# ЗМІСТ

*Змін.*

*Аркуш*

*№ докум.*

*Підпис*

*Дата*

*Розробив*

*Перевірив*

*Затвер.*

*Н.контр.*

*Літ*

*Аркуш*

*Аркушів*

Н

1

Стрілець Д.А.

Гуменна В.В.

Т.П.ПР.151.09.00 ПЗ

ВСТУП 4

1 Структура підприємства 6

2 Організація робочого місця 6

2.1 Робоче місце 6

2.2 Опис власного робочого місця 7

3 Техніка безпеки на робочому місці 10

3.1 Інструктаж та загальні вимоги 10

3.2 Освітлення 12

3.3 Мікроклімат 13

3.4 Рівень шуму та вібрації 13

3.5 Електробезпека 14

4 Опис технологічного процесу 16

4.1 Технологічний процес на підприємстві 16

4.2 Технологічний процес на робочому місці 16

4.2 Застосування прогресивних технологій 17

5 Індивідуальне завдання 21

5.1 Постановка завдання 21

5.2 Короткий опис індивідуального завдання 21

5.3 Опис вхідних даних 22

5.4 Опис вихідних даних 22

5.5 Реалізація та результати 22

ВИСНОВОК 45

ДЖЕРЕЛА 46

ХПК НУ «ЛП»

**Вступ**

Технологічна практика студентів є важливою складовою частиною навчального процесу. Вона дозволяє студентам набути нових знань та умінь для майбутньої професії. Коло задач, які розв’язують під час практики, досить велике. Воно охоплює питання, пов’язані з проектуванням і впровадженням технологічних процесів, вивченням стандартів і інструкцій, сучасного програмного забезпечення тощо.

Практика студентів надає можливість реального (практичного) набуття знань на підприємствах. Значення обсягу освітніх послуг, які надаються в період практики, оцінюється ресурсом часу практики і регламентується навчальними планами.

Практика – один із найбільш ресурсомістких за часом та інтегральних за формою і суттю видів навчання. Це підкреслює і відображає важливу складову освітньої стратегії коледжу – забезпечення цілісності і нерозривності теоретичного навчання та практичної професійної підготовки майбутніх фахівців.

**1 Структура підприємства**

*Веб-програмування*— галузь веб-розробки і різновид дизайну, в завдання якої входить проектування користувальницьких веб-інтерфейсів для сайтів або веб-додатків. Веб-дизайнери проектують логічну структуру веб-сторінок, продумують найбільш зручні рішення подачі інформації, а також займаються художнім оформленням веб-проекту. В результаті перетину двох галузей людської діяльності грамотний веб-дизайнер повинен бути знайомий з останніми веб-технологіями і володіти відповідними художніми якостями.

Web Systems Solutions – це ІТ компанія яка займається розробкою інтернет рішень.

Наразі вона має офіси у 2 країнах:

У Польщі за адресою Poland, Wrocław, Powstańców Śląskich 95, 53-332.

Та в Україні за адресою м. Кам’янець-Подільский, вул. Князів Коріатовичів 1 та м. Хмельницьий, вул. Кам’янецька 137, де я й проходив технологічну практику.

Офісу Web Systems Solutions у Хмельницькому вже 9 років.

Наразі команда Web Systems Solutions у місті Хмельницькому містить близько 60 працівників: розробників, менеджерів, маркетологів. Цією командою розроблено безліч проектів, наприклад: Velvet Armor System (українська компанія-виробник військової продукції та сільськогосподарської техніки).

Команда Web Systems Solutions розробляє сайти адаптивними під усі типи пристроїв, та з можливістю зміни мов.

Також вони зайняли друге місце у ТОП-10 розробників інтернет-магазинів на CMS OpenCart за 2016 рік. Робота у сфері електронної комерції їх пріоритетний напрямок.

**2 Організація робочого місця**

**2.1 Робоче місце**

Головними елементами робочого місця програміста є письмовий стіл і крісло. Основним робочим положенням є положення сидячи. Робоче місце для виконання робіт у положенні сидячи організується відповідно до стандартів.

Робоча поза сидячи викликає мінімальне стомлення програміста. Раціональна планування робочого місця передбачає чіткий порядок і сталість розміщення предметів, засобів праці і документації. Те, що потрібно для виконання робіт частіше, розташовано в зоні легкої досяжності робочого простору.

Моторне поле - простір робочого місця, в якому можуть здійснюватися рухові дії людини.

Максимальна зона досяжності рук - це частина моторного поля робочого місця, обмеженого дугами, описуваними максимально витягнутими руками при русі їх в плечовому суглобі.

Оптимальна зона - частина моторного поля робочого місця, обмеженого дугами, описуваними передпліччями при русі в ліктьових суглобах з опорою в точці ліктя і з відносно нерухомим плечем.

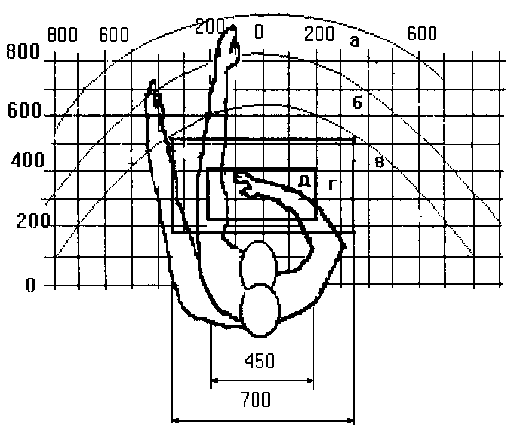


Рисунок 1 – Зони досяжності рук у горизонтальній площині

Зони досяжності рук у горизонтальній площині:

* А - зона максимальної досяжності;
* Б - зона досяжності пальців при витягнутій руці;
* В - зона легкої досяжності долоні;
* Г - оптимальний простір для грубої ручної роботи;
* Д - оптимальний простір для тонкої ручної роботи.

Розглянемо оптимальне розміщення предметів праці і документації в зонах досяжності рук:

* Дисплей розміщується в зоні а (в центрі);
* Клавіатура - в зоні г / д;
* Системний блок розміщається в зоні б (зліва);
* Принтер знаходиться в зоні а (справа);
* Документація (література і документація, необхідна при роботі) в зоні легкої досяжності долоні - в (ліворуч);
* Література, яка не постійно використовуваною в висувних ящиках столу.

При проектуванні письмового столу варто враховувати наступне:

* Висота столу повинна бути вибрана з урахуванням можливості сидіти вільно, в зручній позі, при необхідності спираючись на підлокітники;
* Нижня частина столу повинна бути сконструйована так, щоб програміст міг зручно сидіти, не був змушений підбирати ноги;
* Поверхня столу повинна мати властивості, що виключають появу відблисків в полі зору програміста;
* Конструкція столу повинна передбачати наявність висувних ящиків (не менше 3 для зберігання документації, лістингів, канцелярських приналежностей, особистих речей).

Параметри робочого місця вибираються відповідно до антропометричних характеристик.

При використанні цих даних у розрахунках варто виходити з максимальних антропометричних характеристик (М +2).

При роботі в положенні сидячи рекомендуються такі параметри робочого простору:

* Ширина не менше 700 мм;
* Глибина не менше 400 мм;
* Висота робочої поверхні столу над підлогою 700-750 мм.

Оптимальними розмірами столу є:

* Висота 710 мм;
* Довжина столу 1300 мм;
* Ширина столу 650 мм.

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, позитивно впливає на продуктивність праці. Забарвлення приміщень та меблів має сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою. У службових приміщеннях, у яких виконується одноманітна розумова робота, що вимагає значної нервової напруги і великого зосередження, забарвлення повинне бути спокійних тонів - малонасичені відтінки холодного зеленого або блакитного кольорів.

При розробці оптимальних умов праці програміста необхідно враховувати освітленість , шум і мікроклімат.

Раціональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин:

* Недостатність освітленості
* Надмірна освітленість;
* Неправильний напрямок світла.

Недостатність освітлення приводить до напруги зору, послаблює увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає осліплення, роздратування і різь в очах. Неправильний напрям світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть привести до нещасного випадку або профзахворювань, тому настільки важливий правильний розрахунок освітленості.

Розрахунок освітленості робочого місця зводиться до вибору системи освітлення, визначенню необхідного числа світильників, їхнього типу і розміщення.

**2.2 Опис власного робочого місця**

Конфігурація робочої машини (персонального комп’ютера), яка використовувалась в процесі проходження практики, ноутбук DESKTOP-TOFPII6 7587:

* оперативна пам'ять 3 Гб;
* процесор AMD E-350 1.6 GHz;
* відеокарта AMD Radeon HD 6310 Graphics;
* дисплей 15.6" (1366\*768).

Встановлене програмне забезпечення та операційна система:

* операційна система: Windows 10;
* JetBrains PhpStorm;
* WAMP;
* Visual Studio Code;
* Opera Internet Browser;
* Open Server;
* Sublime Text 3;
* MySQl Server.

