

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Ігор ШЕЛЕХОВ  
(підпис)

19 травня 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 - Комп'ютерних наук,  
освітньо-наукової програми «Інформатика»  
на тему: «Інформаційна технологія прогнозування курсу валют»  
здобувача групи ІН.м-11н Теницького Олександра Володимировича

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело.

\_\_\_\_\_ Олександр ТЕНИЦЬКИЙ  
(підпис)

Керівник,  
в.о. завідувача кафедри,  
кандидат технічних наук, доцент

Ігор ШЕЛЕХОВ

\_\_\_\_\_ (підпис)

**Суми – 2023**

**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

**ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**

**на здобуття освітнього ступеня магістр**

зі спеціальності 122 - Комп'ютерних наук, освітньо-наукової програми «Інформатика»  
здобувача групи ІН.м-1 Ін Теницького Олександра Володимировича

1. Тема роботи: «Інформаційна технологія прогнозування курсу валют»  
затверджую наказом по СумДУ від «08» травня 2023 р. № 0475-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 19 травня 2023 року
3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.  
2) Огляд технологій, що використовуються для прогнозування курсу валют. 3) Розробка інтелектуальної системи з прогнозування курсу валют. 4) Аналіз результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_
6. Консультанти до проекту (роботи), із зазначенням розділів проекту, що стосується їх \_\_\_\_\_

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд технологій, що використовуються для прогнозування курсу валют</i>		
3	<i>Розробка інтелектуальної системи з прогнозування курсу валют</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

## АНОТАЦІЯ

**Записка:** 50 стор., 27 рис., 2 табл., 1 додаток, 36 джерел.

**Обґрунтування актуальності теми роботи** – Тема кваліфікаційної роботи є актуальною, оскільки присвячена розв’язанню важливої практичної задачі прогнозування курсу валют шляхом розробки відповідних методів, моделей та інформаційної технології.

**Об’єкт дослідження** — процес прогнозування курсу валют.

**Мета роботи** — розробка інформаційної системи прогнозування курсу валют з використанням побудови графіків на основі математичної моделі.

**Методи дослідження** — алгоритми прийняття рішень і прогнозування подій та інструменти побудови математичних моделей.

**Результати** — розроблено інформаційну систему, яка зчитує дані курсу валют, обробляє їх, надає змогу користувачу робити прогнози курсів валют, формує графіки зміни курсу валют з часом, зберігає дані у файл. Проведено тестування розробки на реальних даних курсів валют українських банків.

ІНФОРМАЦІЙНА СИСТЕМА, ПРИЙНЯТТЯ РІШЕННЯ, ПРОГНОЗУВАННЯ  
КУРСУ ВАЛЮТ, PYTON, PANDAS, ARIMA.

## ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Дослідження актуальності проблеми.....	6
1.2 Аналіз алгоритмів прогнозування.....	6
1.3 Постановка задачі .....	18
2 ВИБІР МЕТОДУ РІШЕННЯ.....	20
2.1 Вибір мови програмування.....	20
2.2 Вибір фреймворку.....	23
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ .....	25
3.1 Формування та обробка вхідних даних .....	25
3.2 Опис програмної реалізації.....	28
3.3 Аналіз результатів .....	34
ВИСНОВКИ .....	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39
ДОДАТОК .....	43

## ВСТУП

**Актуальність.** Сьогодні питання дослідження тенденцій зміни валютного курсу надзвичайно актуальне. Якщо розглядати вплив наукових прогнозів в сфері економіки – то можна з упевненістю сказати, що зниження рівня невизначеності позитивно впливатиме на економічний зріст економіки. Багато самостійних інвесторів, власників бізнесу та інші будуть зацікавлені у використанні інтерактивної інформаційної системи прогнозування зміни валютного курсу зі зручним та зрозумілим графічним інтерфейсом. Таким чином розробка інформаційної системи прогнозування курсів валют буде мати позитивний вплив на розвиток системи наукового прогнозування та мати попит серед учасників економічного ринку країн.

