. Як наслідок, були зміщені акценти в розробці Sun з побутової електроніки на програмування для Інтернет.

JavaScript– назва реалізації стандарту мови програмування ECMA-Script компанії Netscape, базується на принципах прототипного програмування. Найпоширеніше і найвідоміше застосування мови – написання сценаріїв для web-сторінок, але, також, використовується для впровадження сценаріїв керування об'єктами вбудованими в інші програми.

Незважаючи на схожість назв, JavaScript та мова програмування Java мають дуже мало спільного. Права на назву JavaScript належать компанії Sun Microsystems. Розроблена в компанії Netscape, на даний час підтримується більшістю браузерів. Текст програми включається безпосередньо в HTML-документ і інтерпретується самим браузером. Застосовується в основному для часткової автоматизації обробки і маніпуляції даними, які використовує сторінка.

В 1992 році компанія Nombas розпочала розробку вбудованої скриптової мови Cmm (Сі-мінус-мінус), яка, по замислу розробників, повинна була стати достатньо потужною, щоб замінити макроси, зберігаючи при цьому схожість з Сі, щоб програмістам було легко оволодіти нею. Головною відмінністю від Сі була робота з пам’яттю. В новій мові все управління пам’яттю здійснювалось автоматично: не було необхідності створювати буфери, оголошувати змінні, здійснювати перетворення типів. В іншому мови були доволі схожі між собою: зокрема, Cmm підтримував стандартні функції і оператори Сі. Cmm була перейменована в ScriptEase, оскільки початкова назва звучала надто негативно, а згадка про Сі «відлякувало» людей.

На основі цієї мови було створено CEnvi. В кінці листопада 1995 року Nombas розробили версію CEnvi, впроваджувану в web-сторінки. Сторінки, які можна було міняти за допомогою скриптової мови, отримали назву Espresso Pages – вони демонстрували використання скриптової мови для створення гри, перевірки користувацького вводу в форми і створення анімації.

Станом на 2006 рік, останній стандарт мови – JavaScript 1.7. Якщо спрощено, то ECMA-262 є стандартизованим варіантом мови JavaScript. Стандартом ECMA-357 визначено E4X, розширення мови для роботи з XML.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, – функції як об'єкти першого рівня, об'єкти як списки, каррінг – анонімні функції, замикання – що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою Сі має такі корінні відмінності:

* об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;
* функції як об'єкти першого класу;
* обробка винятків;
* автоматичне приведення типів;
* автоматичне прибирання сміття;
* анонімні функції.

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто в кінцевому рахунку успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, приміром, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може стояти в тексті програми після неї.

Основні архітектурні особливості JavaScript: динамічна типізація, автоматичне керування пам’яттю, прототипне програмування, функції як об’єкти першого класу.

На JavaScript вплинуло багато мов, під час розробки основною ціллю було зробити мову схожою на Java, але при цьому легкою для використання не професійними програмістами. Мовою JavaScript не володіє будь-яка компанія чи організація, що відрізняє її від ряду мов програмування, що використовуються в web-розробках.

**Від ASP до ASP.NET**

ASP (англ. *Active Server Pages*, укр. *активні серверні сторінки*) — це технологія від компанії Microsoft, що дозволяє динамічно формувати автоматично оновлювані веб-сторінкиз боку веб-сервера. Технологія подається у формі додатку до веб-серверу Internet Information Services (IIS).

Більшість сторінок, що створені за допомогою даної технології, написані мовою VBScript, але веб-майстер вільний використовувати будь-яку мову, за умови, що для неї існує та встановлений відповідний Active Scripting механізм. JScript, що є реалізацією ECMAScript від компанії Microsoft, — це зазвичай друга мова, котру використовують разом із технологією ASP. Існують також досить багато інших мов програмування, що постачаються сторонніми компаніями.

Технології InstantASP та ChiliASP дозволяють використовувати технологію ASP без операційної системи Microsoft Windows. Існують вельми багато спілок відкритого програмного коду, що створюють ASP скрипти, компоненти та програми, котрі можуть бути використані безкоштовно за умови дотримання певних ліцензійних угод.

ASP.NET — це технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft. NET і розвитком старішої технології Microsoft ASP. У цей час останньою версією цієї технології є ASP.NET 4.5.

ASP.NET зовні багато в чому зберігає схожість із старішою технологією ASP, що дозволяє розробникам відносно легко перейти на ASP.NET. У той же час внутрішній устрій ASP.NET істотно відрізняється від ASP, оскільки вона заснована на платформі. NET і, отже, використовує всі нові можливості, що надаються цією платформою.

Після випуску сервера Internet Information Services 4.0 в 1997 році, компанія Microsoft почала досліджувати можливість нової моделі веб-застосунків, яка задовольнить скарги на ASP, особливо пов'язані з відділенням оформлення від змісту, і яка дозволить писати «чистий» код. Робота з розробки такої моделі була доручена Марку Андерсу, менеджеру команди IIS, і Скотту Гутрі, що надійшов на роботу в Microsoft в 1997. Андерс і Гутрі розробили початковий проект протягом двох місяців, і Гутрі написав код первісного прототипу під час різдвяних канікул 1997 року.

Початковий проект називався «XSP»; Гутрі пояснив в інтерв'ю 2007 року що, «завжди запитують, що означає буква X. У той час вона нічого не значила. XML починається з неї; XSLT починається з неї. Все кльове починається з X, тому ми його так і назвали.» Прототип XSP був написаний на Java, але скоро було вирішено побудувати нову платформу на основі Common Language Runtime (CLR), бо на платформу Java у компанії Microsoft закінчувалась ліцензія. Гутрі описав це рішення як «величезний ризик», тому що успіх нової розробки був пов'язаний з успіхом CLR, яка, як і XSP, перебувала на ранній стадії розробки.

Переваги ASP.NET перед ASP:

* Компільований код виконується швидше, більшість помилок відловлюється ще на стадії розробки;
* Значно поліпшена обробка помилок часу виконання, з використанням блоків try .. catch;
* Користувальницькі елементи управління (controls) дозволяють виділяти часто використовувані шаблони, такі як меню сайту;
* Використання метафор, вже застосовуються в Windows-застосунках, наприклад, таких як елементи керування та події;
* Розширюваний набір елементів управління і бібліотек класів дозволяє швидше розробляти застосунки;
* ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на VB.NET, Delphi.NET, Visual C/C++ тощо;
* Можливість кешування всієї сторінки або її частини для збільшення продуктивності;
* Можливість кешування даних, що використовуються на сторінці;
* Можливість поділу візуальної частини та бізнес-логіки з різних файлів («code behind»);
* Розширювана модель обробки запитів;
* Розширена подієва модель;
* Розширювана модель серверних елементів керування;
* Наявність master-сторінок для завдання шаблонів оформлення сторінок;
* Підтримка CRUD-операцій при роботі з таблицями через GridView;
* Вбудована підтримка AJAX;
* ASP.NET має перевагу у швидкості в порівнянні з іншими технологіями, заснованими на скриптах.

Тут можна навести певні порівняння. Так, ASP — похідна від Win32, XML і HTML; PHP — від XML, HTML, Java і CDI, тоді ASP.NET — від HTML і .NET (XML і XAML відповідно). При цьому, якщо зазвичай Rich Media Application створюють за допомогою Flash, тепер це робиться за допомогою модуля Silverlight, так само через сам ASP.NET. ASP.NET — багатше середовище для розробки та розгортання веб-ресурсів. У ASP.NET можна працювати з будь-якою .NET мовою, аж до Managed C++ і Visual Basic, що дозволяє не замислюватися про перехід на C#.

**C Sharp**

C# (C Sharp) – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (при фірмі Microsoft).

Синтаксис C# близький до С++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато що від своїх попередників — мов С++, Delphi, Модула і Smalltalk — С#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++).

C# є дуже близьким родичем мови програмування Java. Мова Java була створена компанією Sun Microsystems, коли глобальний розвиток інтернету поставив задачу роззосереджених обчислень. Взявши за основу популярну мову C++, Java виключила з неї потенційно небезпечні речі (типу вказівників без контролю виходу за межі). Для роззосереджених обчислень була створена концепція віртуальної машини та машинно-незалежного байт-коду, свого роду посередника між вихідним текстом програм і апаратними інструкціями комп'ютера чи іншого інтелектуального пристрою.

Java набула чималої популярності, і була ліцензована також і компанією Microsoft. Але з плином часу Sun почала винуватити Microsoft, що та при створенні свого клону Java робить її сумісною виключно з платформою Windows, чим суперечить самій концепції машинно-незалежного середовища виконання і порушує ліцензійну угоду. Microsoft відмовилася піти назустріч вимогам Sun, і тому з'ясування стосунків набуло статусу судового процесу. Суд визнав позицію Sun справедливою, і зобов'язав Microsoft відмовитися від позаліцензійного використання Java.

У цій ситуації в Microsoft вирішили, користуючись своєю вагою на ринку, створити свій власний аналог Java, мови, в якій корпорація стане повновладним господарем. Ця новостворена мова отримала назву C#. Вона успадкувала від Java концепції віртуальної машини (середовище .NET), байт-коду (MSIL) і більшої безпеки вихідного коду програм, плюс врахувала досвід використання програм на Java.

Нововведенням C# стала можливість легшої взаємодії, порівняно з мовами-попередниками, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проектів. Якщо програми на різних мовах виконуються на платформі .NET, .NET бере на себе клопіт щодо сумісності програм (тобто типів даних, за кінцевим рахунком).

Станом на сьогодні C# визначено флагманською мовою корпорації Microsoft, бо вона найповніше використовує нові можливості .NET. Решта мов програмування, хоч і підтримуються, але визнані такими, що мають спадкові прогалини щодо використання .NET.

C# розроблявся як мова програмування прикладного рівня для CLR і, як такий, залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 значно збагатився і сам C#; подібної взаємодії слід чекати і надалі. (Проте ця закономірність буде порушена з виходом C# 3.0, що є розширеннями мови, що не спираються на розширення платформи .NET.) CLR надає C#, як і всім іншим .NET-орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, збірка сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# точно так, як і це робиться для програм на VB.NET, J# тощо.

На даний час існує безліч інструментальних засобів для розробки веб-сторінок. Розглянемо деякі з них.

**Microsoft Visual Studio**

Microsoft Visual Studio – серія продуктів фірми Microsoft, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Версії продукту:

* MS Visual Studio 97;
* MS Visual Studio 6.0;
* MS Visual Studio .NET;
* MS Visual Studio 2002;
* MS Visual Studio 2003;
* MS Visual Studio 2005;
* MS Visual Studio 2008;
* MS Visual Studio 2010;
* MS Visual Studio 2012;

Розглянемо детальніше 2 останні на даний момент версії — Visual Studio 2010 та Visual Studio 2012.

**Visual Studio 2010**

Представлений 12 квітня 2010 року. Включає .NET Framework 4.0. З'явилася нова мова F#, Visual C++ підтримує стандарт C++0x.

Інструменти Visual Studio 2010 допоможуть не тільки в створенні звичних програм для мобільних телефонів і персональних комп'ютерів, але в розробці хмарних застосунків. При цьому процес тестування, зневадження і розгортання програм в «хмарі» аналогічний створенню .NET-застосунків. Іншим важливим доповненням в Visual Studio 2010 є інструменти для багатонитевої розробки з використанням як некерованого коду, так і .NET Framework.

У Visual Studio 2010 повністю перероблений інтерфейс з використанням Windows Presentation Foundation (WPF), упроваджено наступне покоління інструментів ASP.NET, є підтримка динамічних розширень в мовах програмування C# і Visual Basic, використовуються нові шаблони проектів, інструментарій для документування тестових сценаріїв і велика кількість нових бібліотек, що підтримують Windows 7.

Visual Studio Ultimate 2010, формально це Visual Studio Team System 2010, кодове ім’я Rosario, це новий інструмент для спільної розробки застосунків.

**Visual Studio 2012**

Представлений 2 серпня 2012 року. Включає .NET Framework 4.5. Головні нововведення це підтримка Windows RunTime, C++/CX (Component Extensions), бібліотека C++ AMP для GPGPU програмування, компілятор Visual C++ майже підтримує стандарт C++11. З'явився новий тип проектів, котрі дозволяють писати рідні застосунки (у стилі Windows Metro) для операційної системи Windows 8.

**Internet Information Services**

IIS (Internet Information Services, до версії 5.1 — Internet Information Server) — це набір серверів для декількох службІнтернету від компанії Microsoft. IIS поширюється з операційними системами родини Windows NT.

Основний компонент IIS — веб-сервер, який дозволяє розміщувати в Інтернеті сайти. IIS підтримує протоколи HTTP, HTTPS, FTP, POP3, SMTP, NNTP. IIS другий за популярністю веб-сервер за кількістю сайтів, після Apache HTTP Server. За даними компанії Netcraft на 11.10.2007, понад 37.13% сайтів обслуговується веб-сервером IIS.

Перша версія IIS була випущена, як додатковий набір Інтернет сервісів для Windows NT 3.51. IIS 2.0 з'явився в операційній системі Windows NT 4.0. В IIS 3.0 вже була представлена Active Server Pages, технологія динамічного формування веб-сторінок. В IIS 4.0 відмовились від підтримки протоколу Gopher і розповсюджувався окремо на диску «Option Pack» CD-ROM for Windows NT.

Версії IIS, які підтримуються в цей час: 7.0 для Windows Vista, 6.0 для Windows Server 2003 та 5.1 для Windows XP Professional. Windows XP має обмежену версію IIS 5.1, яка підтримує тільки 10 одночасних підключень й один веб-сайт. В IIS 6.0 підтримує IPv6. FastCGI модуль також доступний для IIS5.1, IIS6 та IIS7.

Окремо варто виділити IIS 7.0. У Windows Vista IIS 7.0 не встановлюється за замовчуванням, але можна вибрати як додатковий компонент для встановлення. IIS 7.0 на Vista не обмежує кількість підключень, але є обмеження при одночасному виконнані декількох запитів.

Вперше з'явилась у Windows Vista а також включена до складу Windows Server 2008. IIS 7.0 має модульну архітектуру. На відміну від монолітних серверів, які надають всі свої сервіси, IIS 7 має ядро двигуна веб-сервера. Модулі зі специфічною функціональністю можуть бути додані до двигуна. Переваги такої архітектури в тому, що тільки необхідні функції можуть бути увімкнені, а функціональність може бути розширена за рахунок використання спеціальних модулів.

IIS 7.0 розповсюджується з невеликою кількістю модулів, але Microsoft обіцяє зробити інші модулі доступними онлайн. Наступний набір модулів розповсюджується з сервером:

* HTTP модулі;
* Модулі безпеки;
* Модулі контенту;
* Модулі стиснення;
* Модулі кешування;
* Модулі для протоколювання та діагностики;

Значна зміна в порівнянні з попередніми версіями IIS полягає в тому що вся конфігураційна інформація зберігається виключно в конфігураційних XML файлах, а не в метабазах. Сервер має глобальний кофігураційний файл, який містить налаштування за замовчуванням, і кожна коренева віртуальна веб директорія(так само як і її піддиректорії) можуть містити web.config файл, в якому глобальні налаштування розширюються або заміщуються. Внесені зміни в ці файли починають діяти відразу.

IIS 7.0 має повністю змінений адміністративний інтерфейс, який використовує такі переваги сучасної MMC, як панель задач та асинхронні операції. Налаштування ASP.NET ще більше інтегроване в адміністративний інтерфейс.

Для розробки електронного посібника з дисципліни «Комп’ютерна логіка» я використав: ASP.NET C#, HTML, каскадні таблиці стилів CSS, JavaScript, Java, SQL Server 2008 та IIS 7.0. Головним інструментом розробки був Notepad++ 5.9.2. Додатково використовував Adobe Photoshop CS5 та Paint.NET.