На рисунку 3 представлена схема робочого місця:

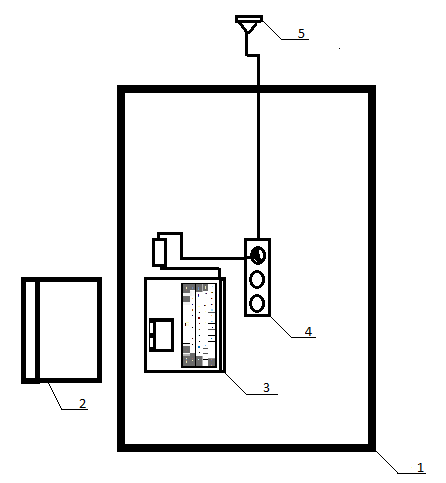


Рисунок 2 – Схема робочого місця.

На рисунку 3 зображено схему мого робочого місця на якому зображено мій робочий стіл (1), стілець(2), комп’ютер (3), мережевий фільтр для підзарядки та живлення електронних пристроїв та робочої машини (4), розетка 220V(5).

Робоче місце відповідає усім вимогам для забезпечення мікроклімату а саме: об’єм його становить 20 м3 та площа становить 6м2. Штучне освітлення виконується за допомогою електричних джерел світла: люмінесцентних ламп. Використовуються саме люмінесцентні лампи, які в порівнянні з лампами розжарювання мають суттєві переваги:

* По спектральному складу світла вони близькі до денного, природного освітлення;
* Мають підвищену світловіддачу (в 3-4 рази вище, ніж у ламп розжарювання);
* Більш тривалий термін служби.

Рівень шуму не перевищує 50 дБА, монітор розсташований на безпечній відстані 600-800мм, а кут зору на екран становив 10-15 градусів.

**3.Техніка безпеки на робочому місці**

**3.1 Інструктаж та загальні вимоги**

Інструктажі з охорони праці — один із найбільш ефективних видів навчання з питань безпечного виконання робіт.

Працівники, під час прийняття на роботу та періодично, повинні проходити на підприємстві інструктажі з питань охорони праці, надання першої допомоги потерпілим від нещасних випадків, а також з правил поведінки та дій при виникненні аварійних ситуацій, пожеж і стихійних лих.

Види інструктажів з охорони праці:

За характером і часом проведення інструктажі з питань охорони праці поділяються на вступний, первинний, повторний, позаплановий та цільовий.

Вступний інструктаж проводиться спеціалістом служби охорони праці або іншим фахівцем відповідно до [наказу (розпорядження)](http://oppb.com.ua/news/yak-pidgotuvaty-nakaz-pro-pidgotovku-do-navchannya-ta-perevirku-znan-iz-ohorony-praci) по підприємству, який в установленому типовим положенням порядку пройшов навчання і перевірку знань з питань охорони праці.

Запис про проведення вступного інструктажу робиться в [Журналі реєстрації вступного інструктажу з питань охорони праці](https://mediapro.com.ua/product/zurnal-reestracii-vstupnogo-instruktazu-z-pitan-ohoroni-praci), який зберігається службою охорони праці або працівником, що відповідає за проведення вступного інструктажу, а також у наказі про прийняття працівника на роботу. Журнал реєстрації вступного інструктажу є документом постійного зберігання.

Первинний інструктаж проводиться з працівником до початку роботи безпосередньо на робочому місці.

Повторний інструктаж проводиться на робочому місці індивідуально з кожним працівником або групою працівників, які виконують однотипні роботи, за обсягом і змістом переліку питань первинного інструктажу. Повторний інструктаж проводиться в терміни, визначені нормативно-правовими актами з охорони праці, не рідше:

* на роботах з підвищеною небезпекою — 1 раз на 3 місяці;
* для решти робіт — 1 раз на 6 місяців.

Позаплановий інструктаж проводиться з працівниками на робочому місці або в кабінеті охорони праці:

* при введенні в дію нових або переглянутих нормативно-правових актів з охорони праці, а також при внесенні змін та доповнень до них;  
  при зміні технологічного процесу, заміні або модернізації устаткування, приладів та інструментів, вихідної сировини, матеріалів та інших факторів, що впливають на стан охорони праці;
* при порушенні працівниками вимог нормативно-правових актів з охорони праці, що призвели до травм, аварій, пожеж тощо;
* при перерві в роботі виконавця робіт більш ніж на 30 календарних днів — для робіт з підвищеною небезпекою, а для решти робіт — понад 60 днів.

Цільовий інструктаж проводиться з працівниками:

* при ліквідації аварії або стихійного лиха;
* при проведенні робіт, на які відповідно до законодавства оформлюються наряд-допуск, наказ або розпорядження.

Первинний, повторний, позаплановий і цільовий інструктажі проводить безпосередній керівник робіт (начальник структурного підрозділу, майстер) або фізична особа, яка використовує найману працю.

Первинний, повторний, позаплановий і цільовий інструктажі завершуються перевіркою знань у вигляді усного опитування або за допомогою технічних засобів, а також перевіркою набутих навичок безпечних методів праці, особою, яка проводила інструктаж.

Інструкція розроблена на основі ДНАОП 0.00-8.03-93 “Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві”, ДНАОП 0.00-4.15-98 “Положення про розробку інструкцій з охорони праці”, ДНАОП 0.00-4.12-99 “Типове положення про навчання з питань охорони праці”.

Ця інструкція встановлює загальні вимоги з  охорони праці  для програміста.

Усі роботи, які проводяться програмістом, повинні виконуватися відповідно до цієї інструкції.

Робочим місцем протягом усієї робочої зміни для програміста є спеціально обладнане місце.

До виконання робіт допускаються особи, які пройшли навчання, стажування, інструктаж з питань охорони праці, у тому числі при виконанні робіт з підвищеною небезпекою, ознайомлені з правилами поводження при виникненні аварій та надання першої допомоги потерпілим від нещасних випадків відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 № 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за №  231/10511 (НПАОП 0.00-4.12-05).

Графік роботи програміста встановлюється згідно правил внутрішнього трудового розпорядку.

Програміст зобов’язаний:

* піклуватися про особисту безпеку і здоров’я, а також про безпеку і здоров’я оточуючих людей у процесі виконання будь-яких робіт або під час знаходження на території підприємства;
* знати і виконувати вимоги інструкцій з охорони праці і по видах робіт на своєму робочому місці;
* виконувати роботу відповідно до вимог інструкційно-технологічної карти;
* вміти користуватися засобами індивідуального і колективного захисту;
* знати і виконувати Правила поводження з устаткуванням, інвентарем, користуватися технічним паспортом на устаткування;
* знати і виконувати обов’язки з охорони праці, передбачені колективним договором ( трудовим договором), правилами внутрішнього трудового розпорядку підприємства, в тому числі:
* вчасно починати і закінчувати роботу, дотримуватися розкладу технологічної і обідньої перерв;
* не виконувати роботи, що не передбачені змінним завданням;
* не перебувати на роботі в неробочій час без відповідного розпорядження керівника;
* дотримуватись правил корпоративного поводження;
* проходити в установленому порядку медичні огляди;
* вміти надати першу допомогу потерпілому від нещасного випадку;
* перед початком роботи перевіряти справність устаткування, огороджень, інженерно-технічних засобів безпеки, інвентарю, засобів пожежогасіння;
* співпрацювати з роботодавцем у справі організації безпечних і нешкідливих умов праці, особисто вживати можливих заходів щодо усунення будь-якої ситуації, що створює загрозу її життю чи здоров’ю або людям, які її оточують та навколишньому природному середовищу;
* при виявленні недоліків чи небезпеки зобов’язана повідомляти безпосереднього керівника або іншу посадову особу.

**3.2 Освітлення**

Для створення сприятливих умов для здорової роботи, які б запобігали швидкій втомлюваності очей, виникненню професійних захворювань, нещасних випадків і сприяли підвищенню продуктивності праці та якості продукції, виробниче освітлення повинно відповідати наступним вимогам:

* створювати на робочій поверхні освітленість, що відповідає характеру зорової роботи і не є нижчою за встановлені норми;
* забезпечити достатню рівномірність та постійність рівня освітленості у виробничих приміщеннях, щоб уникнути частої переадаптації органів зору;
* не створювати засліплювальної дії як від самих джерел освітлення, так і від інших предметів, що знаходяться в полі зору;
* не створювати на робочій поверхні різних та глибоких тіней (особливо рухомих);
* повинен бути достатній для розрізнення деталей контраст поверхонь, що освітлюються;
* не створювати небезпечних та шкідливих виробничих чинників (шум, теплові випромінювання, небезпека уражений струмом, пожежо- та вибухонебезпека світильників);
* повинно бути надійним і простим в експлуатації, економічним та естетичним.

**3.3 Мікроклімат**

Під оптимальними мікрокліматичними умовами розуміють поєднання параметрів мікроклімату, які при тривалому і систематичному впливі на людину забезпечують зберігання нормального теплового стану організму без активізації механізмів терморегуляції. Вони забезпечують відчуття теплового комфорту і створюють передумови для високого рівня працездатності.