**Об’єкт дослідження.** Процес прогнозування курсу валют.

**Предмет дослідження.** Методологія прогнозування курсу валют з використанням інтегрованої моделі авторегресії.

**Гіпотеза.** Прогнозування курсів валют з побудовою високо точних графіків коливань значень можна досягнути шляхом застосування інформаційної системи, що реалізує модель розподіленого запізнення.

**Наукова новизна.** На відміну від існуючих аналогів інформаційних систем, описане у даній роботі програмне рішення дозволить спрогнозувати максимально точні значення, представити їх у вигляді інтерактивних діаграм, порівнювати тенденції змін значень кількох валют за певний проміжок часу та надавати обґрунтування до інвестицій.

**Структура.** Дане робота складається зі вступу, аналізу публікацій, постановки задачі дослідження, вибір методики та інструментів для рішення поставленої проблеми, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Дослідження актуальності проблеми

Зміна валютних курсів має велике значення для економіки країни та її громадян. Наприклад, зміна курсу може впливати на ціни на імпортовані товари та послуги, на діяльність експортерів, на міжнародні інвестиції та інші аспекти економіки. Одним з підходів до вивчення коливань валютних курсів є аналіз часових рядів. Цей метод полягає в тому, щоб використовувати історичні дані про курси валют та їх коливання, щоб прогнозувати майбутні зміни.

Розробка інформаційної системи дружньої до користувача з побудовою зрозумілих графіків та нотифікацією користувачів у випадку коливань курсу валют може бути корисною для трейдерів та інших учасників фінансового ринку, оскільки дозволяє швидко отримувати інформацію про зміни курсів та реагувати на них вчасно. При цьому розробка такої системи потребує використання наукових методів та алгоритмів, таких як аналіз часових рядів, машинне навчання та інші. Наприклад, для прогнозування змін валютних курсів можуть використовуватись алгоритми глибокого навчання, що базуються на штучних нейронних мережах та дозволяють отримати більш точні прогнози.

Отже, розробка інформаційної системи, що включає графіки та сповіщення користувачів про коливання курсів валют, є перспективною для фінансових учасників і потребує застосування наукових методів та алгоритмів для досягнення більш точних результатів.

## 1.2 Аналіз алгоритмів прогнозування

Згідно з дослідженням [1] збільшення прибутку від капіталовкладень та зниження фінансових ризиків можна досягти шляхом максимально точного розрахунку валютного курсу. Існує безліч методів прогнозування у практиці трейдингу [2]. Але серед тих, що користуються попитом варто виділити метод обліку ПКС, метод обліку економічної стабільності, метод побудови економетричної моделі та метод на основі аналізу часових рядів.

Метод обліку ПКС [3] полягає у використанні купівельної спроможності.

Тобто однакові товари мають ідентичну вартість, при цьому не враховують локацію продажів, витрати на транспортування та проведення операції обміну. За вказаним положенням, курс валют завжди прагне такої зміни, яка компенсує зростання цін, викликане інфляцією. Наприклад, припустимо, що ціни в США збільшаться на 2,5% за поточний рік, тоді як в Австралії через зменшення попиту ціни зростуть лише на 0,5%. У такому випадку інфляційний диференціал складе 2%, адже  $2,5\% - 0,5\% = 2\%$ . Згідно з принципом ПКС, австралійський долар буде оцінюватися вище за долар США на приблизно 2% в цьому прикладі. Якщо припустити, що обмінний курс дорівнює 92 центам США за один австралійський долар, то згідно з теорією купівельної сили, прогнозований курс буде наступним:  $(1 + 0,02) \times (0,92 \text{ долара США за } 1 \text{ австралійський долар}) = 0,938 \text{ долара США за } 1 \text{ австралійський долар}$ .

Метод обліку економічної стабільності [4] - це система економічних заходів, спрямованих на забезпечення стабільності економіки країни в умовах зовнішніх і внутрішніх шоків.

Цей метод включає ряд заходів, таких як контроль за інфляцією, збалансованість бюджету, регулювання процентних ставок та валютного курсу, сприяння інвестиціям, розвитку підприємництва та інноваційних технологій. Метою методу економічної стабільності є зменшення впливу зовнішніх і внутрішніх факторів, що можуть призвести до різких змін економічних показників, таких як інфляція, безробіття, падіння ВВП тощо.