* 1. **Аналіз технічного завдання**

Завданням дипломного проекту є модернізація електронного посібника з дисципліни «Комп’ютерна логіка». Код посібника буде переписаний на мові ASP.NET C# з використанням HTML розмітки, CSS-стилів, Java та JavaScript. Розробка дизайну здійснюватиметься за допомогою Adobe Photoshop CS4, Paint,NET та матеріалів із мережі Інтернет. Електронний посібник складатиметься з домашньої сторінки, лекцій, лабораторних та практичних робіт, списків індивідувальних завдань, списків груп, тестів для перевірки набутих знань по темах, реєстрації та сторінки адміністрування.

Удосконалений посібник буде більш надійним, більш функцональним, а наявність бази даних дасть можливість не лише швидко та легко зберігати дані про користувачів (студентів, викладачів), матеріали посібника, а й надійнше зберігати та використовувати інформацію.

* + 1. **Найменування та область застосування**

Темою дипломного проекту є «Модернізація електронного посібника з дисципліни «Комп’ютерна логіка».

Метою даного дипломного проекту є вдосконалення електронного посібника з дисципліни «Комп’ютерна логіка», універсального посібника з викладеним матеріалом, тестами для перевірки знань, лабораторними та практичними роботами, індивідуальними завданнями. Даний електронний посібник повинен відповідати стандартам існуючих аналогічних електронних посібників в мережі Інтернет. Сторінки посібника повинні бути невеликими, для швидкого завантаження та містити лише необхідну для користувачів інформацію, бути приємним на вигляд та легким для навігації.

Об’єктом використання даної розробки є будь-який користувач ПК з доступом до мережі Інтернет.

* + 1. **Призначення розробки**

Електронний посібник призначений для оптимізації та спрощення процесу навчання, а також з метою впровадження новітніх технологій в процес навчання, підвищення доступності навчального матеріалу для студентів не лише ТК ТНТУ, ТНТУ але й інших навчальних закладів.

* + 1. **Вимоги до програмного забезпечення**

Перед створенням електронного посібника, необхідно встановити на нашому комп’ютері відповідне програмне забезпечення, яке буде необхідне нам у майбутньому, а саме:

* MS Visual Studio 2010 або вище;
* SQL Server 2005 або вище;
* SQL Server Management 2005 або вище;
* IIS 7.0 або вище;
* Navicat або SQLyog будь-яка версія;
* Google Chrome 26.0 або вище;
* Mozilla Firefox 19.0 або вище.
  + 1. **Вимоги до програмної документації**

Для розробки даного дипломного проекту викладач з дисципліни «Комп’ютерна логіка» надала матеріал для лекцій, тестів для самоперевірки та лабораторних і практичних робіт. На основі цього матеріалу було розроблено основне наповнення сайту.

Після закінчення розробки електронного посібника необхідно підготувати наступну документацію: проблеми, що можуть виникнути при завантаженні електронного посібника та їх усунення.

Паспорт електронного посібника із внесеними у нього наступними даними:

* назва електронного посібника та адреса в Інтернеті (якщо є);
* мінімальні вимоги до апаратного і програмного забезпечення;
* роздільна здатність моніторів, під який оптимізовано посібник;
* оптимальна роздільна здатність монітора для перегляду;
* основний браузер, за допомогою якого буде переглядатися електронний посібник;
* середній час завантаження сторінок електронного посібника;
* час завантаження головної (титульної) сторінки;
* об’єм електронного посібника (в Мб);
* число сторінок, файлів, гіперпосилань (зовнішніх та внутрішніх) електронного посібника.
  + 1. **Стадії та етапи розробки електронного посібника**

1. Розробка дипломного проекту відбуватиметься ступенево. Перш за все створюється макет майбутнього посібника: вигляд, фон, кольорова палітра, оптимальний шрифт, його розмір. Розробляється структура посібника.
2. Створення веб-проекту, структури папок та файлів, налаштування додаткових ресурсів.
3. Створення структури сторінок та взаємозв’язків між ними.
4. Написання HTML коду MasterPage та інтерфейсу решти сторінок.
5. Написання програмного коду MasterPage, сторінки реєстрації, входу в систему.
6. Налаштування доступу до бази даних SQL Server.
7. Написання програмного коду решти сторінок.
8. Збір посібника, розробка титульної сторінки та (за бажанням) розміщення його в Інтернеті. Весь графічний, анімаційний, звуковий та програмний матеріал компілюється в готовий продукт: мультимедійний ресурс посібника.
9. Тестування та доробка. Готовий електронний посібник тестується на предмет помилок веб-програмування і невідповідності технічному завданню.
   * 1. **Порядок контролю та прийому електронного посібника**

Порядок здачі-приймання готового електронного посібника принципово нічим не відрізняється від здачі-прийому звичайного веб-сайту. При чому, чим ретельніше обумовлені всі питання на етапі узгодження технічного завдання на створення електронного посібника або його ре-дизайн, тим менша ймовірність того, що при здачі готового сайту у замовника можуть виникнути які-небудь питання.

Як відомо, вдосконаленню немає меж. Робота над електронним посібником може тривати нескінченно довго. Будь-який веб-продукт можна постійно вдосконалювати, і в цьому немає нічого незвичайного.

При прийманні готового сайту замовники повинні найперше звернути увагу на такі параметри:

* обсяг текстової інформації в друкованих знаках з пробілами, кількість;
* обсяг робіт зі створення електронного посібника або зміни його дизайну, які визначаються відповідними договорами;
* внесення змін до договору здійснюється відповідно до законодавства.

Щоб максимально виключити можливість невідповідності готового сайту естетичним очікуванням замовника, останній зобов'язаний завчасно надати в розпорядження веб-дизайнера контент сайту (текст сайту, таблиці сайту і графіку сайту).

При цьому слід враховувати наступне: перед здачею проекту замовникові, виконується його повне тестування, що включає тестування роботи ЕП в різних браузерах, тестування посилань (перевірка на «биті посилання»), тестування сайту під навантаженням та інші перевірки.

**2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ**

* 1. **Типи електронних посібників та оцінка їх якостей**

Електронний посібник — версія навчального посібника в електронному (цифровому) вигляді. Такі посібники можна читати за допомогою комп'ютерів, мобільних телефонів чи спеціалізованих пристроїв.

В англомовних країнах для позначення цифрових версій [книг](http://uk.wikipedia.org/wiki/%D0%9A%D0%BD%D0%B8%D0%B3%D0%B0) та посібників, для зберігання яких використовують [відкриті](http://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%B8%D0%B9_%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82) формати, засновані на простих [текстових файлах](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B8%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), застосовують також термін «електронний текст». Електронними текстами називають також навчальні електронні посібники, до складу яких, окрім тексту, належать растрові зображення, приклади даних, вправи, спеціально написані коментарі і відповіді на можливі запитання.

Одним із технічних споживачів електронних книг та посібників є Електронна бібліотека, яка надає користувачу (в основному Інтернету) екземпляр даної книги у певному форматі.

Кожна інструментальна розробка має свої недоліки та переваги. Розглянемо переваги розробленого електронного посібника з дисципліни «Комп’ютерна логіка» та його недоліки, які потребують усунення або удосконалення.

Основною перевагою електронного посібника є широкий спектр його застосування.

До переваг відносяться такі його риси:

* зручність при роботі з матеріалом за рахунок можливості застосування розвитої системи гіперпосилань;
* можливість розміщення такого посібника в мережі Інтернет;
* наявність великої кількості програмних засобів для перегляду HTML-документів;
* наявність візуального інтерфейсу;
* можливість доповнення новими розділами та легко вносити зміни у вже існуючі;
* можливість використання при самостійному освоєнні матеріалу;
* простота використання;
* компактність;
* низькі системні вимоги (досить Windows XP);
* наявність гіперпосилань як на внутрішні так і на зовнішні ресурси;
* невеликий обсяг;
* можливість відображати анімаційні малюнки, мультимедійні кліпи і відтворювати аудіо-книжки та mp3-файли.

До недоліків можна віднести:

* екрани комп’ютерів втомлюють очі;
* висока вартість, еквівалентна 20-40 друкованим книгам;
* відсутність деяких мультимедійних компонентів – звуку, відео тощо;
* відсутність внутрішньої пошукової системи та довідки.

Вказані недоліки характерні для першої редакції оновленого електронного посібника, однак в подальшій розробці передбачається використання мультимедійних засобів та удосконалення.

Електронний посібник повинен суміщати в собі функції підручника, викладача, довідниково-інформаційного посібника, консультанта, тренажера а також засобу для самоперевірки по пройденому матеріалу.

Використання системного підходу до розробки електронних навчальних посібників дозволяє зробити серйозний крок на шляху переходу від пізнавальної до прагматичної моделі освіти і сприяє вирішенню проблем створення посібників нового покоління, що дають можливість:

* збільшити кількість користувачів;
* підвищити наочність представлення матеріалу;
* використовувати електронний посібник тривалий час;
* звести до мінімуму витрати на пошук і підбір літератури;
* здійснювати контроль отриманих знань.

Як інструментальний засіб доцільно використовувати електронний посібник, що задовольняє наступним вимогам:

* простота використання електронного посібника:
* вбудовані мультимедійні можливості;
* компактність;
* низькі системні вимоги (досить Windows 95/98/2000/XP/NT).

Для створення інтерактивних електронних посібників необхідні знання, крім знань за курсом, необхідно мати спеціальні навички в області застосування Інтернет-технологій.

При використанні для розробки навчальних матеріалів кожного з великої кількості наявних програмних засобів створення і редагування HTML-документів, наприклад, таких як Notepad, Notepad++, а краще Microsoft Visual Studio 2008 чи вище, програмні засоби для роботи з базами даних, такі як Navicat, SQLyog чи MS Server Managment вимагаються базові знання про гіпертекст, знання в обасті об’єктно-орієнтованого програмування на рівні спеціаліста, навики роботи з базами даних такими як: SQL Server чи MySQL.

Крім того, для надання веб-документам інтерактивних властивостей, що дозволяють оперативно реагувати на інформацію, введену користувачем, необхідні будуть ще більш глибокі знання Інтернет-технологій, пов'язані з необхідністю написання програм, для обробки отриманої інформації з використанням Java, JavaScript та базами даних SQL.

В цьому разі необхідна спільна робота колективу фахівців різного профілю. Ці обставини істотно підвищують трудомісткість створення навчальних веб-посібників і, деякою мірою, стримують процес розвитку такої форми освіти.

До сучасних інформаційних технологій, які використовують в навчальному процесі, відносять електронні та гібридні бібліотеки, електронні посібники, довідково-пошукові системи Інтернет та ін.

Як правило, у даний час бурхливого розвитку комп'ютерних технологій електронні навчальні чи довідково-пошукові системи розробляються з використанням гіпертекстових і мультимедійних технологій.

Такі системи називають інтерактивними навчальними веб-матеріалами, та можуть знайти і знаходять широке застосування як в звичайних формах навчання, так і в достатньо новій формі навчання, як самоосвіта та під час самостійної роботи студентів.

Тому створення навчальних веб-матеріалів пред’являє дуже високі вимоги до кваліфікації розробника в області практичного використання сучасних інформаційних технологій, тому що, окрім знань предмета, необхідно мати спеціальні навички в області застосування Інтернет-технологій, чи вдаватися до допомоги відповідного фахівця чи навіть спільної роботи колективу фахівців різного профілю.

Ці обставини істотно підвищують трудомісткість створення навчальних web-посібників і, деякою мірою, стримують процес розвитку такої форми освіти, як застосування навчальних web-матеріалів.

Електронний посібник– це універсальний методичний посібник, який містить широке коло питань різних навчальних дисциплін, викладених в стислій формі та призначений для використання в процесі навчання.

Аналіз літературних джерел показав, що більшість перших електронних навчальних видань являли собою електронні копії друкованих видань і, здебільшого, не враховували комп’ютерних можливостей подачі матеріалу.

Однак останнім часом значна увага приділяється розробці комп’ютерних посібників, роботу над якими можна розглядати як спробу зробити серйозний крок до вирішення проблеми створення посібників нового покоління.

Для розв’язання цієї проблеми запропоновано використовувати системний підхід при створенні електронних посібників. Системний підхід розглядає об'єкт як систему, що складається з безлічі взаємозалежних елементів, що утворять певну цілісність системних властивостей.

Системний підхід дозволяє не орієнтувати навчальний посібник на конкретну групу користувачів, але створювати його таким чином, щоб ним міг скористатися практично кожен. У залежності від потреб і вже наявних знань, користувач сам обирає матеріал для вивчення, його обсяг, технологію навчання.

Створення різних моделей представлення знань, що в одному випадку представляють об’єкти, характерні для логічного мислення, а в іншому – образи-картини, з якими оперує образне мислення, дає можливість оптимізувати процес навчання. Включення системи тестів на початковому етапі роботи з навчальним посібником дозволяє ідентифікувати особистісні якості студента, а потім оптимізувати налаштування на нього і рекомендувати відповідну методику навчання.

Контроль знань після вивчення кожного розділу може здійснюватися різними способами (за допомогою тестів, контрольних питань і т. д.).  
У залежності від того, наскільки користувач засвоїв матеріал, можна при необхідності повторити вивчений розділ, чи відкоригувати, змінити методику навчання. При низьких результатах можливе з'ясування думки самого користувача про причини поганого засвоєння матеріалу.

Застосування системного підходу при створенні електронного посібника дозволяє розширити область застосування електронних посібників і коло потенційних користувачів. За рахунок включення в посібник блоку нових розробок і проблемно-орієнтованих баз даних, його можна використовувати більш тривалий час і тим самим підвищити його якість, довговічність і точність даних наведених у ньому.

Використання мультимедіа, аудіо та відео-компонентів підвищує наочність представлення матеріалу, а також дає можливість використовувати його людям, що мають різні патології (порушення слуху, зору і т.п.). За рахунок цього можливо різке збільшення кількості користувачів і ефективності використання електронного посібника. Включення перерахованих компонентів в електронний посібник дозволяє перейти від пізнавальної моделі освіти до прагматичної, у якій студент стає активним об'єктом освіти.

Таким чином, використання системного підходу до розробки електронних навчальних посібників дозволяє зробити серйозний крок на шляху переходу від пізнавальної до прагматичної моделі освіти і сприяє рішенню проблем створення посібників нового покоління, що дають можливість збільшити кількість користувачів, підвищити наочність представлення матеріалу, використовувати електронний посібник тривалий час, звести до мінімуму витрати на пошук і підбір літератури, здійснювати контроль отриманих знань та ін.

Розробка навчальних курсів у середовищі мультимедіа є тривалим і дорогим процесом, тому важливо добре уявляти собі всі основні етапи створення комп'ютерного навчального курсу і прийнятні для кожного етапу розробки рішення.

На першому етапі здійснюється вибір навчального курсу для представлення в середовищі мультимедіа. Тут мають бути виявлені вже існуючі курси з даної дисципліни, визначені передбачувані витрати і час, необхідні для створення курсу, а також його можливий тираж і аудиторія, якій адресований даний курс.

На підготовчому етапі передбачається написання тексту курсу, підбір ілюстративного і довідкового матеріалу, створення ескізів інтерфейсу і сценарію навчальної програми, а також сценаріїв окремих блоків (анімаційних фрагментів, відео-фрагментів, програм, що реалізують комп'ютерне моделювання, блоків перевірки знань і т. п.).

При роботі з текстом навчальної дисципліни необхідно виконати його структуризацію з визначенням точного переліку всіх необхідних тем, що повинні бути викладені в даній дисципліні, розподілом на розділи, параграфи і т. д.

Кожен розділ і весь навчальний курс у цілому досягнуть мети, якщо від початку зазначено, які знання і навички студент повинен отримати.