Допустимі мікрокліматичні умови — це поєднання параметрів мікроклімату, які при тривалому та систематичному впливі на людину можуть викликати зміни теплового стану організму, що швидко минають і нормалізуються та супроводжуються напруженням механізмів терморегуляції в межах фізіологічної адаптації. При цьому не виникає ушкоджень чи порушень стану здоров'я, але можуть спостерігатись дискомфортні тепловідчуття, погіршення самопочуття та зниження працездатності.

Оптимальні і допустимі значення температури, відносної вологості та швидкості руху повітря встановлені залежно від категорії важкості робіт (Іа, 16, ІІа, ІІб, ІІІ), періоду року (теплий чи холодний) і характеристики робочого місця (постійне чи непостійне). Для постійних робочих місць нормуються оптимальні і допустимі значення вказаних параметрів, для непостійних — лише допустимі (табл. 3.3).

Період року визначається за середньодобовою температурою повітря зовнішнього середовища (при t сд< +10 °С — холодний період, а якщо t сд > +10 °С — теплий період року).

Робоче місце вважається постійним, якщо працівник знаходиться на ньому більше 50 % робочого часу або 2 години і більше безперервно.

Нормативні значення температури, як допустимі, так і оптимальні збільшуються в разі переходу від найлегших до найважчих робіт, а швидкості руху повітря навпаки — збільшуються. Оптимальні значення вологості для всіх категорій робіт становлять

40—60 %, а допустимі — не повинні перевищувати 75 % у холодний період року і теплий при температурі 24 °С і нижче, а при більш високих температурах — це значення менше.

Для того щоб визначити, чи відповідає повітряне середовище певного приміщення встановленим нормам, необхідно кількісно оцінити кожний з його параметрів.

Температуру вимірюють ртутними чи спиртовими термометрами. Для неперервної реєстрації температури навколишнього повітряного середовища застосовують самозаписувальні прилади — термографи.

Відносна вологість повітря (відношення фактичного вмісту маси водяних парів, що містяться в певний час в 1 м3 повітря, до максимально можливого їх вмісту при певній температурі) визначається психрометром Августа, аспіраційним психрометром, гігрометром та гігрографом.

Для вимірювання швидкості руху повітря використовують крильчасті (0,3—0,5 м/с) та чашкові (1—20 м/с) анемометри, а для визначення малих швидкостей руху повітря (менше 0,5 м/с) — термоанемометри та кататермометри. Температура нагрітих поверхонь вимірюється за допомогою електротермометрів, термопар та інших контактних приладів. Для вимірювання інтенсивності теплового опромінення використовують актинометри, термостовбці, спеціальні радіометри.

Нормалізація параметрів мікроклімату здійснюється за допомогою комплексу заходів та засобів колективного захисту, які включають будівельно-планувальні, організаційно-технологічні, санітарно-гігієнічні, технічні та інші рішення, які полягають в удосконаленні технологічних процесів та устаткування; раціональному розміщенні технологічного устаткування; автоматизації та дистанційному керуванні технологічними процесами; раціональній вентиляції, опаленні та кондиціонуванні повітря; раціоналізації режимів праці й відпочинку; застосуванні теплоізоляції устаткування та захисних екранів. Для профілактики перегрівань і переохолоджень робітників використовуються засоби індивідуального захисту.

Раціоналізація режимів праці та відпочинку досягається скороченням тривалості робочої зміни, введенням додаткових перерв, створенням умов для ефективного відпочинку в приміщеннях з нормальними метеорологічними умовами.

Застосування теплоізоляції устаткування та захисних екранів. Як теплоізоляційні матеріали широко використовуються азбест, азбоцемент, мінеральна вата, склотканина, керамзит, пінопласт та ін.

На виробництві застосовують також захисні екрани для огородження джерел теплового випромінювання від робочих місць. За принципом дії теплозахисні екрани поділяються:

* на тепловідбивні (поліровані або покриті білою фарбою металеві листи, загартоване скло з плівковим покриттям, металізовані тканини, плівковий матеріал);
* теплопоглинальні (металеві листи та коробки з теплоізоляцією, загартоване силікатне органічне скло та ін.);
* тепловідвідні (водяні завіси та металеві листи або сітки, з яких стікає вода);
* комбіновані.

**3.4 Рівень шуму та вібрації**

Рівні шуму та вібрації на робочих місцях осіб, що працюють з ПК, визначаються відповідно до ДСанПіН 3.3.2.007-98.

Суб'єктивне сприйняття шуму людиною значно відрізняється від фізичних характеристик звуку, так як слуховий орган неоднаково чутливий до звуків різних частот. Звуки низької частоти людина сприймає як менш гучні порівняно зі звуками високої частоти тієї ж інтенсивності. Одиницею рівня гучності є фон. Він відповідає різниці рівнів інтенсивності в 1 Б еталонного звуку при частоті 1000 Гц. Таким чином, на частоті 1000 Гц рівень гучності (у фонах) збігається з рівнями звукового тиску (в децибелах). Рівень гучності є фізіологічною характеристикою звукових коливань. Вплив шуму на організм людини індивідуальний.

Ступінь і характер впливу на людину вібрації залежить від амплітуди і частоти коливань. Так, власні частоти внутрішніх органів знаходяться в області 6 – 9 Гц. Отже, вібрація машин, площадок, ручних інструментів і т. д. особливо небезпечна при частотах 8 – 12; 17 – 25 Гц і т. д., тому що вони можуть бути резонансними для органів. При роботі з ручними машинами (їх вібрація знаходиться в області 100 Гц) виникають судинні розлади. Загальна вібрація, що має широкий спектр частоти, справляє несприятливий вплив на центральну нервову систему, вестибулярний апарат, шлунково-кишковий тракт, викликає запаморочення, оніміння кінцівок, захворювання суглобів. Тривалий вплив вібрації викликає фахове захворювання – вібраційну хворобу. 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 важко горючі 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.

**3.5 Електробезпека**

Сучасне виробництво нерозривно пов'язане з використанням електроенергії. В умовах експлуатації потужних енергосистем, електричних машин та апаратів, розвитку обчислювальної техніки і приладобудування, роботизації та комп'ютеризації виробництва важливого значення набуває проблема в електробезпеці — захисті електротехнічного персоналу та інших осіб, які обслуговують електроустаткування від ураження електричним струмом.

Аналіз загальної кількості виробничих нещасних випадків свідчить, що кількість електротравм становить 1,0-1,5%, а в енергетиці навіть 3-5%. Але серед нещасних випадків зі смертельним наслідком електротравми становлять 20-40% на виробництві, а в енергетиці до 60%, займаючи одне з перших місць. При цьому 60-85% смертельних уражень електричним струмом відбувається в електроустановках напругою до 1000 В (127-380 В).

Електротравматизм порівняно з іншими видами травматизму має деякі відмінні особливості.

Перша особливість полягає у тому, що організм людини не має органів, за допомогою яких можна дистанційно визначити наявність напруги, як, наприклад, теплову, світлову енергію, деталі, які рухаються. Тому захисна реакція організму виявляється тільки після потрапляння під напругу.

Друга особливість електротравматизму полягає в тому, що струм, який проходить крізь людину, діє не тільки в місцях контактів та на шляху протікання крізь організм, а й викликає рефлекторну взаємодію, спричиняючи порушення нормальної діяльності окремих органів (серцево-судинної системи, системи дихання).

Третьою особливістю є можливість отримання електротравми, не маючи безпосереднього контакту зі струмопровідними частинами - переміщення по землі поблизу пошкодженої установки (у випадку замикання на землю), ураження через електричну дугу.

Четверта особливість електротравматизму — це те, що у більшості випадків для розслідування, обліку та аналізу доступні тільки електротравми з тяжкими та смертельними наслідками.

Безпека людини на виробництві залежить від багатьох факторів і, зокрема, від рівня електробезпеки. Грамотне вирішення проблеми електробезпеки має забезпечувати людині використання електричної енергії в будь-яких умовах без ризику для життя.

**4 Опис технологічного процесу**

Кожне підприємство є складною виробничою системою, яка призначена для виробництва нового виду продукції. Одержання певного виду продукції є результатом певного виробничого процесу.

Технологі́чний проце́с — це впорядкована послідовність взаємопов'язаних дій та операцій, що виконуються над початковими даними до отримання необхідного результату.

Технологічний процес має складну структуру. Його складові - це операції, кожна з яких є елементарним технологічним процесом.

Операцією називають закінчену частину технологічного процесу, яка охоплює всі дії обладнання та робітника над одним або кількома об'єктами, що одночасно обробляються на одному і тому самому робочому місці.

4.1 Технологічний процес на підприємстві

Команда Web Systems Solutions займається розробкою сайтів.

Розробка сайтів здійснюється в декілька етапів:

* Планування — визначення тематики і призначення майбутнього сайту;
* Розробка — розробка структури сайту, добір матеріалів, вибір програмних засобів для його створення;
* Створення окремих сторінок відповідно до структури, включення до них гіперпосилань;
* Тестування — перевірка та редагування веб-сайту.
* Розміщення — розміщення сайту в Інтернеті;
* Підтримка — оновлення вмісту сайту.

