# ВСТУП

Правильне ведення бухгалтерського обліку - це гарантія фінансової стабільності, тому що бухгалтерія не обмежується тільки перевіркою правильності заповнення звітів та відомості витрат і доходів. Бухгалтерія - це своєрідне дослідження фінансового стану, прямий контроль за всіма пунктами і статтями грошових надходження і збитків. Це початковий етап для проведення оптимізації фінансового становища.

Домашня бухгалтерія представляє собою точний контроль за всіма надходженнямми до сімейного бюджету і облік всіх статей витрат. Завдання на перший погляд просте, але практика показує, що багато людей пускають на самоплив домашню бухгалтерію. У подібному випадку рахунки і квитанції накопичуються в якійсь шухляді комоду, витрати ведуться бездумно, нічого не фіксується. Чому в такому разі дивуватися, коли в сімейному бюджеті раптово з'являється «діра». Для того, щоб уникнути подібних ситуацій необхідно вести мінімальний бухгалтерський облік, прогнозувати можливі статті витрат, планувати вигідні вклади вільних коштів. Для цього й необхідний додаток для зручнішої роботи зберігання та моніторингу грошових потоків.

В роботі розглядається питання розробки нового інструменту для роботи з домашньою бухгалтерією, що допоможе користувачам отримувати якісну та вичерпну інформацію щодо їх доходів та витрат.

# ЕТАПИ РОЗРОБКИ ВЕБ-САЙТУ

Веб-розробка — процес створення веб-сайтів або веб-додатки. Основними етапами процесу є складання технічного завдання, розробка веб-дизайн, програмування для веб на стороні клієнта і сервера, а також конфігурування веб-сервера.

**1.1 Створення технічного завдання**

Складанням технічного завдання для фахівців займається менеджер проекту. Робота з замовником починається із заповнення брифу, в якому замовник викладає свої побажання щодо візуального представлення і структури сайту, вказує на помилки в старій версії сайту, наводить приклади сайтів конкурентів. Виходячи з брифа, менеджер складає технічне завдання, враховуючи можливості програмних і дизайнерських засобів. Етап закінчується після затвердження технічного завдання замовником. Важливо відразу зазначити, що етапи проектування веб-сайтів залежать від багатьох факторів, таких як обсяг сайту, функціональність, завдання, які повинен виконувати майбутній ресурс і багато іншого. Однак, є кілька етапів, які в обов'язковому порядку присутні в плануванні будь-якого проекту

**1.2 Програмування**

Проаналізовано існуючі рішення додатків, котрі візуалізують грошові потоки. Визначені їх основні недоліки, до яких можна віднести застарілий дизайн, складність у внесенні даних користувачем, відсутність можливості синхронізувати декілька аккаунтів, неможливість автоматизованого внесення даних по надходженням та витратам з кредитної карти, відсутність інтеграції з соціальними мережами, відсутність можливості представлення інформації у вигляді графіків, неможливість зручно користуватися додатком з мобільних пристроїв.

**Todo найти аналоги**

Визначені основні вимоги до веб-додатку ведення домашньої бухгалтерії та спроектована його узагальнена архітектура. В якості архітектури сервісу обрана трирівнева архітектура «клієнт-сервер застосунків – сервер бази даних».

У комп'ютерних технологіях трирівнева архітектура, синонім триланкова архітектура передбачає наявність наступних компонентів програми: [клієнтський](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) [застосунок](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) , підключений до [сервера застосунків](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BA%D1%96%D0%B2), який в свою чергу підключений до [серверу бази даних](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).  
[Клієнт](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) — це [інтерфейсний](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" \o "Інтерфейс користувача) ([графічний](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81)) компонент, який представляє перший рівень, власне [застосунок](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) для кінцевого користувача. Перший рівень не має прямих зв'язків з [базою даних](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) (за вимогами безпеки), не навантажений основною бізнес-логікою (за вимогами масштабованості) і зберігає стан програми (за вимогами надійності). На перший рівень винесена найпростіша бізнес-логіка: інтерфейс авторизації, алгоритми шифрування, перевірка значень, що вводяться, на допустимість і відповідність формату, нескладні операції (сортування, групування, підрахунок значень) з даними, вже завантаженими на термінал.