Метод економічної стабільності є важливим для розвитку країни, оскільки дозволяє зберігати стабільність в економіці, зменшує ризик фінансових криз та інших негативних наслідків від зовнішніх і внутрішніх шоків. Цей метод використовується в усіх країнах світу, що мають розвинену економіку, та є однією з ключових складових економічної політики країни.

Прогнозування валютного курсу, засноване на вивченні економічної стабільності країн-емітентів вільно конвертованих валют, може допомогти передбачити динаміку обмінного курсу. Стабільна економіка з високими

темпами зростання може залучити більше іноземних інвестицій, що може підвищити попит на національну валюту та її вартість. Хоча цей метод не дає точних значень курсу, він дозволяє інвесторам отримати загальне уявлення про можливі ринкові тренди та орієнтуватися на них. Даний метод зазвичай використовується в поєднанні з іншими методами для найбільш повної картини зміни валютного ринку.

Метод створення економетричної моделі [5] - це процес розробки статистичної моделі, що описує зв'язок між економічними змінними і використовується для передбачення майбутніх значень цих змінних.

Щоб створити економетричну модель, необхідно визначити, які змінні будуть включені в модель та як вони будуть взаємодіяти між собою. Для цього використовуються методи регресійного аналізу, які дозволяють оцінити статистичний зв'язок між залежною змінною (наприклад, обсягом продажів) та однією або більше незалежними змінними (наприклад, ціною товару, рекламними витратами тощо).

Після визначення змінних і створення регресійної моделі необхідно провести оцінку параметрів моделі, щоб визначити, як кожна з незалежних змінних впливає на залежну змінну. Для оцінки параметрів моделі використовуються методи, такі як метод найменших квадратів або метод максимальної правдоподібності.

За допомогою створеної економетричної моделі можна прогнозувати майбутні значення залежної змінної та оцінювати вплив різних незалежних змінних на цю змінну. Економетричні моделі широко використовуються в економічному аналізі, фінансовому аналізі, маркетингових дослідженнях та інших галузях, де потрібно аналізувати залежності між різними економічними змінними.

Наприклад, можна скласти прогноз зміни обмінного курсу американського долара щодо канадського долара на найближчий рік, використовуючи такі ключові фактори, як різниця процентних ставок Центробанків США та Канади,



різниця темпів підвищення доходів населення в тій та іншій країні, та диференціал темпів зростання ВВП в обох країнах.

Економетрична модель для такого прогнозу може мати вигляд:  $\text{Курс USD/CAD} = a(\text{INT}) + b(\text{IGR}) + c(\text{GDP}) + x$ , де коефіцієнти  $a$ ,  $b$  та  $c$  показують, наскільки сильно впливають перелічені економічні чинники на рух обмінного курсу і його напрям, а значення додаткового фактора  $x$  визначається валютним трейдером.

Метод аналізу часових рядів [6] є статистичним методом дослідження даних, який використовується для аналізу даних, які збираються відповідно до часової шкали. Такі дані містять інформацію про зміну певного показника з часом, наприклад, кількість продажів за місяць, ціну акції на біржі за день або кількість відвідувань веб-сайту за годину.

Основними компонентами методу аналізу часових рядів є:

Тренд - довготривала зміна рівня середнього значення часового ряду.

Сезонність - циклічна зміна величини часового ряду на певних інтервалах (наприклад, щоденна, щотижнева, щомісячна сезонність).

Шум - випадкова складова, яка не може бути пояснена трендом або сезонністю.

Метод аналізу часових рядів включає наступні кроки:

- Візуалізація часового ряду за допомогою графіка.
- Виявлення тренду та сезонності в часовому ряді.
- Розбиття часового ряду на тренд, сезонність та шум.
- Моделювання тренду та сезонності за допомогою математичних методів.
- Прогнозування майбутніх значень часового ряду за допомогою побудованої моделі.