На етапі планування, перш за все, слід визначити призначення майбутнього сайту: це буде персональний сайт або сайт організації, електронна енциклопедія чи [сайт](http://school.xvatit.com/index.php?title=%D0%A2%D0%B5%D0%BC%D0%B0_4._%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%B0_%D0%B7_Internet_Explorer.) бібліотеки, сайт для дистанційного навчання тощо. Тут же доцільно визначити, буде [сайт](http://school.xvatit.com/index.php?title=%D0%A2%D0%B5%D0%BC%D0%B0_4._%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%B0_%D0%B7_Internet_Explorer.)тематичним чи різні його сторінки будуть присвячені різним темам і яким саме.

Наступний крок — розробка структури сайту. При розробці структури сайту слід визначитися з необхідною кількістю сторінок та встановити зв'язки між ними. Розрізняють лінійну, ієрархічну та довільну структури сайту.  
Лінійну структуру веб-сайту доцільно використовувати у разі послідовного представлення інформації, наприклад, про товари та послуги або матеріали навчального посібника. Перегляд таких сайтів здійснюється послідовно: від початкової (головної) до останньої сторінки. Кожна сторінка має посилання тільки на одну, наступну сторінку сайту. Інколи, для зручності навігації по сайту до сторінки також додається посилання на попередню сторінку.

На етапі створення відбувається наповнення веб-сторінок конкретними матеріалами, а також створення гіперпосилань для зручної навігації сайтом. При цьому потрібно слідувати правилам оформлення (дизайну) веб-сторінок, якими ви користувалися при створенні презентацій.

Дуже важливим при створенні сайта є етап тестування. На цьому етапі потрібно перевірити:

* чи правильно працюють усі гіперпосилання;
* чи зручною є навігація сайтом;
* чи відкриваються при відкритті сторінок графічні зображення;
* чи зручно розташовані для сприйняття матеріали на сторінках тощо.

У разі необхідності, потрібно внести зміни в наповнення або структуру сайту, змінити гіперпосилання.

На наступному етапі потрібно визначити, де буде розміщено створений вами сайт. Веб-сайт можна розмістити:

* на власному сервері установи;
* на сервері вашого провайдера;
* на сервері організації, яка спеціалізується у наданні послуг розміщення сайтів користувачам Інтернету;
* на сервері, який надає послуги вільного і безкоштовного розміщення сайтів.

Після розміщення сайта в Інтернеті потрібно здійснювати його підтримку, щоб сайт не втрачав своєї популярності. Ця підтримка полягає в періодичному оновленні та доповненні існуючих матеріалів, створенні нових цікавих сторінок тощо.

Програмне забезпечення, яке використовується в роботі підприємства:

* Abobe Photoshop;
* Sublime Text;
* Jet Bains PhpStorm;
* WAMP;
* Google Chrome.
  1. Технологічний процес на робочому місці

Технологічний процес на робочому місці починається з налаштування середовища для розробки, а далі слідують такі етапи:

* отримання завдання;
* декомпозиція завдання на підзадачі;
* вибір початкової підзадачі;
* виконання завдання;
* тестування внесених змін;

Якщо тестування відбулося успішно то відбувається завантаження змін в систему контролю версій. А якщо не успішно, то відбувається виправлення помилок, та завантаження змін в систему контролю версій.

**4.2 Застосування прогресивних технологій**

Інформаційна індустрія (ІТ-індустрія, ІТ-галузь) — широкомасштабне виробництво інформаційних товарів і послуг різного типу на базі найновіших ІТ-технологій.

Front-end розробники використовують HTML5 та CSS3, та бібліотеки і фреймворки для Java Script:

* AngularJS — [JavaScript](https://uk.wikipedia.org/wiki/JavaScript)-[фреймворк](https://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) з [відкритим](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) [програмним кодом](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%80%D1%86%D0%B5%D0%B2%D0%B8%D0%B9_%D0%BA%D0%BE%D0%B4), який розробляє [Google](https://uk.wikipedia.org/wiki/Google). Призначений для розробки [односторінкових додатків](https://uk.wikipedia.org/wiki/Single_Page_Application), що складаються з одної [HTML](https://uk.wikipedia.org/wiki/HTML) сторінки з [CSS](https://uk.wikipedia.org/wiki/CSS) і [JavaScript](https://uk.wikipedia.org/wiki/JavaScript). Його мета — розширення браузерних [застосунків](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) на основі шаблону [Модель-вид-контролер](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B2%D0%B8%D0%B4-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80) (MVC), а також спрощення їх тестування та розробки;
* React — [відкрита](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) [JavaScript](https://uk.wikipedia.org/wiki/JavaScript) [бібліотека](https://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%B1%D0%BB%D1%96%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC) для створення [інтерфейсів користувача](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87%D0%B0), яка покликана вирішувати проблеми часткового оновлення вмісту [веб-сторінки](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0), з якими стикаються в розробці [односторінкових застосунків](https://uk.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9_%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA). Розробляється [Facebook](https://uk.wikipedia.org/wiki/Facebook), [Instagram](https://uk.wikipedia.org/wiki/Instagram) і спільнотою індивідуальних розробників;
* Vue.js  — [JavaScript](https://uk.wikipedia.org/wiki/JavaScript)-[фреймворк](https://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) що використовує шаблон [MVVM](https://uk.wikipedia.org/wiki/MVVM) для створення інтерфейсів користувача на основі моделей даних, через [реактивне](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) зв'язування даних.

Back-end розробники використовують php та фреймворки такі,як:

* Yii2 — це високопродуктивний веб-[фреймворк](https://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA" \o "Фреймворк), написаний на [PHP](https://uk.wikipedia.org/wiki/PHP), реалізує парадигму модель-вид-контролер. Yii — скорочення від «Yes It Is!»;
* Laravel — безкоштовний, з [відкритим кодом](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D1%82%D0%B8%D0%BA%D0%B0_%D0%B2%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D1%83) [PHP](https://uk.wikipedia.org/wiki/PHP)-[фреймворк](https://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA), створений Taylor Otwell і призначений для розробки веб-додатків відповідно до [шаблону](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%BD%D1%96_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) [model–view–controller](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B2%D0%B8%D0%B4-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80) ([MVC](https://uk.wikipedia.org/wiki/MVC));
* Symfony — відкритий [PHP](https://uk.wikipedia.org/wiki/PHP)-[фреймворк](https://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA), що реалізує концепцію [модель-вид-контролер](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B2%D0%B8%D0%B4-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80) ([MVC](https://uk.wikipedia.org/wiki/MVC)) та автоматизовує найзагальніші веб-задачі, являє собою широконалаштовну систему пов'язаних [класів](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) і призначений для розробки та керування [веб-застосунками](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA). [Веб-сайт](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82) першоі версії symfony-project.com був запущений [18](https://uk.wikipedia.org/wiki/18) [жовтня](https://uk.wikipedia.org/wiki/%D0%96%D0%BE%D0%B2%D1%82%D0%B5%D0%BD%D1%8C) [2005](https://uk.wikipedia.org/wiki/2005) року.

**5 Індивідуальне завдання**

**5.1 Постановка завдання**

Реалізувати веб-сервіс «Пошта» . Передбачити наступні функціональні можливості: калькулятор вартості доставки посилки, калькулятор терміну доставки посилки, відслідковування посилки за номером, редагування особистої інформації у кабінеті користувача. Реалізувати адміністративну частину для додавання, редагування, видалення замовлень, керування відгуками, та частину користувача в якій буде змога реєструватись та входити в обліковий запис , залишати відгуки ,редагувати данні в особистому кабінеті, переглядати свої замовлення доставки. Для реалізації використати наступні технології: HTML, СSS, PHP, Yii2 Framework, MySql.

**5.2 Короткий опис індивідуального завдання**

Для початку було спроектовано базу даних.

База даних була розміщена на сервері Open Server та створена за допомогою сервісу PhpMyAdmin(My Sql). База даних була приведена до першої, другої та третьої нормальних форм. Також у базі даних були виставлені зв’язки між табличками.

Веб-сервіс було розроблено з базового шаблону Yii2. Сервіс був розміщений на локальному сервері Open Server x64, також на сервері було виділене доменне ім’я для веб-сервісу «Mail», тобто ми можемо звертатись до сайту за адресою [http://mail/](http://mail/%20) .

У проекті реалізована частина адміністратора та користувача. Розроблена система відгуків, тобто кожен користувач може залишати відгуки.

Сервіс налічує два калькулятори для розрахунку вартості перевезення посилки та терміну перевезення посилки в залежності від введених даних. Також користувач або просто гість може відслідковувати власну посилку за її номером, тобто переглядати її місцеположення. Також у користувачів є можливість замовити кур'єра на будь яку адресу.

Зокрема кожен користувач може відредагувати свої дані у кабінеті.

**5.3 Опис вхідних даних**

Дані про посилку:

* висота посилки(см) – тип даних double;
* ширина посилки(см) – тип даних double;
* довжина посилки(см) – тип даних double;
* адреса доставки посилки – тип даних string;
* адреса відділення з якого відправляється посилка – тип даних string;
* дата відправлення посилки – тип даних datetime;
* вага посилки(кг) – тип даних double.

Дані про відгуки:

* текст відгуку – тип даних string.