* [Сервер застосунків](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BA%D1%96%D0%B2) розташовується на другому рівні. На другому рівні зосереджена більша частина бізнес-логіки. Поза ним залишаються фрагменти, що експортуються на термінали , а також розміщені в третьому рівні збережені процедури і тригери.
* Сервер бази даних забезпечує зберігання даних і виноситься на третій рівень. Це [об'єктно-орієнтована](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) [СУБД](https://uk.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94). Коли третій рівень являє собою базу даних разом з [збереженими процедурами](https://uk.wikipedia.org/wiki/%D0%97%D0%B1%D0%B5%D1%80%D0%B5%D0%B6%D0%B5%D0%BD%D0%B0_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%B0), [тригерами](https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B8%D0%B3%D0%B5%D1%80_(%D0%B1%D0%B0%D0%B7%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)) і схемою, яка описує застосунок в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що зв'язує клієнтські компоненти з прикладною логікою бази даних.

Програмна реалізація шару уявлень (клієнтської частини веб-сервісу) здійснена за допомогою адаптивного CSS-фреймворку TwitterBootstrap. Для візуалізації даних використана бібліотека jQuery UI, amCharts. Серверна частина веб-сервісу реалізована за допомогою фреймворку yii2. В якості системи управління базою даних використана MуSQL.

TwitterBootstrap — це клієнтський [фреймворк](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D0%BA%D0%B0%D1%80%D0%BA%D0%B0%D1%81" \o "Програмний каркас), тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на [сервері](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80). [Репозиторій](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D1%96%D0%B9_%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" \o "Репозиторій програмного забезпечення) з даним фреймворком є одним з найбільш популярних на [GitHub](https://uk.wikipedia.org/wiki/GitHub" \o "GitHub).  
jQuery UI — [бібліотека JavaScript](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_JavaScript) з відкритим вихідним кодом для створення [насиченого інтерфейсу користувача](https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%B8%D1%87%D0%B5%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) в [веб-застосунках](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), частина проекту [jQuery](https://uk.wikipedia.org/wiki/JQuery" \o "JQuery). Побудована поверх основної бібліотеки [jQuery](https://uk.wikipedia.org/wiki/JQuery" \o "JQuery) та надає розробникові спрощений доступ до її функцій взаємодії, анімації та ефектів, а також набір [віджетів](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%B6%D0%B5%D1%82" \o "Віджет).   
«AmCharts» - велика онлайн-бібліотека, яка пропонує графіки з візуалізацією різних даних.  
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](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).

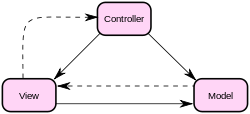
[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:ModelViewControllerDiagram2.svg)

рис. 1.1 «Діаграма взаємодії між компонентами шаблону»

Моде́ль–вигляд–контро́лер (або Модель–представлення–контролер, [англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *Model-view-controller*, MVC) — [архітектурний шаблон](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), який використовується під час проектування та розробки [програмного забезпечення](https://uk.wikipedia.org/wiki/%D0%9F%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).

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: [модель даних](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85), вигляд ([інтерфейс 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)) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

MySQL — [вільна](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BB%D1%8C%D0%BD%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) [система керування реляційними базами даних](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D0%BC%D0%B8_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).

MySQL — компактний [багатопотоковий](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D1%96%D1%81%D1%82%D1%8C" \o "Багатопотоковість) [сервер баз даних](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85&action=edit&redlink=1). Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх застосувань. [Сирцеві коди](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%80%D1%86%D0%B5%D0%B2%D1%96_%D0%BA%D0%BE%D0%B4%D0%B8) [сервера](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80) компілюються на багатьох платформах. Найповніше можливості сервера виявляються в [UNIX](https://uk.wikipedia.org/wiki/UNIX)-системах, де є підтримка [багатопоточності](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D1%96%D1%81%D1%82%D1%8C" \o "Багатопотоковість), що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

* простота у встановленні та використанні;
* підтримується необмежена кількість користувачів, що одночасно працюють із БД;
* кількість рядків у таблицях може досягати 50 млн;
* висока швидкість виконання команд;
* наявність простої і ефективної системи безпеки.