Паралельно з написанням тексту курсу проводиться робота над сценарієм складової мультимедіа-курсу. Він має містити докладний перелік відповідних компонентів і тем дисципліни, що вивчається, а також попередній опис її структури, що буде реалізовуватися. Сюди відносяться: опис анімаційних, аудіо і відео фрагментів, ілюстрацій, і т. п. Написання сценарію проводиться з урахуванням можливостей обраного програмного забезпечення і наявних вихідних матеріалів.

Повний сценарій курсу має на меті використання звичайного тексту і гіпертексту з посиланнями на зв'язані теми, розділи чи поняття, на зображення, звуки, відео фрагменти, використання табличної інформації, ілюстративного матеріалу (графіків, схем, малюнків), анімаційних малюнків, фотоматеріалів, аудіо і відео фрагментів, комп'ютерних моделей.

На основному етапі виконуються роботи з безпосереднього створення мультимедіа-курсу. Зміст при цьому повинен домінувати над формою його викладу.

Форма представлення матеріалу повинна бути якомога стислішою. Сторінка не повинна містити зайвої інформації (графічної чи текстової), що могла б відвернути увагу читача.

При підборі гарнітури шрифту варто виходити з того, що читабельність тексту, написаного гарнітурою без зарубок, вища, ніж тексту, написаного гарнітурою із зарубками. При цьому варто цілком відмовитися від використання дрібних розмірів шрифтових гарнітур. При включенні в програму графічних зображень потрібно враховувати, що сторінки будуть проглядатися в системах з різним графічним розширенням і глибиною кольору, і орієнтуватися на апаратні засоби, які доступні більшості потенційних користувачів навчальної програми. Використання графічних форматів, що підтримують скомпресоване зображення (GIF, JPEG і т.п.), дозволить скоротити загальний обсяг навчальної програми.

Після проведення завершального етапу відбувається тестування і доробка курсу. Мультимедіа-курс, що пройшов тестування повинен бути зареєстрований як інтелектуальна власність. При цьому необхідно врахувати авторські права колективу розробників, що приймали участь у створенні мультимедіа-курсу.

Корисним є також надання супроводу курсу після його тиражування (зрозуміло, тільки для «легальних» користувачів): оперативне усунення можливих помилок, постачання нових додаткових модулів, поновлення довідкової інформації і т. д.

Необхідно відзначити, що при створенні текстів електронного посібника з дисципліни «Комп’ютерна логіка» необхідно дотримуватися правил розробки електронних сторінок згідно «Закону про електронні документи та електронний документообіг», який встановлює основні організаційно-правові засади електронного документообігу та використання електронних документів.

* 1. **Огляд та структура електронного посібника**

Інтерфейс електронного посібника з дисципліни «Комп’ютерна логіка» побудований наступним чином:

* лекції;
* лабораторні роботи;
* практичні роботи;
* тести для перевірки знань;
* індивідуальні завдання;
* списки груп;
* інформація про користувачів;
* реєстрація;
* профіль користувача / вхід в систему:
* корисна інформація.

Будова всіх сторінок є схожою, відрізняється лише вміст. Всі сторінки посібника складаються з декількох частин:

* Заголовок;
* Ліва панель;
* Основна частина;

Структурну схему сторінки ЕП можна побачити на рисунку 2.1.



Рисунок 2.1 – Структурна схема сторінки електронного посібника

А зараз більш детально розглянемо кожну з частин.

Заголовок (див. рис. 2.2) завжди незмінний і містить назву посібника, дату, привітання та інформацію про автора. Ліва панель (див. рис. 2.3) містить: меню розділів та список гіперпосилань на ресурси Інтернет, які можуть бути корисними.

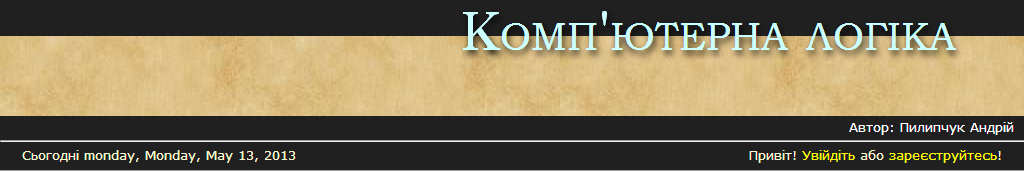


Рисунок 2.2 – Заголовна частина електронного посібника

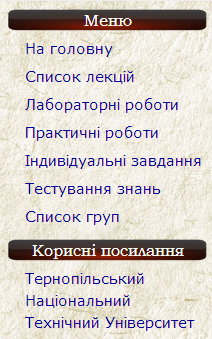


Рисунок 2.3 – Ліва панель: меню та список посилань

В основній частині розташований вміст того чи іншого розділу: чи то список тем, лабораторних, практичних робіт, тестових чи індивідуальних завдань, чи безпосередньо самі лекції, тести, лабораторні чи практичні роботи, індивідуальні завдання, списки груп, студентів чи інша інформація.

Сторінки з лекціями, лабораторними, практичними роботами та індивідуальними завданнями всі мають схожу структуру, відрізняється лише наповнення. Зразок сторінки з лекцією показано на рисунку 2.4.

Тести для перевірки знань є динамічними. Кожен тест характеризується декількома параметрами: загальна кількість питань, кількість питань за одну спробу, часові обмеження на тест загалом та на одне запитання, кількість спроб. Тести поділяються на пробні та звичайні. Пробні тести можуть проходити незареєстровані користувачі, в них зазвичай немає часових обмежень та при проходженні такого тесту інформація про результат не буде внесена до бази даних. Звичайні тести доступні лише зареєстроааним користувачам, інформація про них заноситься в базу даних. Один із таких тестів зображений на рисунку 2.5.



Рисунок 2.4 – Зразок сторінки з лекцією



Рисунок 2.5 – Зразок тесту

Перехід між розділами електронного посібника з дисципліни «Комп’ютерна логіка» здійснюється за допомогою меню розділів з лівого боку, зображеного на рисунку 2.6. Якщо користувач авторизований як адміністратор (викладач), то в меню буде доступний додатковий пункт, що дозволяє потрапити на сторінку адміністратора.



Рисунок 2.6 – Меню розділів (зліва – студент, зправа – викладач)

Авторизовані користувачі можуть переглядати інформацію про інших користувачів своєї групи (рис. 2.7). Користувачі групи «Викладачі» можуть переглядати інформацію про всіх користувачів.

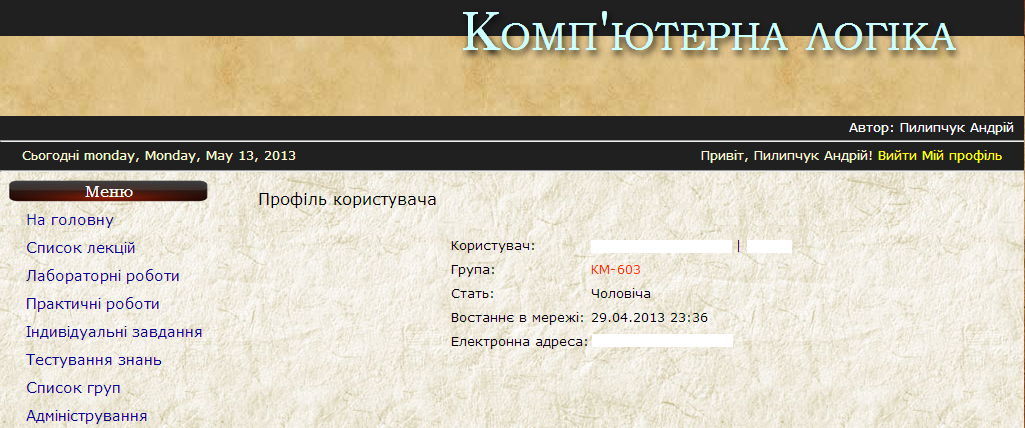


Рисунок 2.7 – Профіль користувача

* 1. **Розробка структури електронного посібника**

Внаслідок загальної комп’ютеризації суспільства, разом з традиційними навчальними посібниками почали використовуватися, так звані, електронні підручники, зручні і для домашнього навчання, і для використання в освітніх установах. Електронні підручники істотно підвищують якість візуальної інформації, вона стає яскравішою, барвистою, а як відомо, чим цікавіше представлений матеріал, тим цікавіше його вивчати. Тому можливості електронних підручників не обмежуються тільки викладом і демонстрацією матеріалу, вони покликані зацікавити учня.

Слідуючи сучасним тенденціям, мною було розроблено електронний підручник з дисципліни «Комп’ютерна логіка».

Мета дипломного проекту **–** створити електронний підручник, який можна було б використовувати на уроках «Комп’ютерної логіки» і не тільки, а також для самостійного навчання.

Тема підручника «Комп’ютерна логіка» входить в курс вищих навчальних закладів. Метою профільного курсу «Комп’ютерна логіка» є формування важливої складової інформаційної культури учнів – знань та вмінь, необхідних для повноцінного та ефективного знань в галузі розробки програмних продуктів.

Структура підручника складається з декількох пунктів: меню, лекції, лабораторні та практичні роботи, індивідуальні завдання, тести, сторінка реєстрації, інформації про користувачів, інформації про групи користувачів та сторінка входу в систему. Структура веб-проекту складається з декількох папок: Content – для вмісту основних сторінок посібника; Source – де зберігаються файли дизайну та інтерфейсу посібника; Uploads – сюди заносяться усі файли завантажені користувачами.

Сторінка другого ряду може містити:

* Лекції, у яких наведено теоретичні матеріали. Такі сторінки мають фреймову структуру і для зручності користування розбиті на дрібні підтеми, відповідно до навчального змісту матеріалу.
* Лабораторні та практичні роботи, призначені для самостійного засвоєння та практичного відпрацювання матеріалу. Вони являють собою набір інструкцій для чіткого і правильного виконання робіт.
* Індивідуальні завдання розбиті по темах – місять варіанти завдань до лабораторних та практичних робіт.
* Тестові завдання, пристосовані для перевірки знань. В підручнику розроблено інтерактивні тести за допомогою об’єктно-орієнтованого програмування ASP.NET та базами даних SQL Server. Студенту пропонується обрати одну вірну відповідь з кількох запропонованих. Тести такого роду є гарним допоміжним засобом викладачеві для здійснення поточного контролю. При завантаженні тесту, якщо кількість питань доступних за одну спробу перевищує загальну кількість питань – випадковим чмном вибирається необхідна кількість запитань, запитання в межах одного тесту не можуть повтоюватись. Всі запитання ще раз перемішуються, щоб знизити до мінімуму шанс а те, що запитання будуть певним чином впорядковані. Варіаним відповіді до кожного запитання теж в свою чергу перемішуються. Таким чином шанс на те що 2 чи більше запитань будуть в декількох тестах в однаковому порядку, та шанс та те, що студент запам’ятає порядковий номер правильної відповіді майже рівний нулю.

Фінальним завершенням основи проекту є розміщення електронного посібника в мережі Інтернет для зручного доступу та ще більшої захищеності даних на віддаленому сервері.

Результатом роботи є яскравий електронний підручник, який буде необхідним для самостійної роботи учнів, корисним на практичних заняттях, при тестуванні знань студентів та зручним для викладача.

Робота над підручником проводилася у декілька етапів.

На підготовчому етапі створення електронного підручнику було визначено бібліографію по темі роботи, здійснювався аналіз джерел, збір та узагальнення матеріалу.

Перший етап розробки був присвячений розробці структури підручника, тобто було проведено розбиття матеріалу на розділи, що складаються з модулів, мінімальних за об'ємом, але замкнутих за змістом, а також складено дидактичне наповнення підручника**.**

Другий етап –розробка елементів дизайну.

Третій етап – розробка інтерфейсу MasterPage та кожної сторінки зокрема.

Четвертий етап – налаштування бази даних та доступу до неї.

П’ятий етап – розробка та написання програмного коду основних сторінок: MasterPage, сторінки реєстрації, входу в систему, та адміністрування.

Шостий етап – створення зразків сторінок: текстових (лекції, лабораторні, практичні роботи, індивідуальні завдання) та сторінки з тестами.

Сьомий етап **–** верстка НТML-сторінок. В результаті створюється працюючий електронний підручник, який володіє властивостями, що роблять його необхідним для учнів, корисним для самостійних занять і зручним для викладачів..

* 1. **Особливості серверного коду**

Код сторінок електронного посібнка виконується на стороні сервера, перетворюється в готовий HTML-документ та відправляєтьчя клієнту. Код сторінок написано на мові C# та для кожної сторінки є окремим. Код сторінки MasterPage є спільким для всіх сторінок та виконується кожен раз при завантаженні сторінки, містить лише функції та дії загального виконання, такі як, авторизація користувача, чи вихід з системи.

Під час розробки електронного посібника з дисципліни «Комп’ютерна логіка» було написано декілька тисяч рялків серверного коду. Кожна сторінка має свій власний файл з серверним кодом – файл з розширенням .aspx.cs. В першу чергу на кожній сторінці де вказані обмеження достуду користувачів до інформації на сторінці виконується перевірка авторизації та рівня доступу користувача. Наприклад:

if (Session["student"] != null || Session["teacher"] != null)

{

…

}

else

Response.Redirect("~/Content/Login.aspx?url=" + Request.RawUrl);

Ця перевірка виконується при завантаженні сторінки, яякщо перевірка авторизації пройшла успішо – завантажуться інформація з бази даних, в іншому випадку – перенаправляє користувача на сторінку входу в систему.

Або наприклад перевірка рівня доступу користувача:

if (Session["teacher"] != null)

{

…

}

Ця перевірка використовується в першу чергу на сторінці адміністатора.

Також варто розглянути функцію, що виконується при кожному завантаженні сторінки MasterPage.

protected void Page\_Load(object sender, EventArgs e)

{

Date.Text = "Сьогодні " + DateTime.Today.DayOfWeek.ToString().ToLower() + ", " + DateTime.Today.Date.ToLongDateString();

if (Session["SurName"] != null && (Session["student"] != null || Session["teacher"] != null))

{

Loggedin\_Username.Text = "Привіт, " + Session["SurName"].ToString() + "!";

LoginMultiview.ActiveViewIndex = 1;

if (Session["teacher"] != null)

Administrating\_menu\_link.Visible = true;

// update "last login" info in DB

string Username = Session["teacher"] != null ? Session["teacher"].ToString() : Session["student"].ToString();

string SQL\_UPDATE = "UPDATE " + UsersDB + " SET LastLogin='" + DateTime.Now.ToShortDateString() + " " + DateTime.Now.ToShortTimeString() + "', PRecover=NULL WHERE Username='" + Username + "'";

SqlCommand CMD\_UPDATE = new SqlCommand(SQL\_UPDATE, DB\_Connection);

CMD\_UPDATE.CommandType = CommandType.Text;

DB\_Connection.Open();

CMD\_UPDATE.ExecuteNonQuery();

DB\_Connection.Close();

}

}

Ця функція виконує декілька дій, в першу чергу встановлює правильне відображення дати на сторінці, також перевіряє, чи користувач авторизований, обновлює деякі змінні на сторінці, та обновлює інформацію про останній вхід в систему в базі даних.

Функція, що забезпечує вихід з системи:

protected void LogoutButton\_Click(object sender, EventArgs e)

{

if (Session.SessionID == "student")

Session.Remove("student");

else

Session.Remove("teacher");

Session.Remove("SurName");

if (Path.GetFileName(Request.Url.AbsolutePath) == "Administrating.aspx")

Response.Redirect("~/Content/Default.aspx");

else

Response.Redirect(Request.RawUrl);

}

Це проста функія, що просто очищає інформацію про поточну сесію користувача, та, за необхідності, перенаправляє користувача на головну сторінку.