Інформація про користувача:

* прізвище – тип даних string;
* ім’я – тип даних string;
* по-батькові – тип даних string;

**5.4 Опис вихідних даних**

* термін доставки посилки – тип даних integer;

Виведення терміну доставки посилки відбувається після обрахування у текстове поле.



Рисунок 3 – вигляд текстового поля, для виведення терміну доставки

* вартість доставки посилки – тип даних integer;

Виведення терміну доставки посилки відбувається після обрахування у текстове поле.



Рисунок 4 – вигляд текстового поля, для виведення ціни доставки

* статус доставки посилки – тип даних string;

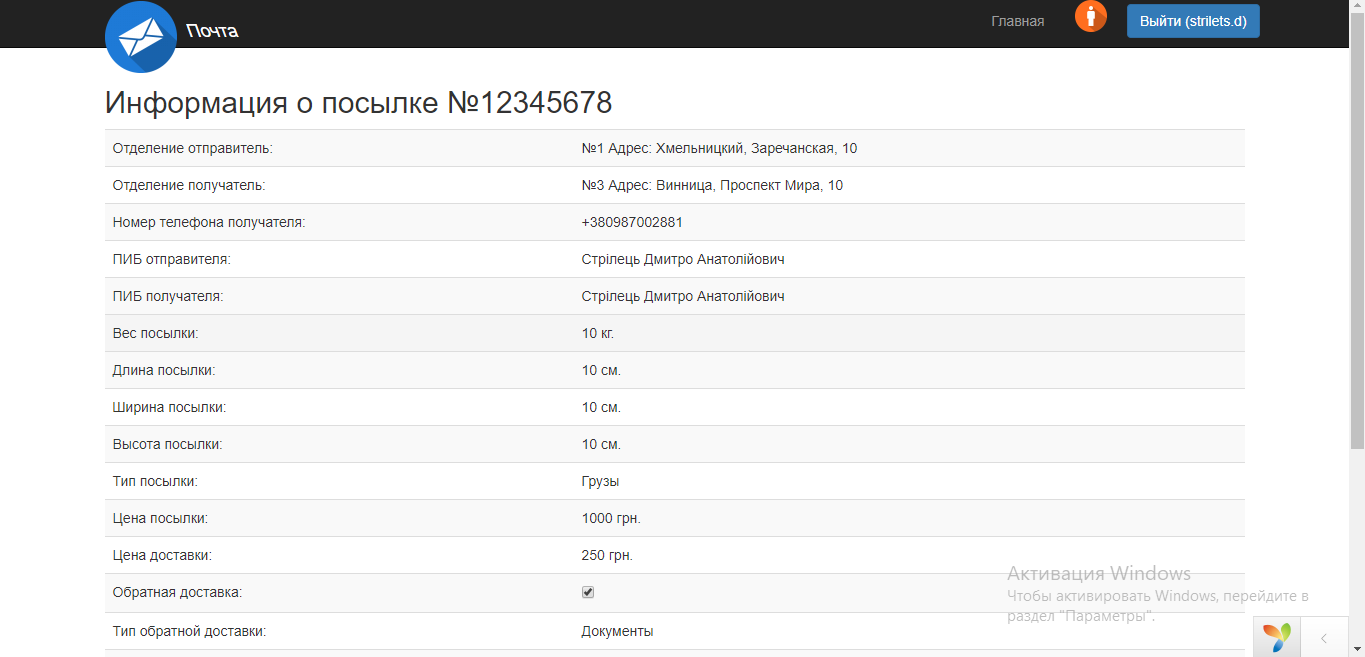
Статус доставки виводиться у вигляді текстового поля.



Рисунок 5 – вигляд текстового поля, для виведення статусу доставки

* дані про посилку.

Дані про посилку виводяться у вигляді таблички.



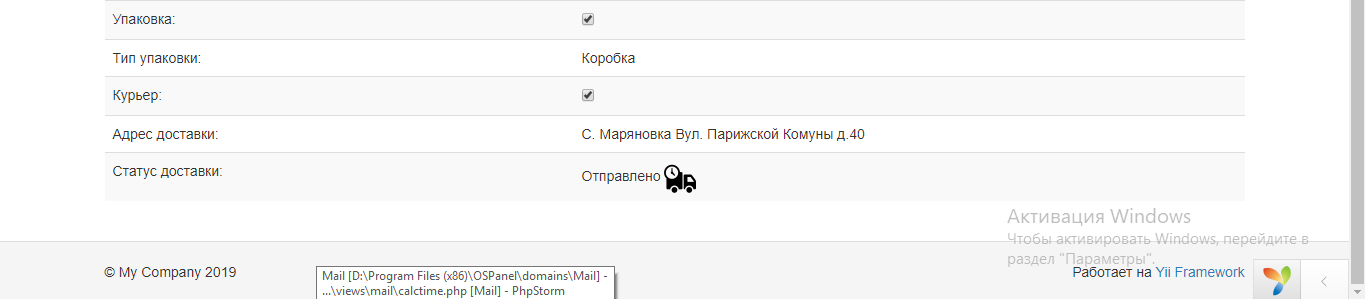


Рисунок 6 – вигляд таблиці з даними про посилку.

**5.5 Реалізація та результати**

Для реалізації індивідуального завдання насамперед використовувалось середовище для розробки веб-сайтів Jet Brains PhpStorm.

PhpStorm являє собою інтелектуальний редактор для [PHP](https://uk.wikipedia.org/wiki/PHP), [HTML](https://uk.wikipedia.org/wiki/HTML), J[avaScript](https://uk.wikipedia.org/wiki/JavaScript) з можливостями аналізу коду на льоту, запобігання помилок у [сирцевому коді](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%80%D1%86%D0%B5%D0%B2%D0%B8%D0%B9_%D0%BA%D0%BE%D0%B4) і автоматизованими засобами [рефакторинга](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3) для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3/5.4/5.5/5.6/7.0/7.1,включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний [SQL](https://uk.wikipedia.org/wiki/SQL)-редактор з можливістю редагування отриманих результатів запитів.

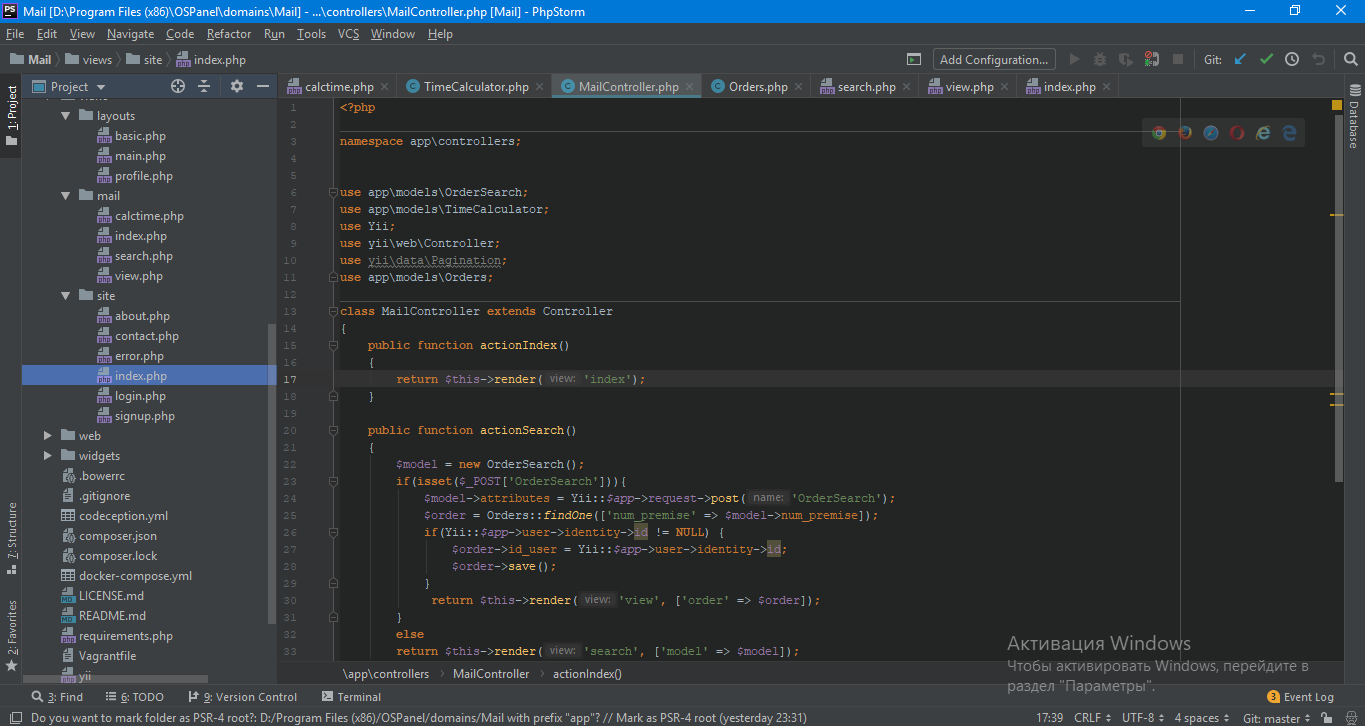
Використовувались спеціальні плагіни для розширення функціональності PhpStorm з PHP фреймворком Yii2, наприклад: Yii2 support. 