# СТВОРЕННЯ, НАЛАШТУВАННЯ ТА ПРОСУВАННЯ САЙТУ

## **2.1 Вибір засобів проектування web-сайтів**

**2.1.1 Інтегроване середовище розробки PHPStorm**

На сьогоднішній момент існує багато потужних засобів створення програмної продукції. Для розроби цього додатку був обраний PHPStorm, він має дуже зручний інтерфейс, а також він однаково гарно підтримує синтаксис як PHP так і JS.

JetBrains PhpStorm — комерційне крос-платформове [інтегроване середовище розробки](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%BE%D0%B2%D0%B8%D1%89%D0%B5_%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B8) для [PHP](https://uk.wikipedia.org/wiki/PHP" \o "PHP),яке розробляється компанією [JetBrains](https://uk.wikipedia.org/wiki/JetBrains" \o "JetBrains) на основі платформи [IntelliJ IDEA](https://uk.wikipedia.org/wiki/IntelliJ_IDEA" \o "IntelliJ IDEA).

PhpStorm являє собою інтелектуальний редактор для [PHP](https://uk.wikipedia.org/wiki/PHP), [HTML](https://uk.wikipedia.org/wiki/HTML) і [JavaScript](https://uk.wikipedia.org/wiki/JavaScript" \o "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" \o "Рефакторинг) для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3/5.4/5.5/5.6/7.0/7.1 (сучасні і традиційні проекти), 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.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на [Java](https://uk.wikipedia.org/wiki/Java" \o "Java). Користувачі можуть розширити функціональність середовища розробки за рахунок установки [плаґінів](https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D2%91%D1%96%D0%BD" \o "Плаґін), розроблених для платформи IntelliJ, або написавши власні плаґіни.

Вся функціональність [WebStorm](https://uk.wikipedia.org/wiki/WebStorm" \o "WebStorm) включена в PhpStorm.

2.1.2 Система керування версіями та місце зберігання коду

Git — [розподілена система керування версіями](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D0%BF%D0%BE%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D1%96%D1%8F%D0%BC%D0%B8) [файлів](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) та спільної роботи. Git є однією з найефективніших, надійних і високопродуктивних [систем керування версіями](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D1%96%D1%8F%D0%BC%D0%B8), що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті [гілок](https://uk.wikipedia.org/wiki/%D0%93%D1%96%D0%BB%D0%BA%D0%B0_(%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D1%96%D1%8F%D0%BC%D0%B8)). Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка [цифрових підписів](https://uk.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BF%D1%96%D0%B4%D0%BF%D0%B8%D1%81) розробників до тегів і комітів.

Віддалений доступ до [репозиторіїв](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D1%96%D0%B9" \o "Репозиторій) Git забезпечується git-[демоном](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D0%BC%D0%BE%D0%BD_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0)), [SSH](https://uk.wikipedia.org/wiki/SSH) або [HTTP](https://uk.wikipedia.org/wiki/HTTP) [сервером](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80). [TCP](https://uk.wikipedia.org/wiki/TCP" \o "TCP)-сервіс git-daemon входить у дистрибутив Git і є разом з SSH найпоширенішим і надійним методом доступу. Метод доступу HTTP, незважаючи на ряд обмежень, дуже популярний в контрольованих мережах, тому що дозволяє використання існуючих конфігурацій мережевих фільтрів.  
В своїй базі Git зберігає все по хешам файлів. Як хешуюча функція використовується [SHA-1](https://uk.wikipedia.org/wiki/SHA-1). Перед кожним збереженням файлів Git обчислює [SHA-1](https://uk.wikipedia.org/wiki/SHA-1) [хеш](https://uk.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88) файлу і отриманий хеш стає індексом файлу в Git. Використовуючи [хеш](https://uk.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88) Git легко відслідковує зміни в файлах.