Умовно моделювання часових рядів можна поділити на 2 напрямки: моделювання невинядкової складової в сукупності та розкладання часового ряду на компоненти з моделюванням кожного окремо. Особливо популярним

підходом сьогодні є комбінування різних методів для оптимального результату прогнозування [7-8]. Іншою класифікацією методів моделювання є інтуїтивні та формалізовані (рис. 1.1). Оскільки дослідження орієнтоване на метод прогнозування часових рядів – далі будуть розглядатися лише формалізовані методи. Адже інтуїтивні методи застосовуються лише у випадку коли неможливо врахувати вплив факторів через складність об'єкту дослідження [9].

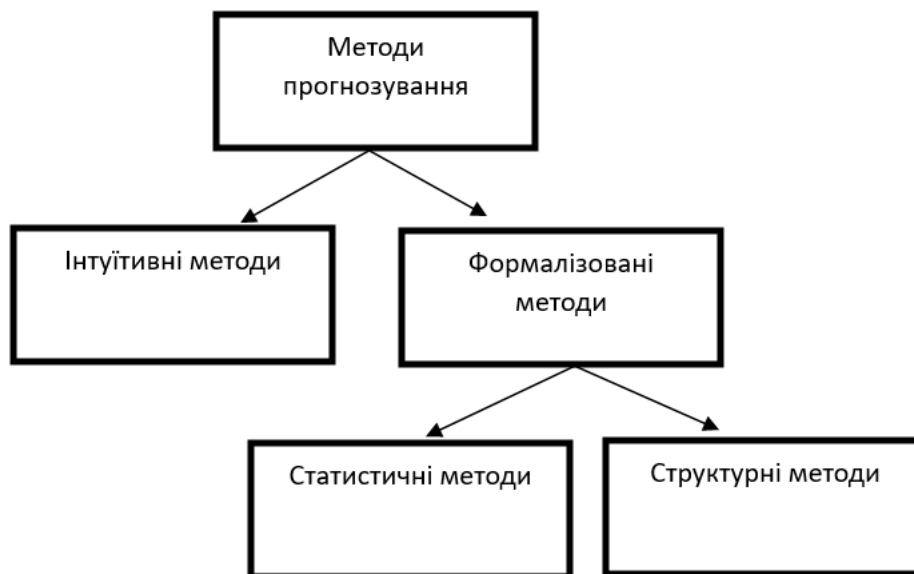


Рисунок 1.1 – Класифікація методів прогнозування

Моделі, що мають функціональну залежність між фактичним та майбутнім значенням ряду й зовнішніми чинниками, яка задана аналітично, називаються статистичними [10]. До цієї групи можна віднести моделі експотенційного згладжування, моделі ковзного середнього, модель максимальної подібності, регресійні та авторегресійні моделі. Визначена структура та правила переходу по ній притаманні структурним моделям: опорні вектори, нейронні мережі, генетичні та мурашині алгоритми, нечітка логіка, ланцюги Маркова та класифікаційно-регресійні дерева [11].

Методи згладжування використовуються в аналізі часових рядів з метою зменшення впливу випадкових коливань та виділення детермінованих даних. Деякі з цих методів спрямовані на згладжування окремих компонент рядів, таких як тренд. Найбільш поширеними моделями згладжування є: просте

експоненційне згладжування (ES), подвійне експоненційне згладжування Брауна (DES), потрійне експоненційне згладжування Брауна (TES), адаптивне згладжування (AS), модель Холта та модель Холта-Вінтера. В результаті застосування цих методів можна отримати більш чисті та точні значення ряду.

Загальну модель можна описати у вигляді:

$$P_t = \alpha X_t + (1 - \alpha)P_{t-1}, \quad (1.1)$$

де  $X_t$  – значення ряду в момент часу  $t$ ,  $t=0..n$ ;

$P_t$  – згладжені значення ряду в момент часу  $t$ ;

$\alpha$  – ваговий коефіцієнт,  $0 < \alpha < 1$

Початкове значення  $P(1)=X(0)$

Перевагами даної моделі можна назвати наступні:

- використання у фінансових та економічних розрахунках;
- простота реалізації;
- одноманітність аналізу;
- ефективна робота в рядах з гладким трендом.

Серед недоліків варто виділити вузьконаправленість, маленька гнучкість та лаговий ефект.