Рисунок 7 – середовище PhpStorm

Основною технологією для реалізації індивідуального завдання був PHP фреймворк Yii2.

Переваги Yii2:

* Продуктивність. Yii2 має дуже високу продуктивність в обробці запитів.
* Безпека. Включені заходи безпеки, такі як Cross-Site Scripting, Request міжсайтовий, Cookie атаки та ін.  YII має вбудовану підтримку аутентифікації.
* Найкращі можливості кешування. YII2 надає різні кеш-компоненти для зберігання кешованих даних в різних середовищах.
* Дружній з кодом сторонніх додатків. YII також призначений для роботи з кодами сторонніх виробників.

Також використовувався сервіс для контролю версій проекту GitHub, на ньому було створено репозиторій під назвою Mail. Цей сервіс забезпечує хмарне зберігання проекту, на випадок, якщо його буде втрачено, та можливість повернення до попередньої версії проекту.

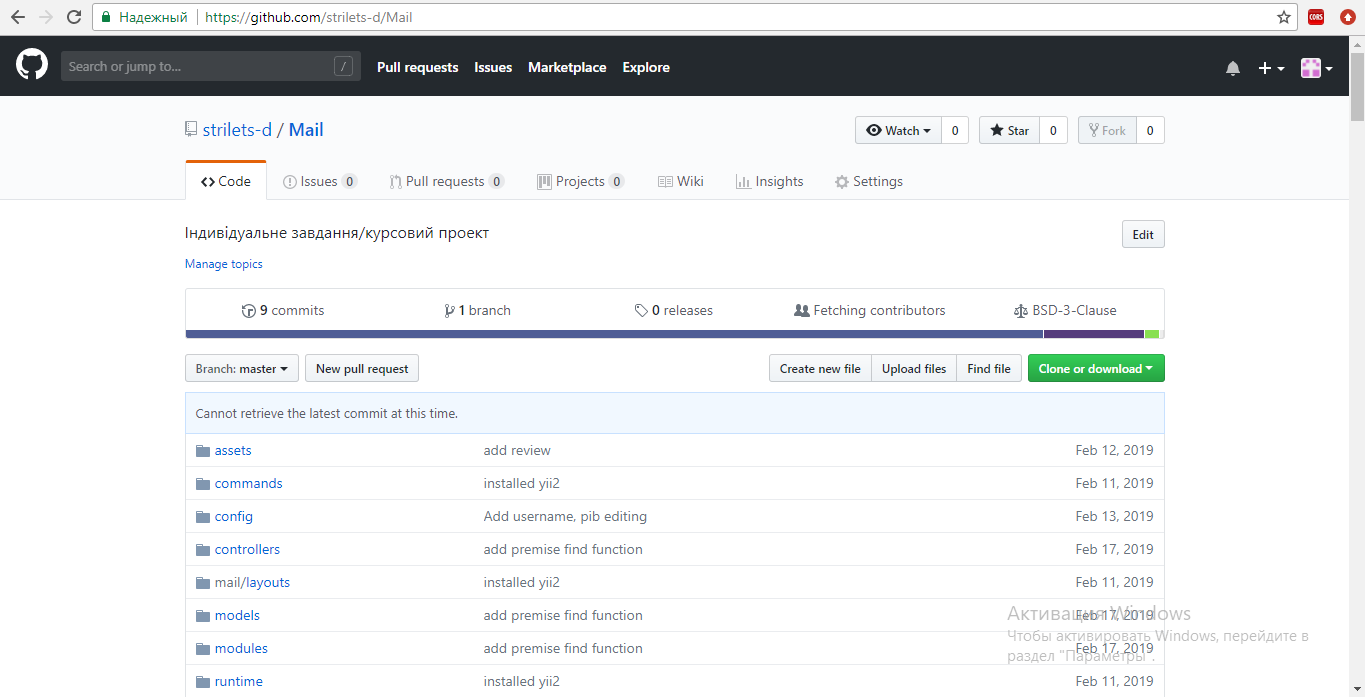


Рисунок 8 – вигляд сервісу GitHub та репозиторію Mail.

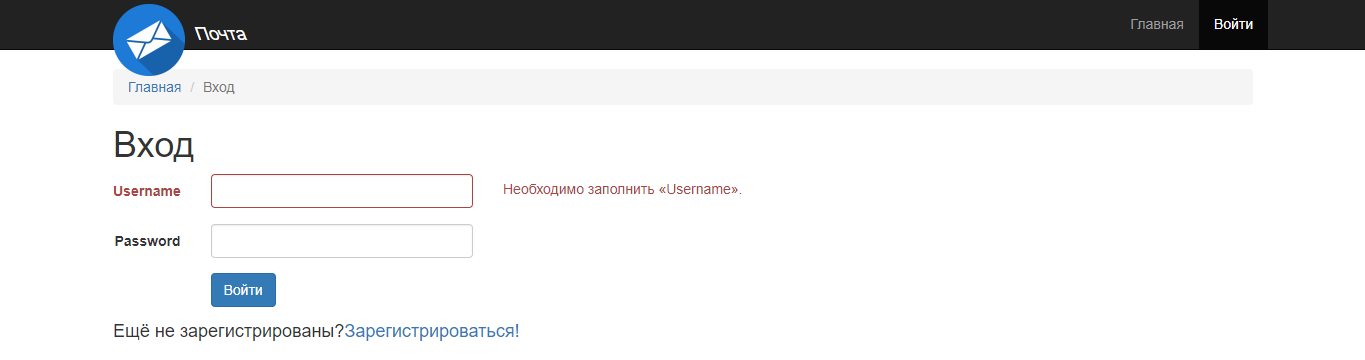
Далі було розпочато реалізацію. Спочатку мною було додано та реалізовано функціонал реєстрації та авторизації користувача. 

Рисунок 9 – вигляд сторінки авторизації

Код функції для перевірки паролю:

public function validatePassword($attribute,$params)

{

if(!$this->hasErrors())

{

$user = $this->getUser();

if(!$user || !$user->validatePassword($this->password))

{

$this->addError($attribute,'Логин или пароль введены неверно1');

}

}

}

Функція для входу:

public function actionLogin()

{

if(!Yii::$app->user->isGuest)

{

return $this->goHome();

}

$login\_model = new Login();

if( Yii::$app->request->post('Login'))

{

$login\_model->attributes = Yii::$app->request->post('Login');

if($login\_model->validate())

{

Yii::$app->user->login($login\_model->getUser());

return $this->goHome();

}

}

return $this->render('login',['login\_model'=>$login\_model]);

}

Далі було реалізовано особистий кабінет користувача:

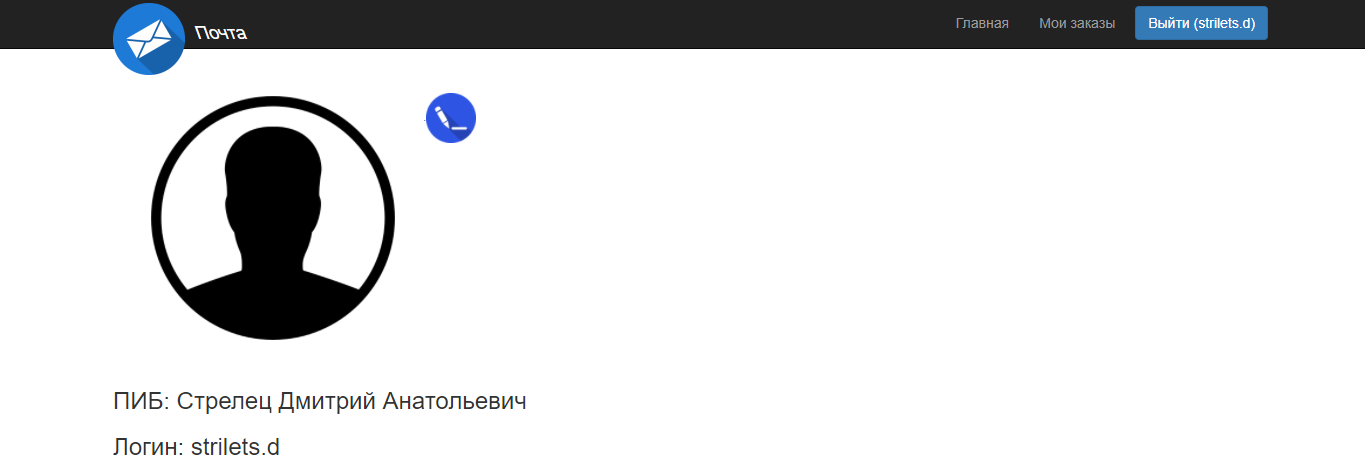


Рисунок 10 – вигляд особистого кабінету користувача

Функція для зміни фото користувача:

public function actionSetImage()

{

$model = new ImageUpload();

if(Yii::$app->request->isPost){

$user = $this->findModel(Yii::$app->user->identity->id);

$file = UploadedFile::getInstance($model, 'image');

$user->saveImage($model->uploadFile($file));

$this->redirect('http://mail/index.php?r=profile');

}

return $this->render('image', ['model' => $model]);