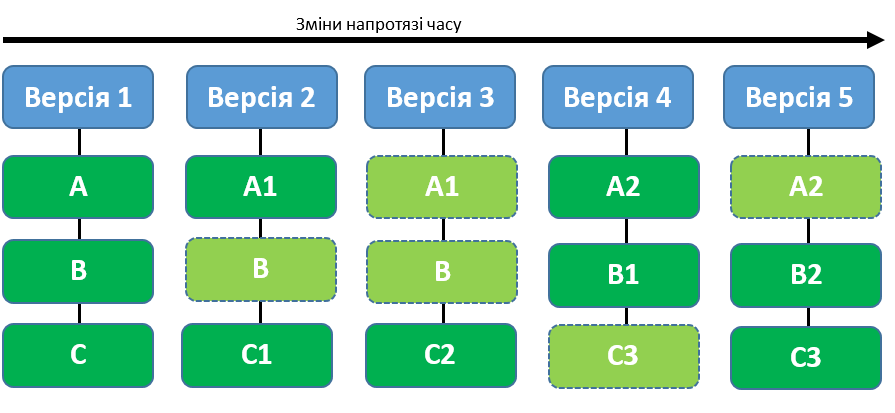


Рис. 2.1 «Дані як зліпки стану проекту в часі»

Для зберігання проекту був вибраний веб-сервіс Bitbucket.  
Bitbucket — веб-сервіс для [хостингу](https://uk.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) проектів на базі [систем керування версіями](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D1%96%D1%8F%D0%BC%D0%B8): [Mercurial](https://uk.wikipedia.org/wiki/Mercurial" \o "Mercurial) та [Git](https://uk.wikipedia.org/wiki/Git" \o "Git) . Bitbucket надає як безкоштовні так і платні послуги. Bitbucket є аналогом [GitHub](https://uk.wikipedia.org/wiki/GitHub" \o "GitHub), проте на відміну від [GitHub](https://uk.wikipedia.org/wiki/GitHub" \o "GitHub), у якого при безкоштовному профілі файли зберігаються лише у відкритому доступі, Bitbucket дозволяє безкоштовно створювати приватні репозиторії з можливостю спільної роботи з файлами до 5-ти користувачів.

**todo Composer**

<p align="center">

<a href="https://github.com/yiisoft" target="\_blank">

<img src="https://avatars0.githubusercontent.com/u/993323" height="100px">

</a>

<h1 align="center"> Yii 2 Основний шаблон проекту </h1>

<br>

</p>

Yii 2 Основний шаблон проекту - це скелет [Yii 2] (http://www.yiiframework.com/), що найкраще підходить для

Створюючи невеликі проекти.  
Шаблон містить основні функції, включаючи вхід користувача / вихід із системи та сторінку контактів. Вона включає в себе всі загальнодоступні конфігурації, які дозволяють зосередитись на додаванні нових особливості програми.

[![Остання стабільна версія](https://poser.pugx.org/yiisoft/yii2-app-basic/v/stable.png)](https://packagist.org/packages/yiisoft/yii2-app-basic)

[![Загальне завантаження](https://poser.pugx.org/yiisoft/yii2-app-basic/downloads.png)](https://packagist.org/packages/yiisoft/yii2-app-basic)

[![Статус збірки](https://travis-ci.org/yiisoft/yii2-app-basic.svg?branch=master)](https://travis-ci.org/yiisoft/yii2-app-basic)

## СТРУКТУРА КАТАЛОГУ

assets/ містить визначення активів

commands/ містить консольні команди (контролери)

config містить конфігурації програми

controllers/ містить класи веб-контролера

mail/ містить файли перегляду для електронних повідомлень

models/ містить моделі класів

runtime/ містить файли, створені під час виконання

tests/ містить різні тести для базової програми

vendor/ містить залежні сторонні пакети

views/ містить файли перегляду для веб-програм

web/ містить сценарій входу та веб-ресурси

ВИМОГИ  
Мінімальна вимога цього шаблону проекту, що ваш веб-сервер підтримує PHP 7.0.