* 1. **База даних SQL та її структура**

В новій версії електронного посібника з дисципліни «Комп’ютерна логіка» використано об’єктно-орієнтовану мову програмування ASP.NET C#, що дозволяє працювати з базами даних типу SQL Server та MySQL.

При розробці даного електронного посібника було використано базу даних SQL, оскільки вона є найбільш сумісною з платформою .NET Framework.

База даних складається з 10 таблиць, кожна з яких має свою структуру та відповідає за певну область інформації, розміщеної в електронному посібнику. Імена таблиць можуть бути змінені, але при цьому також потрібно поправити відповідні пункти в файлі налаштувань електронного посібника (web.config).

Список таблиць бази даних (імена вказані по замовчуванню) та їх основні функції:

* CL\_Users – призначена для зберігання інформації про зареєстрованих користувачів;
* CL\_Groups – містить в собі інформацію про існуючі групи користувачів;
* CL\_Lessons – тут зберігаються всі лекції, що доступні в електронному посібнику;
* CL\_Labs – структура таблиці ідентична з CL\_Lessons, але тут знаходять лабораторні роботи.
* CL\_Practice – структура таблиці схожа з CL\_Lessons, але тут зберігаються практичні робити.
* CL\_Individuals – індивідуальні завдання для лабораторних та практичних робіт.
* CL\_Tests – містить основну інформацію про тести: тему, кількість доступних за 1 спробу питань, кількість допустимих спроб, часові обмеження та ін.
* CL\_TestQuestions – зберігає всі питання до всіх тестів, приналежність певного питання до певного тесту оприділяється унікальним ідентифікатором тесту.
* CL\_TestAnswers – зберігає всі можливі варіанти відповідей до всі питань тестів, приналежність ввідповіді до питання, як і в попередньому випадку, оприділяється унікальним ідентифікатором питання.
* CL\_Marks – тут зберігаються всі оцінки за всі пройдені користувачами тести.

Більшість таблиць має унікальні ключі (один або декілька), що дозволяє уникнути повторного внесення одинакових даних в одну таблицю. Також всі таблиці, що взаємодіють з іншими таблицями оснащені полями з зовнішнім ключем. Це вбереже базу даних від так званих «осиротілих даних», коли певні рядки перних таблиць існують, але через втрачені зв’язки з іншими таблицями (при видаленні даних, чи зміні ключових значень) більше ніколи не можуть бути використані.

* 1. **Розробка специфікацій вимог до ЕП**

Електронний посібник, що буде кінцевим результатом дипломного проекту повинен відповідати усім стандартам вимог IEEE 830, а саме:

1. Вимоги до інтерфейсу:
   1. Загальні вимоги до інтерфейсу:
      1. Колір загального фону повинен бути світлим
      2. Колір фонової частини електронного посібника повинен бути світлим
      3. Колір тексту повинен бути темним
      4. Колір меню понинен бути темнішим від фону
      5. Шрифт тексту повинен бути 14
      6. Шрифт тексту повинен бути читабельним
      7. Проміжки між рядками повинен бути 1,5 рядки
   2. Вимоги до клієнтської частини:
      1. Розмежування інформації в електронному посібнику для авторизованих користувачів та гостей
      2. Простота навігації між розділами посібника
      3. Вміст електронного посібника не можливо копіювати
   3. Вимоги до адміністративної частини:
      1. Легкість навігації
      2. Простота виконання операцій
      3. Обмеження доступу до адміністративних елементів посібника
2. Функції
   1. Можливості, доступні всім користувачам
      1. Перегляд сторінок, що не мають обмежень доступу
      2. Перегляд лекцій
      3. Перегляд лабораторних робіт
      4. Проходження «пробних» тестів
      5. Реєстрація в електронному посібнику
      6. Авторизація в системі
      7. Відновлення втраченого пароля
   2. Можливості, доступні авторизованим студентам
      1. Перегляд практичних робіт
      2. Перегляд індивідуальних завдань
      3. Проходження всіх тестів
      4. Збереження інформації про пройдені тести в базі даних
      5. Зміна електронної адреси до власного облікового запису
      6. Зміна паролю до власного облікового запису
   3. Можливості, доступні авторизованим викладачам
      1. Редагування лекцій
      2. Видалення лекцій
      3. Написання нових лекцій
      4. Редагування лабораторних робіт
      5. Видалення лабораторних робіт
      6. Написання нових лабораторних робіт
      7. Редагування практичних робіт
      8. Видалення практичних робіт
      9. Написання нових практичних робіт
      10. Редагування списків індивідуальних завдань
      11. Видалення списків індивідуальних завдань
      12. Створення нових списків індивідуальних завдань
      13. Видалення тестів
      14. Створення нових тестів
      15. Перегляд результатів пройдених користувачами тестів
      16. Активація облікових записів користувачів
      17. Зміна рівня доступу користувачів
      18. Створення нових груп користувачів
      19. Видалення існуючих груп користувачів
      20. Переміщення користувачів між групами
      21. Завантаження файлів на сервер
      22. Виконання операцій з таблицями баз даних
3. Вимоги до бази даних
   1. Можливість створення додаткових таблиць
   2. Використання основних ключів для ідентифікації даних в таблицях
   3. Використання зовнішніх зв’язків (ключів) між таблицями
   4. Можливість перейменування таблиць (при цьому є обов’язковим редагування файлу налаштувань електронного посібника (web.config)
4. Обмеження проектування
   1. Використання об’єктно-орієнтованого програмування
   2. Використання бази даних SQL Server
   3. Розміщення кінцевого варіанту електронного посібника з дисципліни «Комп’ютерна логіка» на віддаленому сервері (хостингу)
   4. Кінцевий термін здачі проекту – 14 червня 2013р.
5. Атрибути програмної системи
   1. Надійність
      1. Регулярно повинні здійснюватись резервні копії бази даних
      2. При внесенні нових даних в таблиці не повинні бути пошкоджені вже існуючі дані
   2. Доступність
      1. Електронний посібник повинен бути доступним 24 години на добу
      2. Електронний посібник повинен корекнто відображатись незалежно від діагоналі монітору
      3. Коректне відображення електронного посібника незалежно від операційної системи, чи браузера
   3. Захищеність
      1. Відсутність у клієнтській базі модулів, що можуть пошкодити базу даних
6. Організація конкретних вимог
   1. Мінімальні вимоги до апаратної частини користувача
      1. Процесор з мінімальною частотою 800 MHz
      2. Відео карта 128 Мб
      3. Оперативна пам’ять 512 Мб
      4. Наявність підключення до мережі Інтернет
   2. Мінімальні вимоги до програмної частини сервера
      1. Підтримка .NET Framework 4.0 або вище
      2. SQL Server 2008 або вище
   3. Мінімальні вимоги до апаратної частини сервера
      1. Процесор з частотою 2,2 GHz
      2. Оперативна пам’ять 2 Гб
      3. Пропускна здатність каналу 10 Мбіт/c
      4. Необмежений трафік
   4. **Тестування електронного посібника**

Дана дипломна робота виконувалась на комп’ютері Intel Core2Duo 1.8 ГГц 3Гб RAM, відео карта Radeon HD 5670 1GB DDR3 під управлінням операційної системи Windows7. Монітор BenQ sensseye 20”. Встановлено найбільш популярні браузери останньої (на даний момент) версії – Google Chrome 26.0, Mozilla Firefox 19.0 та Opera 12.11.

Оскільки на даний час існує велика кількість різних конфігурацій ПК та встановленого на них програмного забезпечення, виникла необхідність встановити рамки в яких даний веб-ресурс буде коректно працювати.

Даний електронний посібник було повністю протестовано на середньостатистичному ПК який працює з Інтернет з такою конфігурацією:

* швидкість підключення – 5 Мбіт.

Програмне забезпечення, необхідне для роботи з електронним посібником, встановлене на ньому:

* Google Chrome 26.0;
* Mozilla Firefox 19.0;
* Internet Explorer 8;
* Java Platform.

Тестування електронного посібника з дисципліни «Комп’ютерна логіка» виконувалось в декілька етапів. Перший етам тестування був спрямований за клієнтську частину веб-проекту. Було перевірено середню швидкість завантаження кожної сторінки посібника, швидкість завантаження зображень, а також якість відображення стилів в браузерах, в яких електронний посібник повинен працювати, також будо перевірено всі наявні гіперпосилання на внутрішні та зовнішні ресурси.

Другим етапом тестування було виконання операцій пов’язаних з доступом до бази даних, перевірка правильності виконання всіх дій за різних умов та в різній послідовності. Виявлені помилки було виправлено.

Під час третього етапу було перевірено роботу електронного посібника з зовнішніми ресурсами, такими, як відправлення повідомлень на електронну пошту через Gmail, підтвердження «безпечного зв’язку» та налаштування пошти електронного посібника.

Для тестування електронний посібник було розміщено за безплатному домені SmarterASP з безплатним пакетом послуг на 60 днів. http:://excalibur-001-site1.smarterasp.net. В подальному планується переміщення його на сервер Технічного коледжу ТНТУ.

**3 СПЕЦІАЛЬНИЙ РОЗДІЛ**

Розробці електронного посібника з дисципліни «Комп’ютерна логіка» передувало вивчення матеріалів спеціалізованих періодичних видань, а також опрацювання інформації в мережі Інтернет.

За результатами пошуку інформації в Інтернеті й спеціалізованих періодичних виданнях, та посібниках було прийнято рішення на основі знайдених матеріалів, розробити електронний посібник з дисципліни «Комп’ютерна логіка».

* 1. **Встановлення і налаштування IIS 7**

Для встановлення служб IIS і працюючих на їх базі додатків можна скористатися програмою встановлення веб-платформи. Ця програма є безкоштовною, має невеликий розмір і дозволяє встановлювати служби IIS та пов’язані з ними технології, такі як ASP.NET, SQL Server Express, Visual Web Developer, інші популярні веб-додатки та багато іншого. Оскільки ця програма встановлює останні версії доступних пропозицій веб-платформи, ви зможете завантажувати і встановлювати будь-які нові засоби та оновлення буквально одним клацанням миші.

Веб-сервер IIS 7.0 можна встановити двома різними способами за допомогою Windows Server 2008.

1. Як частину режиму Корінного Сервера, де IIS 7.0 встановлений в якості однієї з ролей Сервера.
2. Нормальним шляхом через графічний інтерфейс GUI, де власне і встановлений IIS 7.0 в якості однієї з ролей Сервера.

В Windows Vista IIS може бути встановлений двома способами: через нормальный графічний інтерфейс GUI внаслідок чого добавляються нові Windows Features в Vista, або в якості складової автоматичного встановлення.

**IIS в Windows Server 2008**

### IIS 7.0 – це одна із серверних ролей Windows Server 2008. IIS може бути встановлено через графічний інтерфейс GUI з використанням нового інтерфейсу Server Manager після завершення встановлення Windows Server 2008.

### Server Manager являє собою пристрій для встановлення або видалення ролей і засобів сервера. Також він повідомляє про всі встановлені ролі і пристрої. Коли ви вибирете IIS 7.0 для Server Manager, то автоматично вибируться базові компоненти, які необхідні для IIS.

### Встановлення IIS 7.0 в Windows Server 2008 трохи відрізняється від встановлення IIS 6.0, але вона так само легка, якщо користуватись Server Manager.

### Також можна встановити IIS через Server Manager, який знаходится в:

Меню Пуск – Server Manager (також його можна знайти в: Меню Пуск– Административные устройства – Server Manager), далі виберіть “Add Roles” в меню Roles і виберіть “Web Server (IIS)” на сторінці Server Roles (рис 3.1).

Ще тут щось дописати

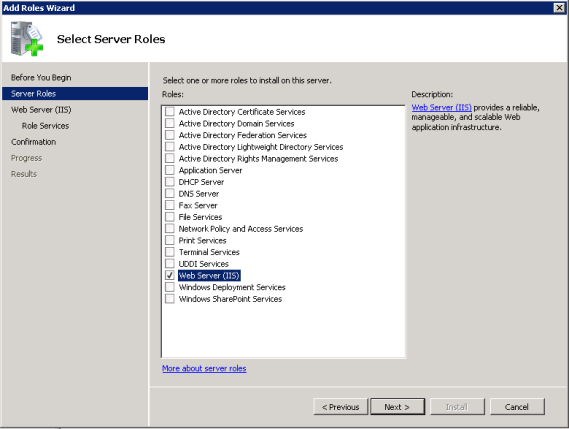


Рисунок 3.1 – Сторінка вибору ролей сервера

Виберіть IIS Services, які потрібно встановити. В цьому випадку буде встановлений ASP.NET. Виберіть сервіси, які потребує ваша прикладна програма, як показано на рисунку 3.2.

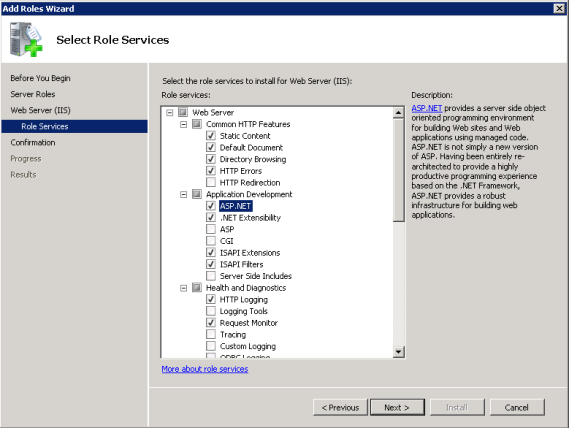


Рисунок 3.2 – Сторінка вибору сервісів

Коли ви будете вибирати сервіси, приведені вище, то вас можуть застерегти про встановлення додаткових сервісів, необхідних для обраних вами, якщо ці сервіси ще не встановлені (рис 3.3).

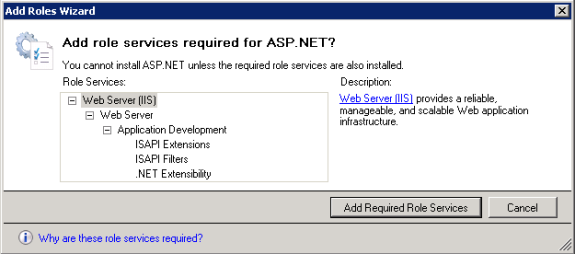


Рисунок 3.3 – Сторінка з попередженням Wizard

На рисунку 3.4 показано вікно з результатами успішного встановлення IIS 7.0 з налаштуваннями по замовчуванню для хостингу ASP.NET в вашому Windows Server 2008.

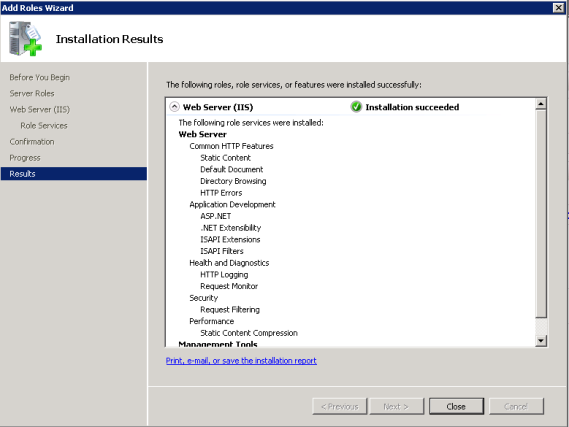


Рисунок 3.4 – Сторінка результатів встановлення

Наступним кроком переконайтесь,що веб-сервер працює з перейшовши на <http://localhost>. Якщо все працює правильно ви побачите заставку IIS 7, як показано на рисунку 3.5

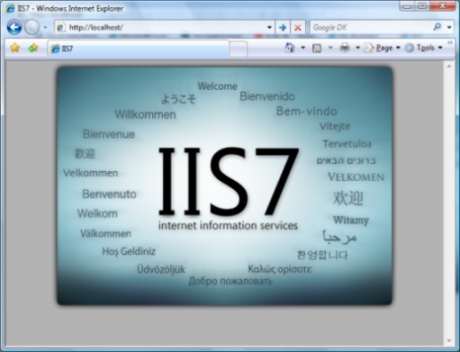


Рисунок 3.5 – Вебсайт по замовчуванню

Встановіть лише найбільш необхідні сервіси IIS, щоб мінімізувати навантаження IIS на систему. Це також зменшить поверхню атаки, що є одним з переваг нового IIS 7.0.