Метод згладжування, що називається ковзне середнє (MA - moving average), має багато модифікацій. Цей метод полягає у побудові нової послідовності, яка містить середнє значення певної кількості попередніх значень. Існують різні варіації методу ковзного середнього, такі як зважене ковзне середнє (WMA - weighted moving average) та експоненційне ковзне середнє (EMA - exponentially moving average). Інші менш поширені види ковзного середнього включають трикутне ковзне середнє (TMA), адаптивне ковзне середнє (AMA), синусоїдально зважене ковзне середнє (SWMA) та ковзне середнє кінцевої точки (ERMA).

Дану модель можна представити у вигляді:

$$P_t = \varepsilon_t + \sum_{j=1}^q \theta_j z_{t-j}, \quad (1.2)$$

де  $q$  – інтервал згладжування;

$\theta_j$  – параметри моделі;

$\varepsilon_t$  – похибка.

Методи згладжування, такі як ковзне середнє, експоненційне згладжування та інші, застосовуються для обробки економічних показників, обробки сигналів у техніці та як технічні індикатори в технічному аналізі. В результаті застосування цих методів, функція тренду може бути наближена до значень ряду, що дозволяє обрати найкращу тенденцію для окремих частин ряду. Хоча методи прості в обчисленнях, вони не ефективні для рядів з лінійною тенденцією та не підходять для перших та останніх рівнів ряду. Регресійні моделі використовуються для вивчення залежності між досліджуваною змінною та зовнішніми факторами, що може бути лінійною, множинною або нелінійною залежністю [12].

Формально регресійна модель виглядає наступним чином:

$$Y = a_0 + \sum_{q=1}^k a_q + E_q, \quad (1.3)$$

де  $a_0, a_q$  – параметри моделі;

$\sum_q F(x_1 \dots x_n)$  – деяка функція від факторів  $x_1 \dots x_n$  (регресор);

$k$  – кількість регресорів.

Серед переваг цих моделей можна виділити:

- доступність аналізу всіх проміжних значень;
- прозорість моделювання;
- гнучкість.

З іншого боку присутні значні недоліки, такі як складність визначення функції залежності, обчислення коефіцієнтів та відсутність майбутніх значень всіх факторів впливу.

Регресійну групу методів можна також доповнити моделлю групового методу обробки даних (group method of data handling, GMDH). Цей метод

використовує опорні функції для створення моделей для кожного аргументу та шляхом перебору знаходить оптимальну модель для кожного набору даних [13].

У загальному вигляді опорну функцію можна зобразити наступним чином:

$$Y(x_1 \dots x_n) = a_0 + \sum_{i=1}^n a_i x_i + \sum_{i=1}^n \sum_{j=i}^n a_{ij} x_i x_j + \sum_{i=1}^n \sum_{j=i}^n \sum_{k=j}^n a_{ijk} x_i x_j x_k + \dots \quad (1.4)$$

де  $a_0, a_i x_i$  – параметри моделі;

$x_1 \dots x_n$  – аргументи.

Метод GMDH утворився в поєднанні переваг регресійного аналізу та штучних нейронних мереж і широко застосовується в комерційних програмах, таких як NeuroShell2, ModelQuest, ASPN, KnowledgeMiner та інші. Авторегресійні моделі відображають процес, в якому поточне значення залежить від попередніх значень цього ж процесу. ARIMA є найбільш поширеною серед них і складається з комбінації функцій авторегресії та ковзного середнього. Ковзне середнє виступає як частотний фільтр, через який проходять значення процесу, і профільтровані значення підлягають авторегресії. Якщо використовуються не самі значення ряду, а їх різниці, то модель називають ARIMA.

Формально модель ARIMA виглядає так [14]:

$$\Delta^d X_t = c + \sum_{i=1}^p a_i \Delta^d X_{t-1} + \sum_{j=1}^q b_j \varepsilon_{t-j} + \varepsilon_t, \quad (1.5)$$

де  $c, a_j, b_j$  – параметри моделі;

$\Delta^d$  – оператор різниці значень часового ряду порядку  $d$ .