ВСТАНОВЛЕННЯ  
Встановити через Composer — [менеджер пакетів прикладного рівня](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BF%D0%B0%D0%BA%D1%83%D0%BD%D0%BA%D0%B0%D0%BC%D0%B8) для [мови програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [PHP](https://uk.wikipedia.org/wiki/PHP) що забезпечує стандартний формат для управління залежностями у програмному забезпеченні та необхідними [бібліотеками](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).  
Якщо у вас немає [Composer] (http://getcomposer.org/), ви можете встановити його, дотримуючись інструкцій.

[getcomposer.org] (http://getcomposer.org/doc/00-intro.md#installation-nix).

Потім ви можете встановити цей шаблон проекту за допомогою наступної команди:

php composer.phar create-project --prefer-dist --stability=dev yiisoft/yii2-app-basic basic

Тепер ви маєте можливість отримати доступ до програми за допомогою такої URL-адреси, вважаючи, що `basic` це каталог

безпосередньо під веб-кореневим каталогом.

http://localhost/basic/web/

Встановити з архіву  
Витяг архівного файлу, завантаженого з [yiiframework.com] (http://www.yiiframework.com/download/) на каталог з ім'ям `basic`, який знаходиться безпосередньо під веб-кореневим каталогом.

Встановити файл перевірки cookie у файлі `config / web.php` до якогось випадкового секретного рядку

php

'request' => [

// !!! вставте секретний ключ у наведений нижче приклад(якщо він порожній) - це потрібно для перевірки файлів cookie

'cookieValidationKey' => '< Секретний випадковий рядок >',

],

Потім ви можете отримати доступ до програми за допомогою такої URL-адреси:

http://localhost/basic/web/

## КОНФІГУРАЦІЯ

База даних

Відредагуйте файл `config / db.php` з реальними даними, наприклад:

php

return [

'class' => 'yii\db\Connection',

'dsn' => 'mysql:host=localhost;dbname=yii2basic',

'username' => 'root',

'password' => '1234',

'charset' => 'utf8',

];

ПРИМІТКИ  
- Yii не створює базу даних, це потрібно зробити вручну, перш ніж отримувати доступ до неї.

- Необхідно перевірити та відредагувати інші файли в каталозі `config /`, щоб налаштувати програму, як потрібно.

## ТЕСТУВАННЯ

Тести містяться в каталозі `tests`. Вони розроблені з [Codeception PHP Testing Framework] (http://codeception.com/).

За умовчанням існують 3 тестові набори:

- `unit` (Одиничні тести)

- `functional` (Функціональні тести)

- `acceptance` (Тести прийому)

Тести можуть бути виконані запуском

vendor/bin/codecept run

Наведена вище команда виконуватиме одиничні та функціональні тести. Одиничні тести - тестування компонентів системи, в той час як функціональні тести для тестування взаємодії з користувачем. Тести прийому за замовчуванням відключені, оскільки вони потребують додаткового налаштування вони виконують тестування в реальному браузері.

Запуск приймальних випробувань

Для виконання приймальних випробувань виконайте такі дії

1. Перейменуйте `tests / acceptance.suite.yml.example` на` tests / acceptance.suite.yml`, щоб дозволити налаштування пакету

2. Замініть пакет `codeception / base` в` composer.json` за допомогою `codeception / codeception` для встановлення повнофункціональної   версії Codeception

3. Оновити залежності з Composer

     composer update

4. Завантажте [Selenium Server] (http://www.seleniumhq.org/download/) і запустіть його:    java-jar ~ / selenium-server-standalone-x.xx.x.jar