**Корінний сервер (Server Core)**

### Windows Server 2008 може бути встановлений в режим корінного сервера (Server Core). Корінний сервер (Server Core) вибирається протягом початкового встановлення Windows Server 2008.

### Режим корінного сервера (Server Core) – це новий пристрій у Windows Server 2008. Це найменше за обсягом встановлення сервера, коли для запуску певних сервісів, вже встановлених, потрібна мінімальна кількість двійкових файлів, наприклад, веб-сервера. Це означає, що встановлюється навіть не Windows Shell (GUI), а тільки мінімальний обсяг файлів, який необхідний для запуску сервера, і після завантаження у адміністратора з'являється командний рядок.

### IIS 7.0 – одна з ролей сервера, яка може бути встановлена ​​в Windows Server 2008, запускається в режимі корінного сервера (Server Core).

### Введіть наступну команду для встановлення IIS 7.0 в сервер, що запускає режим корінного сервера (Server Core):

### Pkgmgr.exe/iu:IIS-WebServerRole;WAS-WindowsActivationService;WAS-ProcessModel

### Ця команда зробить мінімальне (за замовчуванням) встановлення IIS 7.0.

### Зараз у Windows Server 2008 в режимі корінного сервера (Server Core) доступні тільки класичні веб-прикладні програми, такі як ASP і PHP. У режимі корінного сервера (Server Core) недоступні наступні сервіси IIS services:

* IIS-ASPNET
* IIS-NetFxExtensibility
* IIS-ManagementConsole
* IIS-ManagementService
* IIS-LegacySnapIn
* IIS-FTPManagement
* WAS-NetFxEnvironment
* WAS-ConfigurationAPI

У режимі корінного сервера (Server Core) управління та налаштування IIS повинні проводитися через файли налаштувань. Config.xml або через командний пристрій appcmd.exe. Також доступні такі інтерфейси, як WMI і COM, віддалене усправління можливе з використанням MMC 3.0 і вбудованих модулів.

Переконайтеся, що встановлення проходить за допомогою копіювання або створення деяких .html веб-сторінок і вони поміщуються в кореневу папку Inetpub.

**Автоматичне встановлення**

Автоматичне встановлення – один із способів послідовного та автоматичного встановлення IIS 7.0. Якщо вам потрібно використовувати веб-сервери і ви хочете переконатися, що кожен з них встановлений з ідентичними компонентами та сервісами, то вам буде необхідно скористатися автоматизованим встановленням з використанням автоматичних скриптів.

Це робиться різними способами з використанням Windows Server 2003. У Windows Server 2008 і Windows Vista пристрій, що використовується для автоматизованих скриптів, називається pkgmgr.exe. Це основний компонент, який використовується в процесі автоматичного встановлення – він дозволяє вам встановити або видалити з командного рядка або з скрипта. Pkgmgr.exe замінює sysocmgr.exe з Windows Server 2003.

Перш ніж використовувати pkgmgr.exe, переконайтеся, що ви маєте права адміністратора.

Введіть таку команду через командний рядок або черз скрипт:

pkgmgr.exe /iu:

IIS-WebServerRole;

IIS-WebServer;

IIS-CommonHttpFeatures;

IIS-StaticContent;

IIS-DefaultDocument;

IIS-DirectoryBrowsing;

IIS-HttpErrors;

IIS-HttpRedirect;

IIS-ApplicationDevelopment;

IIS-ASPNET;

IIS-NetFxExtensibility;

IIS-ASP;

IIS-CGI;

IIS-ISAPIExtensions;

IIS-ISAPIFilter;

IIS-ServerSideIncludes;

IIS-HealthAndDiagnostics;

IIS-HttpLogging;

IIS-LoggingLibraries;

IIS-RequestMonitor;

IIS-HttpTracing;

IIS-CustomLogging;

IIS-ODBCLogging;

IIS-Security;

IIS-BasicAuthentication;

IIS-WindowsAuthentication;

IIS-DigestAuthentication;

IIS-ClientCertificateMappingAuthentication;

IIS-IISCertificateMappingAuthentication;

IIS-URLAuthorization;

IIS-RequestFiltering;

IIS-IPSecurity;

IIS-Performance;

IIS-HttpCompressionStatic;

IIS-HttpCompressionDynamic;

IIS-WebServerManagementTools;

IIS-ManagementConsole;

IIS-ManagementScriptingTools;

IIS-ManagementService;

IIS-IIS6ManagementCompatibility;

IIS-Metabase;

IIS-WMICompatibility;

IIS-LegacyScripts;

IIS-LegacySnapIn;

IIS-FTPPublishingService;

IIS-FTPServer;

IIS-FTPManagement;

WAS-WindowsActivationService;

WAS-ProcessModel;

WAS-NetFxEnvironment;

WAS-ConfigurationAPI

Вищенаведена команда встановить IIS 7.0 з усіма доступними сервісами. Виберіть тільки ті сервіси, які вимагає ваша прикладна програма.

Та ж команда може бути запакована в xml-файл і може запускати pkgmgr.exe з файлами параметрів xml, також через командний рядок:

pkgmgr.exe /n:IIS7\_unattend.xml

Нижче наведено приклад IIS7\_unattend.xml файлу, який встановить IIS 7.0 з пристроями за замовчуванням:

<?xml version="1.0" ?>

<unattend xmlns="urn:schemas-microsoft-com:unattend"

xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State">

<servicing>

<package action="configure">

<assemblyIdentity

name="Microsoft-Windows-Foundation-Package"

version="6.0.6001.16659"

language="neutral"

processorArchitecture="x86"

publicKeyToken="31bf3856ad364e35"

versionScope="nonSxS" />

<selection name="IIS-WebServerRole" state="true"/>

<selection name="WAS-WindowsActivationService" state="true"/>

<selection name="WAS-ProcessModel" state="true"/>

<selection name="WAS-NetFxEnvironment" state="true"/>

<selection name="WAS-ConfigurationAPI" state="true"/>

</package>

</servicing>

</unattend>

</pre>

Потрібно зауважити, що версія Windows, сконструйована через xml-файл (версія = "6.0.6001.16659"), є номером повної версії OS, інакше xml-файл не буде працювати.

Відкрийте властивості файлу regedit.exe, щоб отримати номер останньої конструкції, як показано на рисунку 3.6.

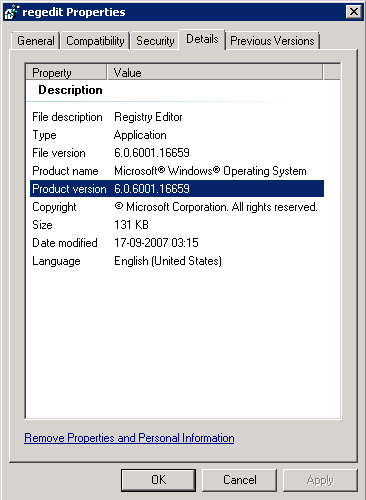


Рисунок 3.6 – Властивості Regedit.exe

* 1. **Встановлення та налаштування SQL Server 2008**

Перш ніж встановлювати SQL Server 2008 потрібно завантажити «Центр встановлення SQL Server» безкоштовно з офіційного веб-сайту компанії Microsoft. Після завантаження файлу встановлення запустіть його та виконайте декілька простих кроків. Зверніть увагу на конфігурацію при завантаженні, x86 та x64 залежно від того під якою конфігурацією працює операційна система.

Після завершення встановлення знайдіть в меню та виконайте файл завантаження програми «Центр встановлення SQL Server». У розділі «Планування» рекомендовано виконати перевірку конфігурації, щоб дізнатись чи може SQL Server 2008 бути успішно встановлений (рис 3.7).

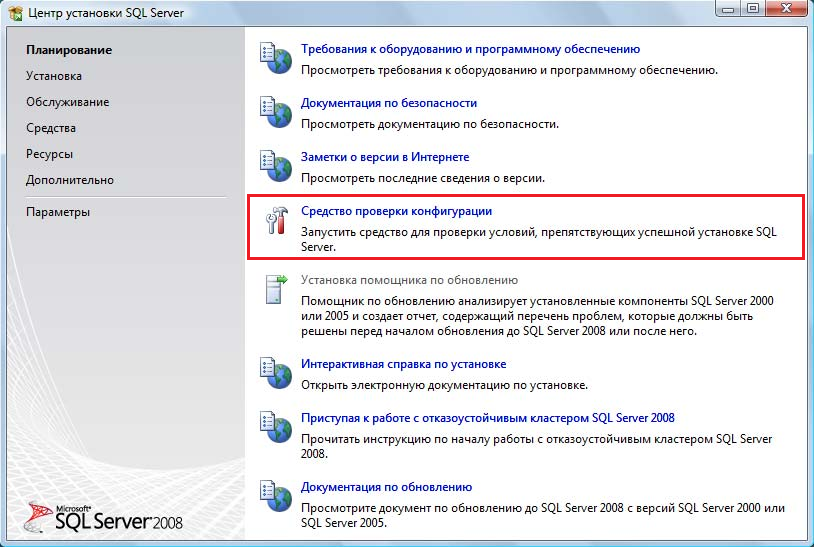


Рисунок 3.7 – Центр встановлення SQL Server

Якщо «Засіб перевірки конфігурації» не виявив помилок – можна приступити безпосередньо до встановлення SQL Server 2008. В меню «Встановлення» оберіть пункт «Нове встановлення ізольованого SQL Server чи додання компонентів до вже існуючого», як покажано на рисунку 3.8.

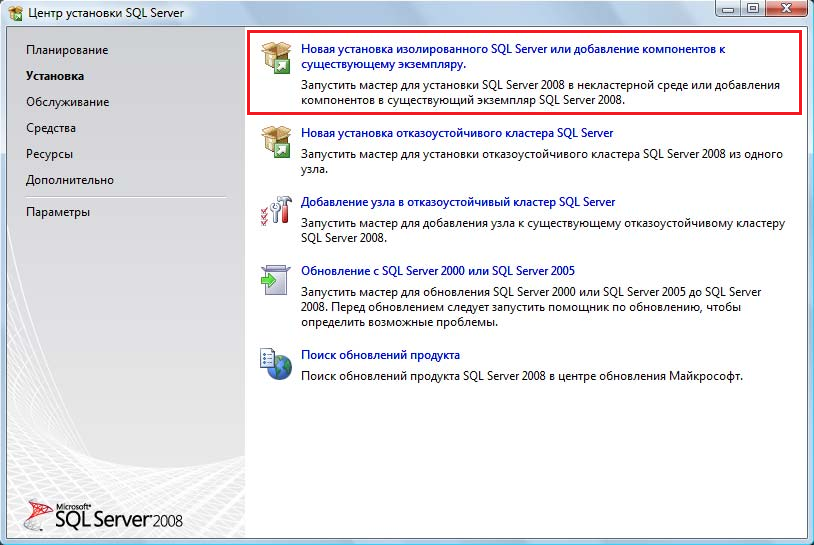


Рисунок 3.8 – Встановлення SQL Server 2008

Пройдіть декілька простих кроків: ввід ліцензійного ключа (для безплатних версій не обов’язково), погодження з правилами ліцензійної угоди та встановіть файли підтримки програми, щоб перейти до майстра встановлення SQL Server (рис 3.9).

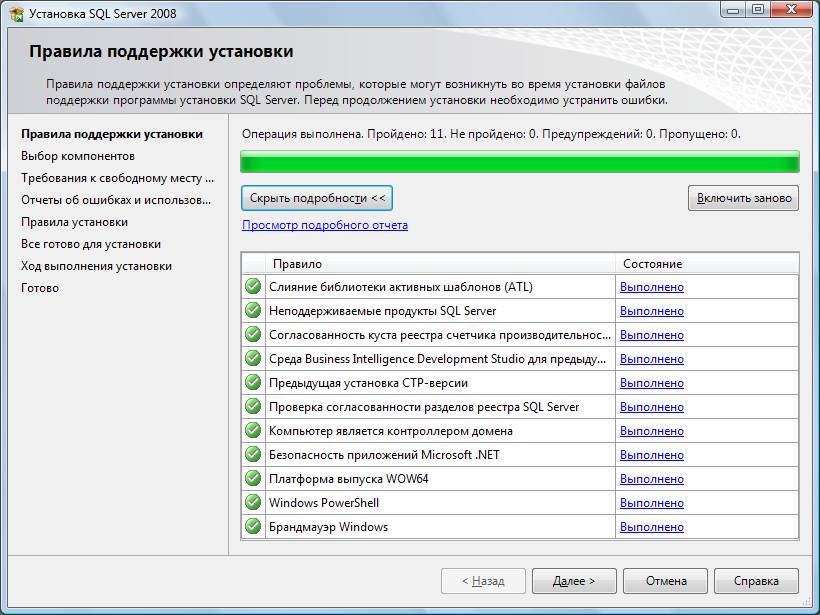


Рисунок 3.9 – Початок встановлення SQL Server 2008

На цьому етапі система знову виконає перевірку системи на сумісність з SQL Server 2008, після чого запропонує Вам обрати компоненти для встановлення. Саме тут і починається кінцевий етап встановлення (рис 3.10).

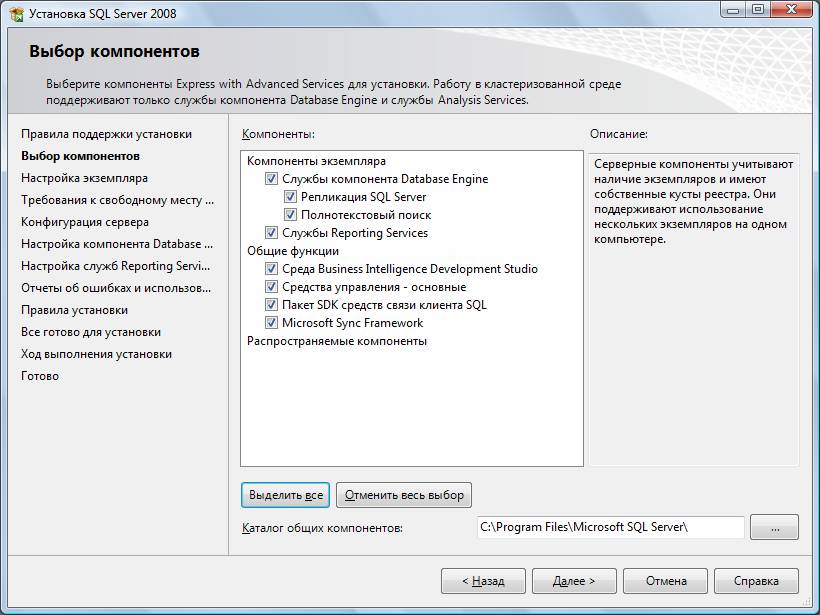


Рисунок 3.10 – Вибір компонентів для встановлення

Рекомендовано обрати всі компоненти. Тепер потрібно вказати основні налаштування майбутнього сервера – ідентифікатор та місце розташування екземпляру. Приклад нашалтувань показано на рисунку 3.11. Після цього буде виконано перевірку вільного місця на вказаному жорсткому диску. В меню конфігурації сервера рекомендовано залишити все без змін. Варто лише звернути увагу на вкладку «Параметри сортування», тут можна налаштувати сортування майбутніх баз даних. В залежності від того якою мовою дані будуть вноситись в базу даних рекомендовано змінити сортування. По замовчуванню встановлено латинське. Для бази даних електронного посібниика з дисципліни «Комп’ютерна логіка» було обрано українську сортування Ukrainian\_CI\_AS. Вікно параметрів вибору сортування показано на рисунку 3.11.