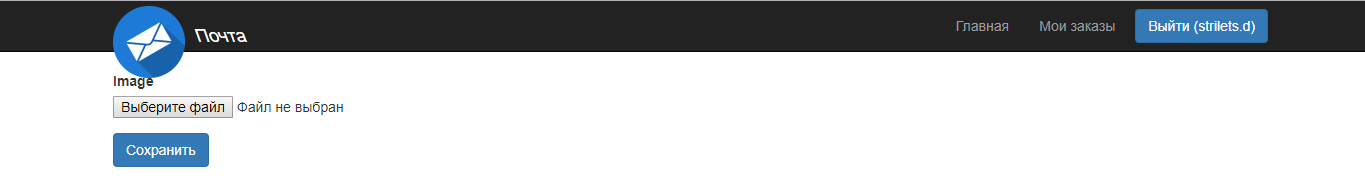
}

Рисунок 11 – вигляд інтерфейсу зміни фото користувача

Функція для оновлення ПІБ та логіну користувача:

public function actionUpdate($id)

{

$model = $this->findModel($id);

if ($model->load(Yii::$app->request->post()) && $model->save()) {

return $this->redirect('http://mail/index.php?r=profile');

}

return $this->render('update', [

'model' => $model,

]);

}

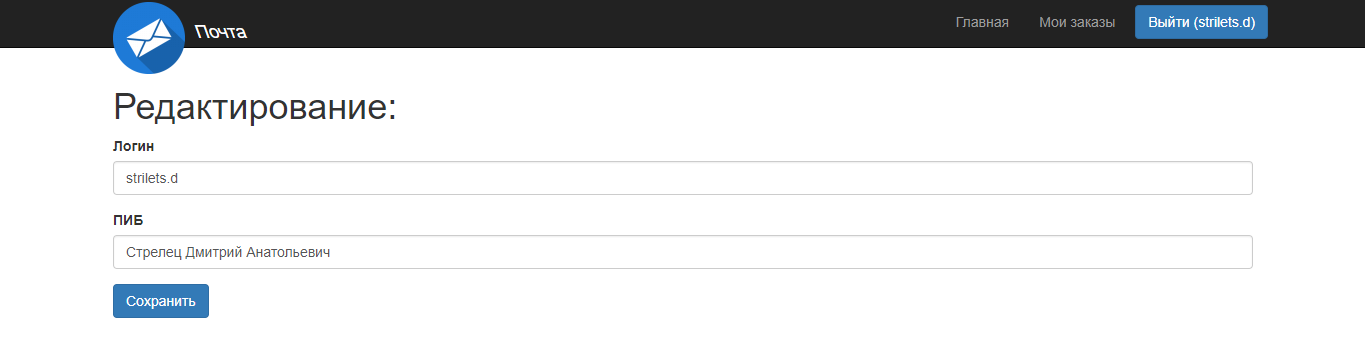


Рисунок 12 – вигляд інтерфейсу зміни данних корстувача

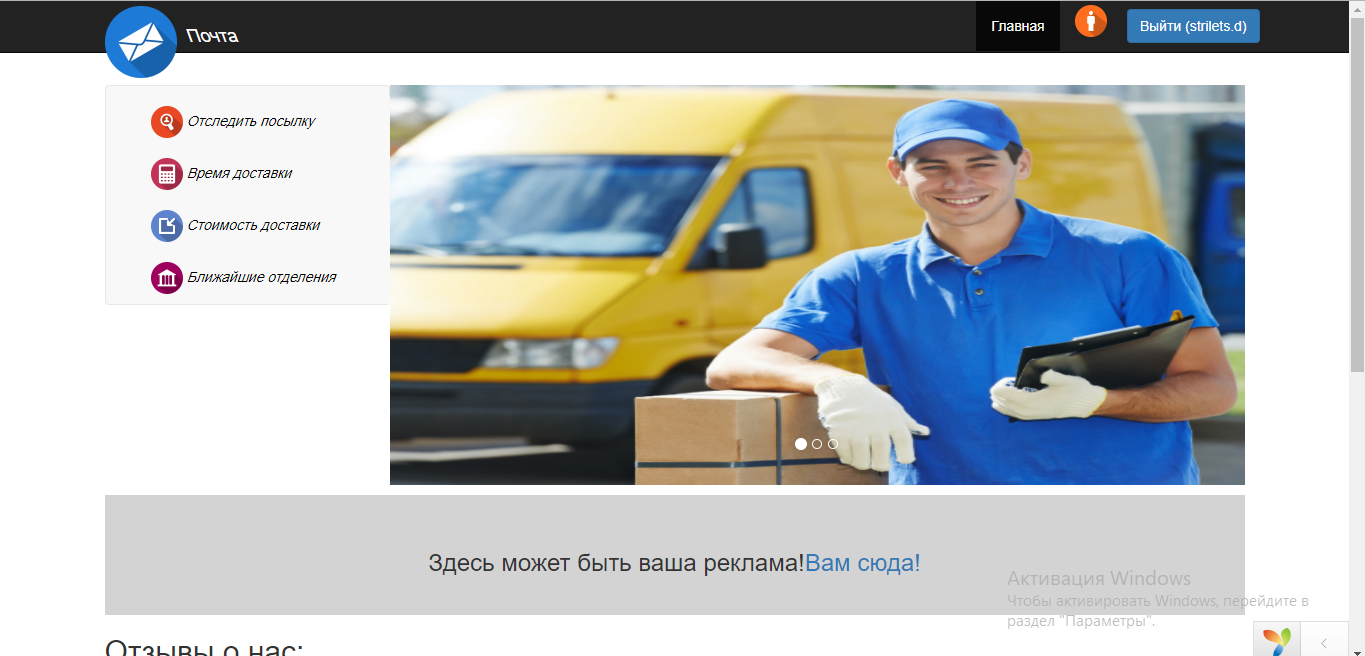
Далі було реалізовано голоний інтерфейс сторінки. 

Рисунок 13 – вигляд головного інтерфейсу сторінки

Далі було реалізовано вивід відгуків на головній сторінці та інтерфейс для введення відгуку користувачем, якщо він зареєстрований.

Функція для виводу відгуків на головній сторінці:

public function actionIndex()

{

$query = Review::find();

$pagination = new Pagination([

'defaultPageSize' => 3,

'totalCount' => $query->count(),

]);

$reviews = $query->orderBy('date\_review')

->offset($pagination->offset)

->limit($pagination->limit)

->all();

if(!Yii::$app->user->isGuest){

$model = new ReviewForm();

if(isset($\_POST['ReviewForm']))

{

$model->attributes = Yii::$app->request->post('ReviewForm');

if($model->validate() && $model->review())

{

return $this->redirect(['index']);

}

}

}

$check\_model = new ReviewForm();

return $this->render('index', [

'reviews' => $reviews,

'pagination' => $pagination,

'review\_model' => $model,

'model' => $check\_model,

]);

}

Функція для виводу форми вводу відгуку, якщо користувач авторизований:

<?= LinkPager::widget(['pagination' => $pagination]) ?>

<?php $form = ActiveForm::begin([

'id' => 'login-form',

'layout' => 'horizontal',

'fieldConfig' => [

'template' => "{label}\n<div class=\"col-lg-3\">{input}</div>\n<div class=\"col-lg-8\">{error}</div>",

'labelOptions' => ['class' => 'col-lg-1 control-label'],

],

]); ?>

<?= (Yii::$app->user->isGuest) ? '' : $form->field($review\_model,'text')->textInput() ?>

<div class="form-group">

<div class="col-lg-offset-1 col-lg-11">

<?= (Yii::$app->user->isGuest) ? '' : Html::submitButton('Оставить отзыв', ['class' => 'btn btn-primary', 'name' => 'login-button'])?>

</div>

</div>

<?php $form = ActiveForm::end();?>

Вивід відгуків у базу даних:

<h2>Отзывы о нас:</h2>

<?php foreach ($reviews as $review): ?>

<div class="block">

<text style="display:inline-block;"><?= Html::img("@web/img/{$review->getUser()->image}", ['style' => 'width: 40px; height: 40px; margin-left:10px; border-radius:50%;']) ?> <b><?= Html::encode("{$review->getUser()->username}");?></b> <text class="data"> <?= Html::encode("{$review->date\_review}");?></text></text><br>

<text class="text"><?= Html::encode("{$review->text\_review}");?></text><br>

</div>

<?php endforeach; ?>

Далі було реалізовано функціонал пошуку посилки за номером.