    У разі використання Selenium Server 3.0 з браузером Firefox від v48 або Google Chrome з V53 необхідно завантажити [GeckoDriver] (https://github.com/mozilla/geckodriver/releases) або [ChromeDriver] (https://sites.google .com / a / chromium.org / chromedriver / downloads) і запустити Selenium з нею:

для Firefox: java -jar -Dwebdriver.gecko.driver=~/geckodriver ~/selenium-server-standalone-3.xx.x.jar

для Google Chrome: java -jar -Dwebdriver.chrome.driver=~/chromedriver ~/selenium-server-standalone-3.xx.x.jar   
Як альтернативний спосіб можливе використання вже налаштованного контейнер Docker зі старими версіями Selenium та Firefox

docker run --net=host selenium/standalone-firefox:2.53.0

5. (Необов'язково) Створіти базу даних `yii2\_basic\_tests` та оновіти її, застосуючи міграції, якщо у вас є :     tests/bin/yii migrate

Конфігурацію бази даних можна знайти за адресою `config / test\_db.php`.

6. Запустити веб-сервер:  tests/bin/yii serve

7. Тепер ви маємо можливість запустити всі доступні тести

   - виконувати всі доступні тести: vendor/bin/codecept run

   -запустити прийомні тести: vendor/bin/codecept run acceptance

   - запускати лише одиничні та функціональні тести: vendor/bin/codecept run unit,functional

Підтримка коду покриття коду  
За умовчанням кодове покриття вимикається у файлі конфігурації `codeception.yml ', щоб розпізнати необхідні рядки, щоб збирати покриття коду. Необхідно запустити свої тести та збирати покриття за допомогою такої команди :

збирати охоплення для всіх тестів: vendor/bin/codecept run -- --coverage-html --coverage-xml

поширювати покриття тільки для одиничного тестування: vendor/bin/codecept run unit -- --coverage-html --coverage-xml

збирати охоплення для одиничних та функціональних випробувань: vendor/bin/codecept run functional,unit -- --coverage-html --coverage-xml

Вихідний код можна побачити в каталозі `tests / \_output '

## Основні можливості додатку

Авторизація та регистрація   
Додаток має відкриту регестрацію , що дозволяє кожному бажаючому створити свій аккаунт для користування.  
Для регестрації необхідно ввести свою пошту та вигадати та ввести пароль.  
Після регестрації користувача автоматично буде перенаправлено до системи.   
Для авторизації є можливість введення логіну та паролю, котрий був створений при регестрації , або використати функціонал регестрації через соціальні мережі.  
При використанні авторизації за допомогою соціальних мереж попередня регестрація у додатку не потрібно.   
  
Можливість керування транзакціями між аккаунтами

Додаток позиціоную себе як засіб для збереження моніторину саме домашньої бухгалтерії та розраховано, що користуватися ним будуть сім’ї, тому додатоу має функціонал котрий дозволяє, управляти транзакціями з різних аккаунтів.  
для цього необхідно кожному з членів сім’ї зареєструватися в додатку, з одного з аккаунтів відправити запрошення на інші аккаунти, використовуючи пошту . Далі усі користувачі, котрі отримали запрошення мають можливість, редагувати, створювати та переглядати транзакції за сімейною бухгалтерією.

### Багатовалютність

Додаток підтримую багато валют. Завдяки інтеграції API Приват Банку, можливе використання усіх валют, котрі використовує банк, якщо нам недостатньо тых валют, є можливість запрограмувати додаток на потрібну нам валюту. Конвертація валют виконується по курсу на даний проміжок часу.   
 Теги та категоріїТеги – це ідентифікатор для категоризації, опису, пошуку даних і завдання внутрішньої структури;  
Категорії – це таксоном

Категорії та теги створюються самим користувачем.

у транзакції може бути лише одна категорія. Категорії мають деревоподібну структуру**,** тобто кожна категорія може мати під категорію, може бути багато рівнів. У транзакції може бути багато тегів. Теги мають зв'язок багато до багатьох.  
   
 **Віджети**

**Інтеграція с Приват Банком**  
  
**Запити,котрі приймає додаток**