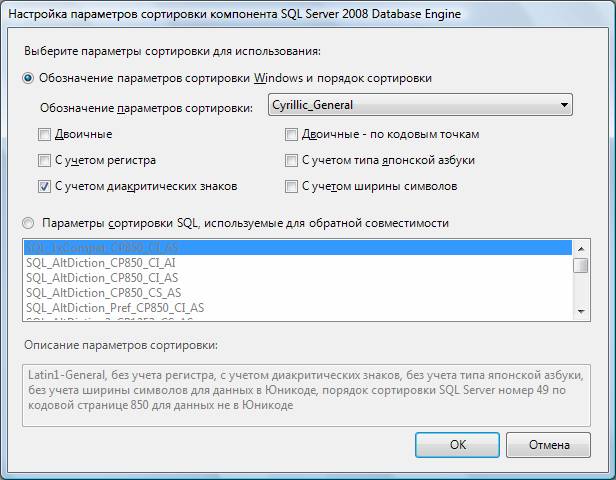


Рисунок 3.11 – Вікно налаштування параметрів сортування

Першим кроком є вибір методу аутентифікації користувачів бази даних: засобами Windows або змішаний – і Windows і SQL Server. Вкажіть пароль та добавте користувачів, які зможуть користуватись базою даних (рис 3.12).

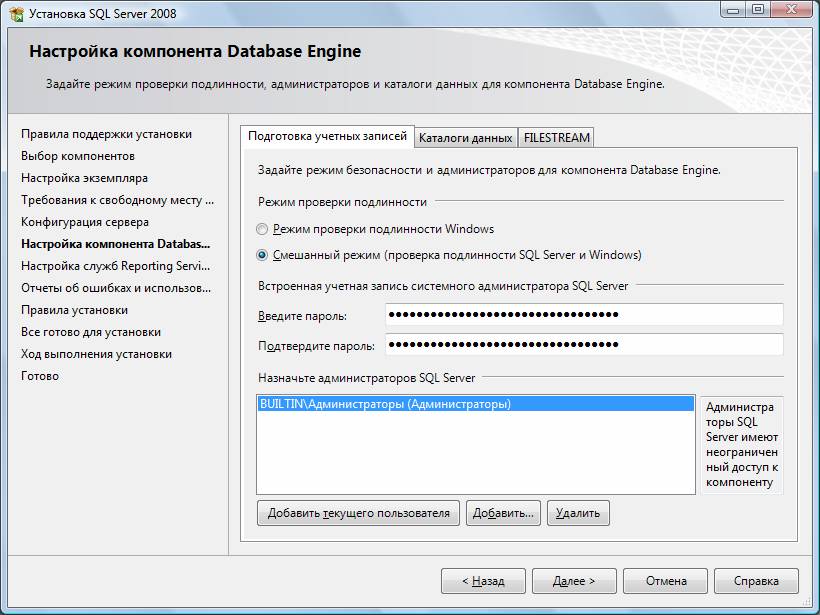


Рисунок 3.13 – Налаштування параметрів користування

Параметри налаштування каталогів рекомендовано залишити без змін.

Підготовчий етап завершений, тепер все готово для встановлення SQL Server 2008. Ще раз перевірте усі параметри встановлення і розпочинайте встановлення.

Після того, як SQL Server 2008 буде успішно встановлено можна розпочати користування зайшовши у SQL Server Management Studio (рис 3.14) використовуючи свої дані залежно від того, який спосіб аутентифікації Ви обрали під час встановлення SQL Server 2008.

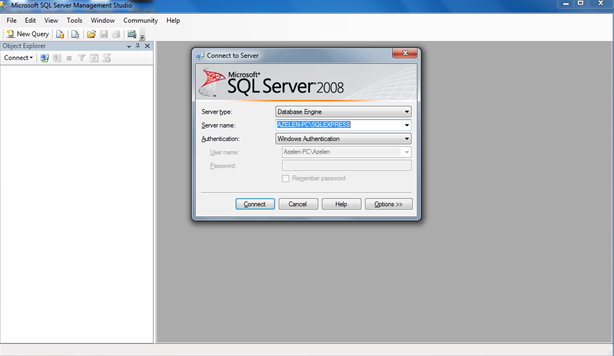


Рисунок 3.14 – Аутентифікація в SQL Management Studio

* 1. **Інструкція з експлуатації електронного посібника**

Важко очікувати, що кожен викладач у достатньому обсязі освоїть техніку web-програмування, але значну частину роботи, необхідну для створення сучасного мультимедіа-курсу, орієнтованого на мережеве використання (веб-курсу), може виконати практично кожний.

Нижче наводяться рекомендації, що можуть бути корисними для такої підготовчої роботи:

По-перше в цьому випадку необхідно створити електронну версію курсу в текстовому редакторі Word. При цьому можна використовувати інструментальні засоби створення малюнків, формульний редактор, можливості вставки зображень і т. д.

Маючи веб-версію курсу, особливу увагу варто приділити структуруванню матеріалу – поділу його на розділи, підрозділи, складанню змістів розділів. Кожна мінімальна логічна частина (підрозділ) повинна бути поміщена в окремий файл, а всі створені файли варто розмістити в одному каталозі, що при необхідності має підкаталоги.

При структуруванні посібника варто враховувати, що розмір кожного файлу повинен бути таким, щоб при його відображенні в web-браузері він не займав більше 2-3 екранів. Звичайно це відповідає 2-3 сторінкам тексту в doc-файлі при розмірі шрифту 14.

Наступним кроком необхідно створити необхідні гіпертекстові посилання, що встановлюють зв'язки, як між окремими розділами, так і з необхідними ресурсами Інтернет. Файл, з яким установлюється зв'язок, повинен вже існувати в створеному каталозі посібника.

В принципі, робота з текстом матеріалу на розглянутому (підготовчому) етапі на цьому може завершитися. Проте, якщо викладач має намір подальшу роботу проводити самостійно чи активно брати участь у роботі разом із програмістом, корисно зробити ще одну операцію – перетворення тексту в HTML- формат.

Для цього в меню «Файл» необхідно вибрати режим «Зберегти як…», у вікні «Збереження документа» у полі «тип файлу» встановити «HTML Document». У результаті цієї операції буде створений файл \*.html, де \* – ім'я doc-файлу. При цьому усі формули, написані у формульному редакторі Word, будуть автоматично перетворені у файли-зображення формату .jpg (імена цих файлів за замовчуванням image\*.jpg, де \* – порядковий номер 1,2,3...).

На сам кінець потрібно скопіювати всі html-файли з дотриманням структури каталогів посібника і передати матеріал програмісту web-майстру сервера, на якому передбачається розміщення посібника.

Отримані файли ще вимагатимуть визначеної редакції фахівцем у web-програмуванні, але пророблена робота дозволить істотно скоротити час остаточної доробки посібника.

Щоб перейти до використання електронного посібника спочатку необхідно скопіювати наступні файли та папки:

* каталог «Bin» необхідний для роботи зовнішнього модуля CKEditor;
* каталог «ckeditor»з файлами зовнішнього модуля CKEditor;
* каталог «Content» в якому розміщені всі сторінки веб-проекту (.aspx та .aspx.cs);
* каталог «Source», де розміщені всі файли дизайну електронного посібника;
* каталог «Uploads» в який будуть занесені всі файли завантажені користувачаси (викладачами перш за все).
* MasterPage.aspx та MasterPage.aspx.cs – головна сторінка-основа всіх сторінок електронного посібника.
* Файл web.config з налаштуваннями веб-проекту;

При цьому структура дерева папок не повинна бути зміненою чи пошкодженою, оскільки всі внутрішні гіпертекстові посилання мають строго визначену адресу.

Надзвичайна «гнучкість» ASP.NET забезпечує коректну роботу електронного посібника в будь-якому браузері.

Після завантаження на екрані з’явиться головна сторінка. Після завантаження домашньої сторінки можна обрати один і розділів із меню, що розташоване зліва. Щоб перейти до конкретного розділу, достатньо натиснути на ньому. Деякі із розділів містять список тем, чи перелік пунктів. Кожен із пунктів також є гіперпосиланням, і при натисненні відкриє ту чи іншу сторінку.

В першій версії електронного посібника був досить низький рівень захищеності від несанкціонованого доступу. В оновленій версії цей захист значно збільшився завдяки використанню сервеного коду, баз даних та розміщення електронного посібника на віддаленому сервері.

**Реєстрація в електронному посібнику**

Більшість матеріалу в електронному посібнику з дисципліни «Комп’ютерна логіка» доступна всім. Але є й матеріал, який можуть перегляадти лише зареєстровані та авторизовані користувачі. Для студентів це тести, а для викладачів – засоби адміністрування.

Сторінку реєстрації зображено на рисунку 3.1. Реєстрація проходить в декілька етапів:

* внесення даних в форму;
* перевірка даних сервером;
* підтвердження особи через електронну пошту;
* активація облікового запису.

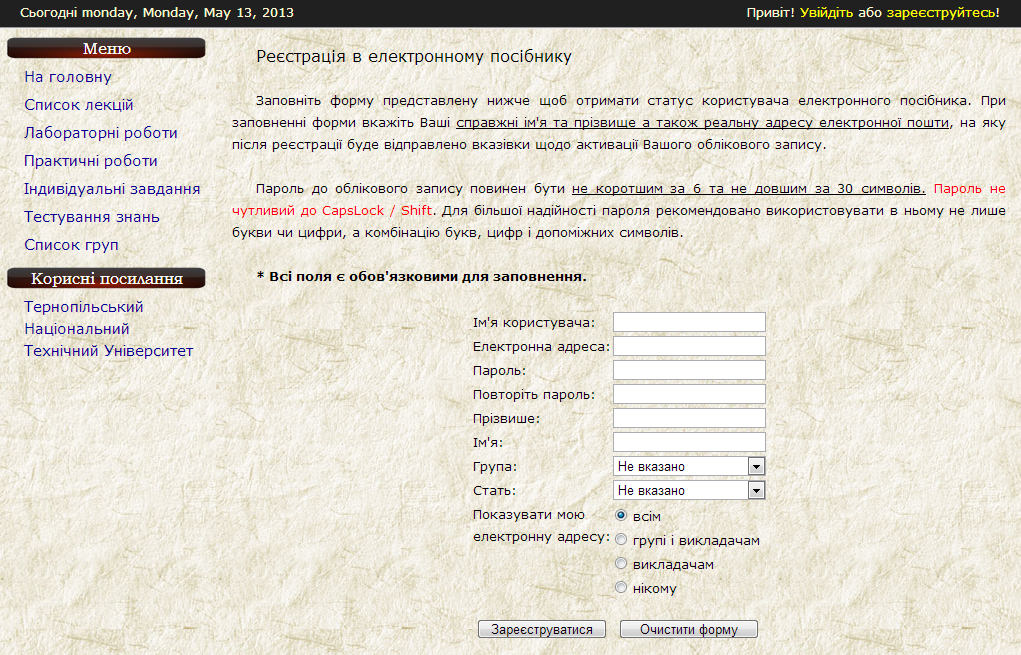


Рисунок 3.1 – Форма реєстрації в електронному посібнику

При перевірці даних форми перш за все перевіряється доступність вибраного користуваачем імені, введений пароль, присутність користувача з таким іменем та прізвищем і групою в базі даних. Якщо всі перевірки пройшли успішно користувач побачить повідомлення, в якому сказано що далі робити (рис. 3.2).

IMAGE HERE

**Авторизація та відновлення пароля**

Авторизуватись в системі користувач може за допомогою імені користувача та пароля від облікового запису (рис. 3.3).

Відновлення втраченого пароля відбувається в 3 етапи:

* на сторінці відновлення паролю користувач вказує свій обліковий запис;
* на електронну пошту, до якої прив’язаний цей обліковий запис відправляється повідомлення з динамічно згенерованим гіперпосиланням, в якому міститься унікальний ключ необхідний для відновлення пароля;
* перейшовши за посиланням користувач проходить перевірку унікального ключа і може вказати новий пароль до облікового запису.

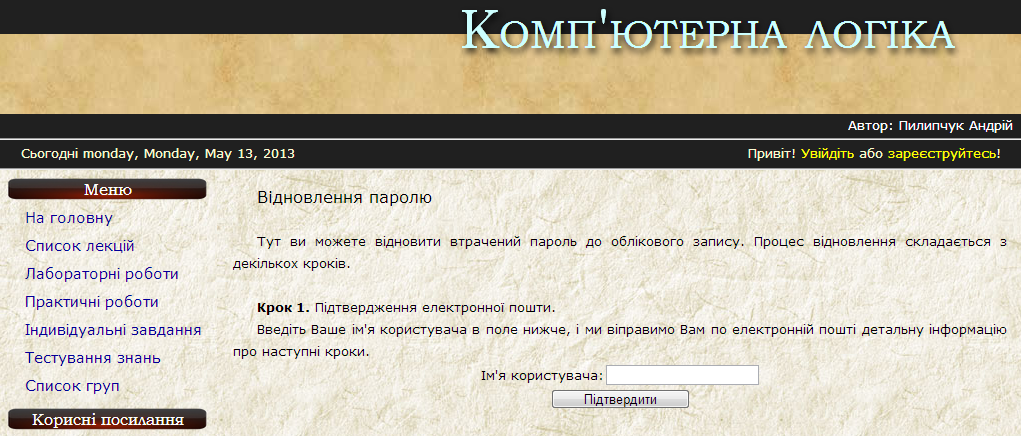


Рисунок 3.3 – Відновлення паролю (крок 1)

**Адміністрування електронного посібника**

В модернізованій версії електронного посібника присутні всі необхідні елементи для зручного та швидкого управління вмістом, наповненням та іншими параметрами електронного посібника. Адміністрування електронного посібника здійсгюється як з спеціальної сторінки (рис. 3.4) так і безпосередньо на сторінках посібниках, Авторизованому адміністратору (користувачу в групі «Викладачі») на сторінках електронного посібника доступні додаткові елементи управління. Таким чином редагування вмісту, виправлення помилок та доповнення вже існуючих сторінок новим матеріалом – справа декількох секунд. На рисунку 3.5 зображено редагування вмісту лекції.