Функція пошуку посилки:

public function actionSearch()

{

$model = new OrderSearch();

if(isset($\_POST['OrderSearch'])){

$model->attributes = Yii::$app->request->post('OrderSearch');

$order = Orders::findOne(['num\_premise' => $model->num\_premise]);

if(Yii::$app->user->identity->id != NULL) {

$order->id\_user = Yii::$app->user->identity->id;

$order->save();

}

return $this->render('view', ['order' => $order]);

}

else

return $this->render('search', ['model' => $model]);

}



Рисунок 14 – вигляд інтерфейсу пошуку посилки

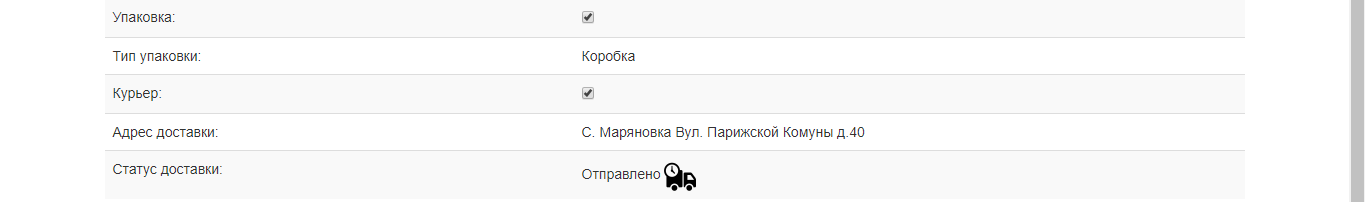
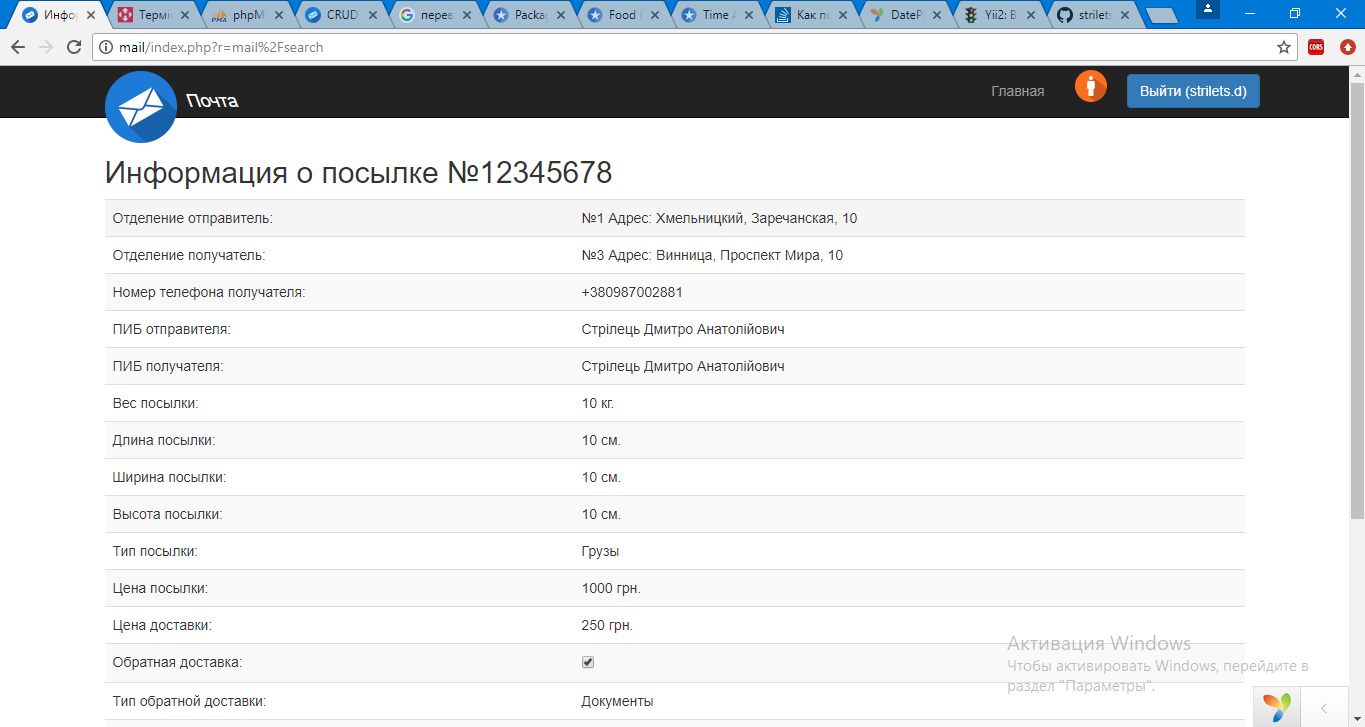


Рисунок 15 – вигляд інтерфейсу виведення інформації про посилку

Код для виведення інформації про посилку:

<?php

/\*\*

\* Created by PhpStorm.

\* User: dimas

\* Date: 17.02.2019

\* Time: 15:26

\*/

use app\models\Departments;

use app\models\Packaging;

use app\models\ReverseDelivery;

use app\models\Status;

use app\models\TypePremise;

use yii\helpers\Html;

use yii\bootstrap\ActiveForm;

$this->title='Информация';

?>

<title><?= Html::encode($this->title) ?></title>

<h2>Информация о посылке №<?= $order->num\_premise ?></h2>

<table class="table table-striped table-hover">

<tr>

<td>Отделение отправитель:</td>

<td>№<?= Departments::findOne(['id\_department' => $order->id\_department])->num\_department ?> Адрес: <?= Departments::findOne(['id\_department' => $order->id\_department])->address\_department ?></td>

</tr>

<tr>

<td>Отделение получатель:</td>

<td>№<?= Departments::findOne(['id\_department' => $order->id\_dep\_rec])->num\_department ?> Адрес: <?= Departments::findOne(['id\_department' => $order->id\_dep\_rec])->address\_department ?></td>

</tr>

<tr>

<td>Номер телефона получателя:</td>

<td>+<?= $order->phone\_user ?></td>

</tr>

<tr>

<td>ПИБ отправителя:</td>

<td><?= $order->pib\_sender ?></td>

</tr>

<tr>

<td>ПИБ получателя:</td>

<td><?= $order->pib\_recipient ?></td>

</tr>

<tr>

<td>Вес посылки:</td>

<td><?= $order->weight\_premise ?> кг.</td>

</tr>

<tr>

<td>Длина посылки:</td>

<td><?= $order->length\_premise ?> см.</td>

</tr>

<tr>

<td>Ширина посылки:</td>

<td><?= $order->width\_premise ?> см.</td>

</tr>

<tr>

<td>Высота посылки:</td>

<td><?= $order->width\_premise ?> см.</td>

</tr>

<tr>

<td>Тип посылки:</td>

<td><?= TypePremise::findOne(['id\_type' => $order->id\_type])->name\_type ?></td>

</tr>

<tr>

<td>Цена посылки:</td>

<td><?= $order->price\_premise ?> грн.</td>

</tr>

<tr>

<td>Цена доставки:</td>

<td><?= $order->price\_delivery ?> грн.</td>

</tr>

<?php if($order->reverse\_delivery == NULL){

echo "<tr>

<td>Обратная доставка:</td>

<td><input type='checkbox' checked='false'></td>

</tr>";

}else {

echo "<tr>

<td>Обратная доставка:</td>

<td><input type='checkbox' checked='true'></td>

</tr>";

echo '<tr>

<td>Тип обратной доставки:</td>

<td>'.ReverseDelivery::findOne(['id\_reverse\_del' => $order->reverse\_delivery])->type\_reverse\_del.'</td>

</tr>';

}

?>

<?php if($order->packaging == NULL){

echo "<tr>

<td>Упаковка:</td>

<td><input type='checkbox' checked='false'></td>

</tr>";

}else {

echo "<tr>

<td>Упаковка:</td>

<td><input type='checkbox' checked='true'></td>

</tr>";

echo '<tr>

<td>Тип упаковки:</td>

<td>'.Packaging::findOne(['id\_packaging' => $order->packaging])->type\_packaging.'</td>

</tr>';

}

?>

<?php if($order->packaging == NULL){

echo "<tr>

<td>Курьер:</td>

<td><input type='checkbox' checked='false'></td>

</tr>";

}else {

echo "<tr>

<td>Курьер:</td>

<td><input type='checkbox' checked='true'></td>

</tr>";

echo '<tr>

<td>Адрес доставки:</td>

<td>'.$order->address\_delivery.'</td>

</tr>';

}

?>

<tr>

<td>Статус доставки:</td>

<td><?= Status::findOne(['id\_status' => $order->status])->name\_status?> <?= Html::img('@web/img/'.Status::findOne(['id\_status' => $order->status])->image\_status) ?></td>

</tr>

</table>

**ВИСНОВОК**

Я пройшов технологічну практику у Web Systems Solutions. Навчився отримувати завдання по проекту, декомпозувати його на підзадачі. Навчився розпочинати з підзадачі, яка найпростіша. Поглибив свої зання PHP фреймворку Yii2, вивчив стандарти кодінгу SOLID та PSR-2. Навчився обговорювати проект з командою. Навчився використовувати документацію по фреймворку yii2. Навчився користуватись генератором коду GII.

ДЖЕРЕЛА

1 Законодавство України про охорону праці (збірник нормативних документів. У 4 т. - К.: Держнаглядохоронпраці; Основа, 1995.

2 Жидецький В.Ц. Охорона праці користувачів комп´ютерів. – Львів: Афіша, 2001. – 176 с.

3 https://habrahabr.ru/ - сайт для публікації новин, аналітичних статей, думок пов’язаних з ІТ, бізнесом та Інтернетом.

4 https://www.github.com/ - сайт для спільної розробки прграмного забезпечення.