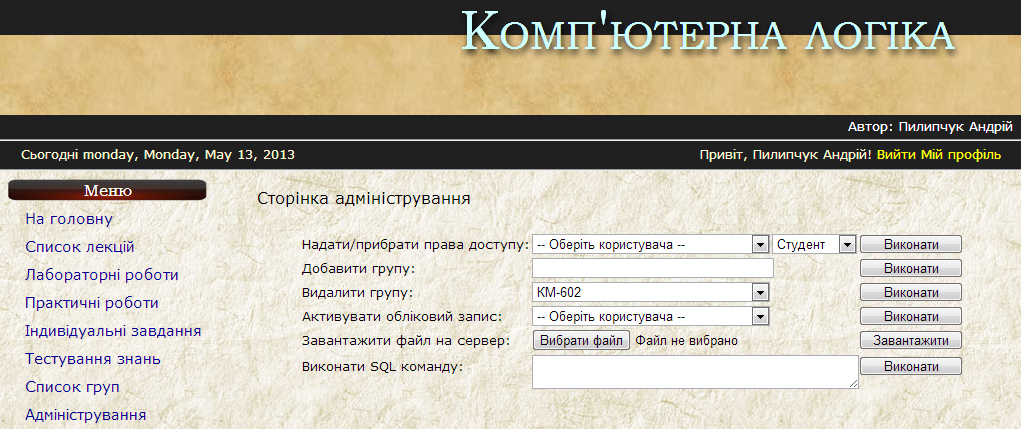


Рисунок 3.4 – Сторінка адміністрування

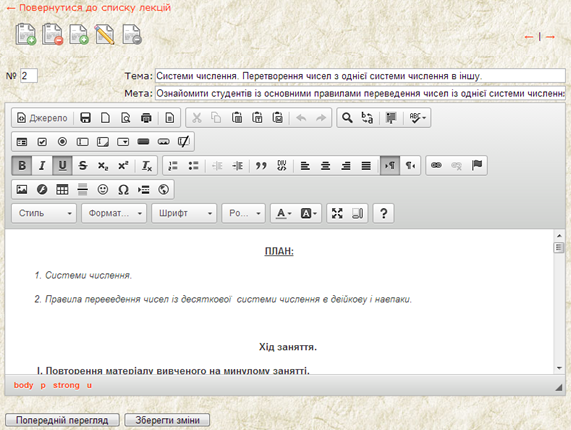


Рисунок 3.5 Релагування лекції

На сторінці адміністрування (доступна лише авторизованим користувачам групи «Викладачі», можна перейти через загальне меню) є декілька основних функцій:

* змінення рівня доступу користувача (студент/викладач);
* створення нової групи;
* видалення групи;
* активування облікового запису користувача;
* завантаження файлів на сервер;
* виконання SQL команди в базі даних.

**Створення тесту**

Окремим пунктом варто виділити створення тесту для перевірки знань.

На сторінці створення тесту (рис. 3.6) є форма, яку потрібно заповнити:

* назва (тема) тесту;
* кількість запитань (за 1 раз);
* ліміт часу для проходження тесту;

Далі потрібно заповнити форму самого тесту – запитання та варіанти відповіді. Можливі варіанти відповіді повинні бути поміщені в квадранті дужки, а номер правильної відповіді в фігурні. Наприклад:

[варіант 1] [варіант 2] [варіант 3] [варіант 4] {3}

В цьому вмпадку правильним буде [варіант 3].

При заповненні тесту допускається використання HTML тексту, для виділення слів, речень, вставлення зображень і так далі.

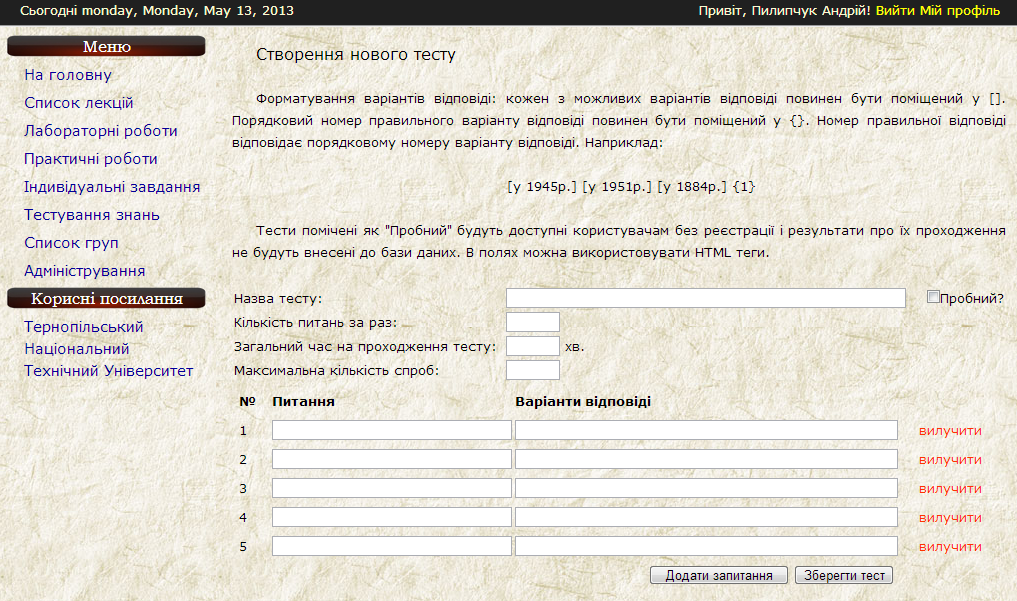


Рисунок 3.6 – Форма створення тесту

**4 ЕКОНОМІЧНИЙ РОЗДІЛ**

* 1. **Загальні положення**

Метою економічної частини дипломного проекту є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності електронного посібника з дисципліни «Комп’ютерна логіка» і прийняття рішення про її подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Для розрахунку вартості НДР необхідно виконати наступні етапи:

* Описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
* Визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
* Визначити суму матеріальних витрат;
* Обчислити витрати на електроенергію для науково-виробничих цілей;
* Розрахувати транспортні витрати;
* Нарахувати суму амортизаційних відрахувань;
* Визначити суму накладних витрат;
* Скласти кошторис та визначити собівартість НДР;
* Розрахувати ціну НДР;
* Визначити економічну ефективність та термін окупності продукту.
  1. **Визначення стадій технологічного процесу та загальної тривалості проведення НДР**

Для визначення загальної тривалості проведення НДР доцільно дані витрат часу по окремих операціях технологічного процесу звести в таблицю (див табл. 4.1).

Таблиця 4.1 – Середній час виконання НДР та стадії (операції) технологічного процесу

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Назва операції (стадії) | Виконавець | Середній час виконання операції, год. |
| 1. | Підготовка опису задачі | інженер | 3 |
| 2. | Розробка алгоритму рішення задачі | інженер | 9 |
| 3. | Розробка блок-схеми алгоритму | інженер | 4 |
| 4. | Розробка графічного дизайну посібника | інженер | 23 |
| 5. | Написання кодів електронного посібника | інженер | 66 |
| 6. | Тестування готового продукту | інженер | 7 |
| **Разом** | |  | 112 |

* 1. **Визначення витрат на оплату праці та відрахувань на соціальні заходи**

Відповідно до Закону України «Про оплату праці» заробітна плата – це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та допоміжної оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна заробітна плата розраховується за формулою:

Зосн. = Тс ∙ Кг, де

Тс – тарифна ставка, грн.;

Кг – кількість відпрацьованих годин.

Рекомендовані тарифні ставки: інженер – 10-12 грн./год.

Зосн. = 12 ∙ 112 = 1344 (грн.)

Додаткова заробітна плата становить 10-15% від основної зарплати.

Здод. = Зосн. ∙ Кдопл., де

Кдопл. – коефіцієнт додаткових виплат працівникам.

Здод. = 1344 ∙ 0,10 = 134,4 (грн.)

Звідси загальні витрати на оплату праці (Во.п.):

Во.п. = Зосн. + Здод.

Во.п. = 1344 + 134,4 = 1478,4 (грн.)

Крім цього, слід визначити відрахування на соціальні заходи (Вс.з.):

* відрахування в фонд страхування на випадок безробіття (1,6%);
* фонд по тимчасовій втраті безробіття (1,4%);
* пенсійний фонд (33,2%);
* внески на страхування від нещасного випадку на виробництві та професійного захворювання (1,4%).

Разом зазначені відрахування становлять 37,6%. Отже отримаємо:

Вс.з. = ФОП ∙ 0,376, де

ФОП – фонд оплати праці, грн.

Вс.з. = 1478,4 ∙ 0,376 = 555,8 (грн.)

Зведемо розрахунки витрат на оплату праці в таблицю 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Категорія працівн. | Основна заробітна плата | | | Дод. з/пл. грн. | Нарахув. до ФОП, грн. | Всього витрати на оплату праці, грн. |
| Тариф. ставка, грн. | К-сть відпрац. год. | Фактично нарах. з/пл., грн. |
| 1 | Інженер | 12 | 112 | 1344 | 134,4 | 555,8 | 2034,2 |
| **Разом** | | | 112 | 1344 | 134,4 | 555,8 | 2034,2 |

* 1. **Розрахунок витрат на електроенергію**

Затрати на електроенергію 1-ці обладнання визначається за формулою:

Зе = W ∙ T ∙ S, де

W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-годин електроенергії;

Підставивши в формулу значення W = 0,5 кВт, S = 0,77 грн., кількість годин затрачених на розробку – 112.

Зе = 0,5 ∙ 112 ∙ 0,77 = 43,12 (грн.)

* 1. **Розрахунок суми амортизаційних відрахувань**

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп’ютери та оргтехніка належать до четвертої групи основних фондів. Мінімально допустимі строки їх корисного використання 2 роки.

Для визначення амортизаційних відрахувань застосуємо формулу:

, де

А – амортизаційне відрахування за звітний період, грн.;

БВ – балансова вартість групи основних фондів на початок звітного періоду, грн.;

НА – норма амортизації, %;

Приймемо вартість ПК за БВ = 3840. Підставивши ці дані в формулу отримаємо:

* 1. **Обчислення накладних витрат**

Накладні витрати пов’язані із обслуговуванням виробництва та утриманням апарату управління підприємства (або фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб’єкта, накладні витрати можуть становити 20-60% від суми основної та додаткової заробітної плати працівників.

Нв = Во.п. ∙ 0,2…0.6.

За розрахунками проведеними раніше маємо, що Во.п. = 1478,4 грн. Приймемо, що накладні витрати становлять 40% від цієї суми і отримаємо:

Нв = 1478,4 ∙ 0,4 = 591,36 (грн.)

Зведемо кошторис витрат на розробку електронного посібника з дисципліни «Комп’ютерна логіка» в таблицю 4.3.

Таблиця 4.3 – Кошторис витрат на розробку електронного посібника

|  |  |  |
| --- | --- | --- |
| Зміст витрат | Сума, грн. | В % до загальної суми |
| Витрати на оплату праці | 1478,4 | 53,11 |
| Відрахування на соціальні заходи | 555,8 | 19,97 |
| Витрати на електроенергію | 43,12 | 1,55 |
| Амортизаційні відрахування | 114,69 | 4,12 |
| Накладні витрати | 591,36 | 21,25 |
| **Собівартість** | 2783,37 | 100 |

* 1. **Розрахунок ціни електронного посібника**

Ціну електронного посібника з дисципліни «Комп’ютерна логіка» можна обрахувати за наступною формулою:

, де

Ррен. – рівень рентабельності, 30%;

К – кількість замовлень, од.;

Вн.і. – вартість носія інформації, грн.;

ПДВ – ставка податку на додану вартість (20%).

Отже, підставивши значення в формулу отримаємо ціну електронного посібника з дисципліни «Комп’ютерна логіка»:

* 1. **Визначення економічної ефективності і терміну окупності капітальних вкладень**

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві (фірмі) за певний проміжок часу.

Економічна ефективність (Ер) полягає у відношенні результату виробництва до затрачених ресурсів:

, де

П – прибуток, грн.;

Св – собівартість, грн.

Підставимо значення в формулу визначення економічної ефективності:

З цього випливає, що термін окупності капіталовкладень становить:

.

Таблиця 4.4 – Результати техніко-економічних показників

|  |  |  |
| --- | --- | --- |
| № п/п | Показник | Значення |
| 1. | Собівартість, грн. | 2783,37 |
| 2. | Плановий прибуток, грн. | 1002,02 |
| 3. | Ціна, грн. | 4342,06 |
| 4. | Економічна ефективність | 0,36 |
| 5. | Термін окупності, рік | 2,8 |

Підсумовуючи всі проведені вище розрахунки можна сказати, що розробка електронного посібника з дисципліни «Комп’ютерна логіка» є економічно доцільною і має допустимий термін окупності, а затрати на створення такого посібника є порівняно не великими.

1. **ОХОРОНА ПРАЦІ ТЕХНІКИ БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ**
   1. **Вимоги до виробничих приміщень**

Облаштування робочих місць, обладнаних відео терміналами, оргтехнікою, повинно забезпечувати:

* належні умови освітлення приміщення і робочого місця, відсутність відблисків;
* оптимальні параметри мікроклімату (температура, відносна вологість, швидкість руху та рівень іонізації повітря);
* належні ергономічні характеристики основних елементів робочого місця.

А також вжити заходи проти таких небезпечних і шкідливих чинників:

* наявність шуму та вібрації;
* м’яке рентгенівське випромінювання;
* електромагнітне випромінювання;
* ультрафіолетове та інфрачервоне випромінювання;
* електростатичне поле між екраном монітору та оператором;
* наявність пилу, озону, оксидів азоту й аероіонізації.

Будівлі, і ті їх частини, в яких розташовуються ЕОМ, повинні мати не нижче ІІ ступеня вогнестійкості. Неприпустимим є розташування приміщень виробництв з мокрими технологічними процесами поряд з приміщеннями, де розташовуються ЕОМ, виконується їх обслуговування, налагодження і ремонт, а також над такими приміщеннями або під ними.

Службові приміщення, в яких розташовані ЕОМ, не повинні межувати з приміщеннями, де рівні шуму та вібрації перевищують норму (механічні цехи, майстерні тощо).

Робочі місця з відеотерміналами або персональними ЕОМ у приміщеннях з джерелами шкідливих виробничих чинників повинні розміщуватися в ізольованих кабінах з обладнаним повітрообміном.

Стіни кабін виготовляються з негорючих матеріалів. Дозволяється виготовляти їх зі скла та металевих! конструкцій. У кабіні мусить бути оглядове вікно (вінка). Висота оглядового вікна має бути не менше 1,5м., а відстань від підлоги не більше 0,8м.

Відповідно до ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», затверджених МОЗ України 10.12.98, є неприпустимим розташування приміщень для роботи з відеотерміналами та ЕОМ у підвалах та цокольних поверхах.

Площу приміщень, у яких розташовують відеотермінали визначають згідно з чинними нормативними документами, виходячи з розрахунку на одне робоче місце, обладнане відеотерміналом: площа – не менше 6,0 кв. м., об’єм – не менше 20,0 куб. м., з урахуванням максимальної кількості осіб, які одночасно працюють у зміні.

Стіни, стеля, підлога приміщень, де розміщені ЕОМ, повинні виготовлятися з матеріалів, дозволених для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду.

Обслуговування, ремонт та налагодження ЕОМ, вузлів та блоків ЕОМ слід виконувати в окремому приміщенні (майстерні).

Підлога всієї зони обслуговування, ремонту та налагодження ЕОМ, вузлів та блоків ЕОМ має бути вкрита діелектричними килимками, термін використання яких після їх випробування на електричну міцність не закінчився, або викладена ізольованими підстилками (шириною не менше, ніж 0,75-08м.) для ніг.

Приміщення комп’ютерних класів (залів) у яких проводиться навчання на ЕОМ, повинні мати суміжне приміщення (лаборантське) площею не менше 18 км. м. з двома входами: в навчальне приміщення та в коридор (на сходову клітку).

Заземлені конструкції, що знаходяться в приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику.

У приміщеннях з ЕОМ слід щоденно проводити вологе прибирання, повинні бути медичні аптечки першої допомоги.

Приміщення з ЕОМ, крім приміщень, у яких розміщуються ЕОМ типу ЕС, СМ та інші великі ЕОМ загального призначення, повинні бути оснащені системою автоматичної пожежної сигналізації.

Підходи до засобів пожежогасіння повинні бути вільними.

Умови праці осіб, які працюють з ЕОМ, повинні відповідати І або ІІ класу згідно з Гігієнічною класифікацією праці за показниками шкідливості та небезпечності чинників виробничого середовища, важкості та напруженості трудового процесу № 4137-86, затвердженого МОЗ СРСР 12.08.86.

* 1. **Вимоги до рівнів шуму та вібрації**

Для забезпечення нормованих рівнів шуму у виробничих приміщеннях та на робочих місцях застосовуються шумопоглинальні засоби, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками.

Як засоби шумопоглинання повинні застосовуватися негорючі або, важкогорючі спеціальні перфоровані плити, панелі, мінеральна вата з максимальним коефіцієнтом звукопоглинання в межах частот 31,5-8000 Гц, або інші матеріали аналогічного призначення, дозволені для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду. Крім того, необхідно застосовувати підвісні стелі з аналогічними властивостями.

Приміщення з ЕОМ повинні бути обладнані системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією відповідно до СНиП 2.04.05-91 «Отопление вентиляция и кондиционирование».

Параметри мікроклімату, іонного складу повітря, вміст шкідливих речовин на робочих місцях, оснащених відеотерміналами, повинні відповідати вимогам пункту 2.4 СН 4088-86 «Санітарні норми мікроклімату виробничих приміщень», затверджених МОЗ СРСР, ГОСТ 12.1.005-88 «ССБТ Общие санитарно-епидемиологические требования к воздуху рабочей зоны» продемонстровано в табл. 5.1, СН 2152-80 «Санітарно-гігієнічні норми допустимих рівнів іонізації повітря виробничих та громадських приміщень», затверджених МОЗ СРСР показано в табл. 5.2.

Таблиця 5.1 – Нормовані параметри мікроклімату для приміщень з ВДТ та ПЕОМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Пора року | Категорія робіт згідно з ГОСТ | Температура повітря, °С | Відносна вологість повітря, % | Швидкість руху повітря, м/с |
|  | 005-88 | оптимальна | оптимальна | Оптимальна |
| Холодна | легка-1 а | 22-24 | 40-60 | 0,1 |
|  | легка-1 б | 21-23 | 40-60 | 0,1 |
| Тепла | легка-1 а | 23-25 | 40-60 | 0,1 |
|  | легка-1 б | 22-24 | 40-60 | 0,2 |

Таблиця 5.2 – Рівні іонізації повітря приміщень при роботі на ВДТ та ПЕОМ (відповідно до СН 2152-80)

|  |  |  |
| --- | --- | --- |
| Рівні | Кількість іонів в 1 см. куб. повітря | |
| n | n” |
| Мінімально необхідний | 400 | 600 |
| Оптимальний | 1500-3000 | 3000-5000 |
| Максимально допустимий | 50 000 | 50 000 |

Для підтримки допустимих значень мікроклімату та концентрації позитивних та негативних іонів необхідно передбачити установки або прилади зволоження та/або штучної фонації, кондиціонування повітря.

* 1. **Вимоги до організації робочого місця користувача ЕОМ**

Організація робочого місця користувача відеотерміналу та ЕОМ повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ГОСТ 12.2.032 «ССБТ Рабочее место при выполнении работ сидя. Общие эргономические требования»; характеру та особливостям трудової діяльності.

Площа, виділена для одного робочого місця з відеотерміналом або персональною ЕОМ, повинна складати не менше 6 кв. м., а об’єм – не менше 20 куб. м.

Робочі місця з відеотерміналами відносно світлових прорізів повинні розмішуватися так, щоб природне світло падало збоку, переважно зліва.

При розміщенні робочих місць з відеотерміналами та персональними ЕОМ необхідно дотримуватись таких вимог:

* Робочі місця з відеотерміналами та персональними ЕОМ розміщуються на відстані не менше 1м від стін зі світловими прорізами;
* Відстань між бічними поверхнями відеотерміналів має бути не меншою за 1,2м;
* Відстань між тильною поверхнею одного відеотерміналу та екраном іншого не повинна бути меншою 2,5м;
* Прохід між рядами робочих місць має бути не меншим 1м.

Вимоги щодо відстані між бічними поверхнями відеотерміналів та відстані між тильною поверхнею одного відеотерміналу та екраном іншого враховуються також при розміщенні робочих місць із відеотерміналами та персональними ЕОМ в суміжних кімнатах, з урахуванням конструктивних особливостей стін та перегородок.

Конструкція робочого місця користувача відеотерміналу (при роботі сидячи) має забезпечувати підтримання оптимальної робочої пози з такими ергономічними характеристиками: ступні ніг – на підлозі або на підставці для ніг; стегна – в горизонтальній площині; передпліччя – вертикально; лікті – під кутом 70-90° до вертикальної площини; зап’ястя зігнуті під кутом не більше 20° відносно горизонтальної площини; нахил голови – 15-20° відносно вертикальної площини.

Якщо користування відеотерміналом та персональною ЕОМ є основним видом діяльності, то вказане обладнання розміщується на основному робочому столі, як правило, з лівого боку.

Якщо використання відеотерміналу та персональної ЕОМ є періодичним, то устаткування, як правило, розміщується на приставному столі, переважно з лівого боку від основного робочого столу. Кут між поздовжніми осями основного та приставного столів має бути 90-140°.

При періодичному використанні відеотерміналів та персональних ЕОМ дозволяється обладнувати в приміщенні, що відповідає вимогам даних Правил, окремі робочі місця колективного користування з відеотерміналом та персональною ЕОМ.

Висота робочої поверхні столу для відеотерміналу має бути в межах 680-800мм, а ширина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля.

Рекомендовані розміри столу: висота – 725мм, ширина – 600-1400мм, глибина – 800-1000мм.

Робочий стіл для відеотерміналу повинен мати простір для ніг висотою не менше 600мм., шириною не менше 500мм., глибиною на рівні колін – не менше 650мм.

Робочий стіл для відеотерміналу, як правило, має бути обладнаним підставкою для ніг шириною не менше 300мм. Та глибиною не менше 400мм., з можливістю регулювання по висоті в межах 150мм. Та кута нахилу опорної поверхні – в межах 20°. Підставка повинна мати рифлену поверхню та бортик на передньому краї заввишки 10мм. Робоче сидіння (сидіння, стілець, крісло) користувача відеотерміналу чи персональної ЕОМ повинно мати такі основні елементи: сидіння, спинку та стаціонарні або знімні підлокітники.

У конструкцію сидіння можуть бути введені додаткові елементи, що не є обов’язковими: підголівник та підставка для ніг.

Робоче сидіння користувача відеотерміналу чи персональної ЕОМ повинно бути підйомно-поворотним, таким, що регулюється за висотою, кутом нахилу сидіння та спинки, за відстанню спинки до переднього краю сидіння, висотою підлокітників.

Регулювання кожного параметра має бути незалежним, плавним або ступінчастим, мати надійну фіксацію.

Хід ступінчастого регулювання елементів сидіння має становити для лінійних розмірів – 15-20мм, для кутових – 2-5°.

Зусилля під час регулювання не повинні перевищувати 20 Н.

Ширина та глибина сидіння повинні бути не меншими за 400мм. Висота поверхні сидіння має регулюватися в межах 400-500мм., а кут нахилу поверхні – від 15° вперед до 5° назад.

Поверхня сидіння має бути плоскою, передній край – заокругленим.

Висота спинки сидіння має становити 300х20мм., ширина – не менше 380мм., а радіус кривизни в горизонтальній площині – 400мм.

Кут нахилу спинки повинен регулюватися в межах 0-30° відносно вертикального положення. Відстань від спинки до переднього краю сидіння повинна регулюватись у межах 260-400мм.

Для зниження статичного напруження м’язів рук необхідно застосовувати стаціонарні або знімні підлокітники довжиною не менше 250мм., шириною – 50-70мм., що регулюються по висоті над сидінням у межах 230±30мм. Та по відстані між підлокітниками в межах 350-500мм.

Поверхня сидіння, спинки та підлокітників має бути напівм’якою з неслизьким, ненаелектризовуючим, повітронепроникним покриттям та забезпечувати можливість чищення від бруду.

Екран відеотерміналу та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм., з урахуванням розміру алфавітно-цифрових знаків та символів.

Розташування екрану відеотерміналу має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30° від лінії зору працівника.

Клавіатуру слід розміщувати на поверхні столу або на спеціальний робочій поверхні окремо від столу на відстані 100-300 мм. Від краю, ближнього до працівника. Кут нахилу клавіатури має бути в межах 5-15°.

Робоче місце з відеотерміналом слід оснащувати пюпітром (тримачем) для документів, що легко переміщується.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрану відеотерміналу, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля: по висоті 900-1300мм., по глибині 400-500мм.

* 1. **Розрахунок освітлення**

Проведемо розрахунок освітлення дільниці монтажу та складання плати субблоку автоматики аналогового.

Розрахунок штучного робочого освітлення проведемо за методикою згідно [Л5]. Застосуємо метод коефіцієнта використання світлового потоку, який застосовується для розрахунку загального рівномірного освітлення приміщень при відсутності вагомих затемнень робочої поверхні.

Загальне освітлення розраховуємо по формулі:

де: ЕН – нормативна мінімальна освітленість, яка становить, згідно табл. 12.1 [Л5] для виробництв з високою точністю зорової роботи і середнім контрастом об’єкта розрізнення на фоні середньої світлості, ЕН = 300 лк.;

КЗ – коефіцієнт запасу, приймемо КЗ = 1,5;

S – площа дільниці, згідно (3.2.7) розділу 3 S = 24 м2;

Z – коефіцієнт нерівномірності освітлення, вибираємо згідно Z = 1,3;

h – коефіцієнт використання світлового потоку, h = 0,68.

Для освітлення дільниці приймемо люмінесцентні лампи типу ЛБЦ потужністю 40 Вт, які мають номінальний світловий потік Fл = 3039 лм.

Необхідну кількість ламп розраховуємо за формулою:

Приймемо кількість ламп за 7.

Розрахуємо кількість світильників по формулі:

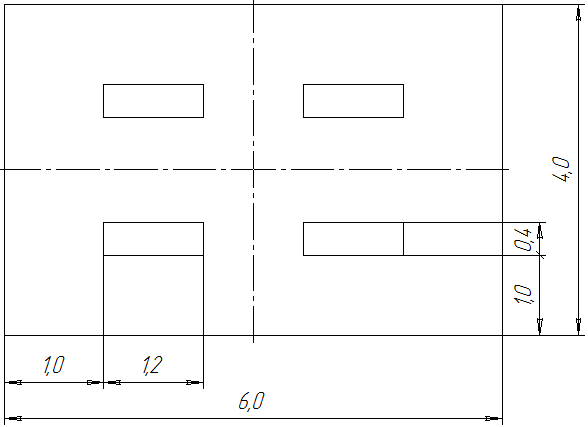
де n – кількість ламп у вибраному типі світильника.

Використаємо на дільниці світильники типу ЛПО-01В-2x40, в яких використовується по дві люмінесцентні лампи.

Розрахуємо необхідну кількість світильників:

(шт.)

Приймаємо кількість світильників NС ≈ 4 шт.



* 1. **Заходи з покращення умов праці і техніки безпеки на дільниці**

Одним із основних завдань щодо покращення умов праці є збереження і зміцнення здоров’я працівників, підвищенням працездатності і тривалості життя, а також задоволеність працею.

Виконання цього завдання забезпечується, перш за все, дотриманням правил техніки безпеки на робочих місцях. Організація техніки безпеки на потрібному рівні можлива тільки при умові постійного режимного аналізу виробничої обстановки на основі результатів проведення вимірів параметрів шкідливих виробничих факторів, засобів захисту працівників. Усе це містить Паспорт санітарно-технічного стану цеху, який допомагає отримувати інформацію для повної характеристики виробництва.

Для забезпечення систематичного покращення умов праці на дільниці необхідно намітити комплекс заходів і проводити їх в плановому порядку. Для забезпечення покращення умов праці та попередження виробничого травматизму і захворюваності, в план робіт по охороні праці мають бути включені такі заходи:

* модернізація виробничого обладнання;
* впровадження автоматичного і дистанційного управління виробничими процесами;
* впровадження систем автоматичного контролю і сигналізації про наявність небезпечних і шкідливих виробничих факторів;
* впровадження блокуючих пристроїв, які забезпечують аварійне відключення виробничого обладнання у випадку його несправності;
* впровадження технічних пристроїв, які забезпечують захист працюючих від враження електричним струмом;
* встановлення запобіжних і захисних пристосувань та технологічному обладнанні.

Важливим заходом також є атестація робочих місць на відповідність до вимог і норм охорони праці, яка має на меті визначення стану технічної безпеки, санітарно-гігієнічних умов. Для покращення умов праці робітників дуже ефективним є проведення ряду заходів, які зменшують втомлюваність, відповідно підвищують продуктивність праці. На підприємствах повинні організовуватись кабінети професійної діагностики, де проводиться професійний відбір на роботу з урахуванням психофізичних, психологічних, фізіологічних особливостей. Одним з найважливіших заходів є раціональне харчування працівників, а також естетичне оформлення виробничого приміщення, робочого місця.

Суттєвим показником виробничого середовища є і сприятливий психологічний клімат, від чого в значній мірі залежить емоційний стан працівників, робочий настрій, при покращенні якого збільшується продуктивність праці, зменшується небезпека травматизму.

**ВИСНОВОК**

Використання електронних посібників як сучасних інформаційних технологій в освіті дозволить внести суттєві зміни в процес навчання.

Аналіз питань використання сучасних засобів показує, що розробка електронних посібників в мережі Інтернет розвивається доволі повільно. В дипломному проекті запропонована розробка електронного посібника, як універсального засобу для організації навчання, в ТК ТНТУ та мережі Інтернет.

Електронний посібник відносяться до достатньо складних програмних продуктів і потребують для реалізації не тільки засобів комп’ютерної графіки, але й спеціалізованих інструментальних засобів розробки.

На основі проведеного порівняльного аналізу авторських систем, в якості інструментальних засобів практичної реалізації дипломного проекту, розробку електронного посібника використано HTML та CSS стилі, об’єктно-орієнтоване програмування ASP.NET C# а також додатково JavaScript та Java.

Крім цього, в економічній частині дипломного проекту виконано розрахунок економічної ефективності за рік, термін окупності такого електронного посібника, ціну та прибуток, який можна отримати від розробки такого проекту.

Слід відмітити, що електронний посібник може бути корисний не тільки для студентів комп’ютерних спеціальностей, але для інших спеціальностей, які вивчають аналогічні предмети, що дозволяє додатково заощадити кошти.

Результати дипломного проекту в розділі загальні вимоги до розробки електронного посібника можуть використовуватися як база для розробки електронних посібників різних напрямків.

Практична розробка – це електронний посібник для студентів комп’ютерних спеціальностей задуманий як універсальна методична розробка для студентів спеціальності 5.091504 «Обслуговування комп’ютерних та інтелектуальних систем та мереж» при вивченні широкого спектру питань комп’ютерної логіки і не тільки.

Електронний посібник розробляється як відкрита система, яка в подальшому дає змогу доповняти її новими розділами, а також вносити необхідні зміни.

Використання такого посібника в навчальному процесі дозволить підвищити ефективність навчання студентів ТК ТНТУ, а також значно зменшити час використання Інтернет для пошуку інформації з широкого спектру питань при вивченні достатньої кількості питань курсу.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Посібник користувача ПК. Друге видання. Тхір Л.І., Калушка В.П., Юзьків А.В.
2. Шевченко А.I. Методика розробки електронних підручників для загальноосвітніх середніх шкіл України. — Донецьк, 2001
3. Зарецька І.Т. та інші, Інформатика, ч. 2. — К., Київ — Форум, 2004
4. Гаєвський О.Ю. Інформатика, 7–11 клас. — К: «Видавництво А.С.К.», 2003
5. Глинський Я.М. Інформатика, ч. 2. — Львів, 2004
6. Гуржій А.М., Жук Ю.О., Волинський В.П. Засоби навчання: Навчальний посібник для студентів вищих педагогічних навчальних закладів та слухачів системи післядипломної освіти; АПН України, Інститут педагогіки. — К., 1997.
7. http://www.zippo.net.ua
8. http://www.rusnauka.com
9. http://www.umniki.com.ua
10. http://www.confcontact.com
11. http://www.chpal.edukit.ck.ua
12. http://www.w3schools.com
13. http://www.htmlbook.ru