Додатково до моделі ARIMA, існують інші моделі авторегресії, такі як умовна гетероскедастичність (GARCH), яка моделює залишки AR, і авторегресійні моделі з розподіленим лагом (ARDLM), яка враховує вплив

попередніх значень на досліджувану змінну. Ці методи мають багато переваг, включаючи точність прогнозування, гнучкість та прозорість моделювання, тому вони широко використовуються в наукових дослідженнях та комерційних програмах. Нові методи та модифікації цих моделей продовжують розвиватися та досліджуватися у наукових публікаціях [15-16].

Одним з недоліків класичних моделей прогнозування є їх низька адаптивність, лінійність процесів та складність ідентифікації параметрів моделі. Для вирішення цих проблем використовуються структурні моделі прогнозування, такі як штучні нейронні мережі, класифікаційно-регресійні дерева, моделі на базі ланцюгів Маркова, моделі на основі нечіткої логіки, опорні вектори, генетичні та мурашині алгоритми [17]. Ці методи орієнтовані на машинне навчання та дозволяють виявляти закономірності всередині процесу. Однією з найбільш популярних моделей прогнозування є штучна нейронна мережа (ANN), яка складається з взаємопов'язаних вузлів (нейронів), що оброблюють та передають дані [18]. ANN може бути одношаровою або багатошаровою, прямого поширення або рекурентною, в залежності від зв'язків між нейронами. Проста схема ANN може виглядати так, як показано на рисунку 1.2.

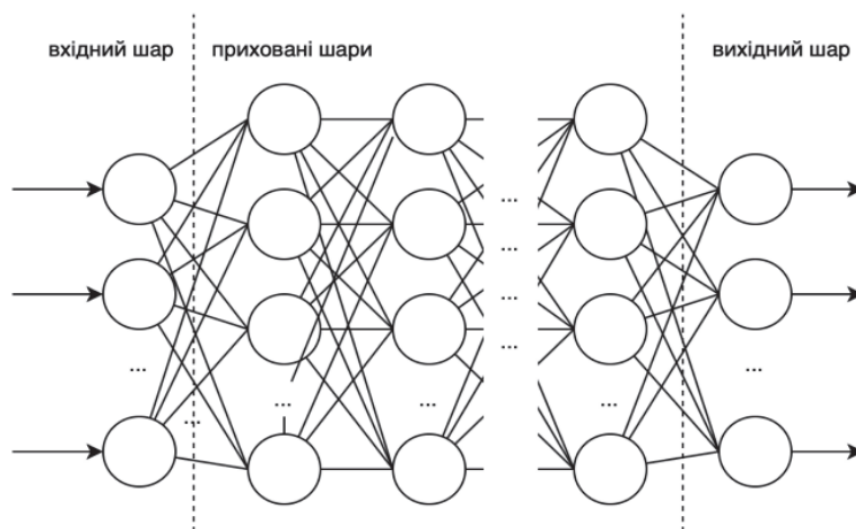


Рисунок 1.2 – Схема багатошарової ANN прямого поширення

Системи здатні самостійно вдосконалювати свої результати, навчаючись

розв'язувати різні задачі. Завдяки цьому вважається, що нейронні мережі дозволяють враховувати “людський фактор” в задачах класифікації та регресії. Головною перевагою цього методу є можливість моделювати складні нелінійні залежності між попередніми значеннями часового ряду та майбутніми станами, тому нейронні мережі застосовуються у багатьох галузях, таких як медицина, автоматизація, геонауки, військові та фінансово-економічні прогнози [19-20]. Крім того, нейронні мережі відрізняються високою точністю прогнозів, а також адаптивністю та можливістю масштабування. Однак основним недоліком нейронних мереж є необхідність їх навчання, яке може бути складним та ресурсомістким процесом, включаючи вибір алгоритму навчання та певні вимоги до навчальної вибірки.

У порівнянні з цими моделями, моделі на основі ланцюгів Маркова (Markov chain model) прогнозують майбутній стан процесу лише залежно від його поточного стану та не враховують попередні значення ряду [21]. Тому для таких моделей характерна коротка пам'ять. Структуру моделей на основі ланцюгів Маркова можна представити у вигляді графу станів з певною ймовірністю переходу між ними (рис. 1.3).

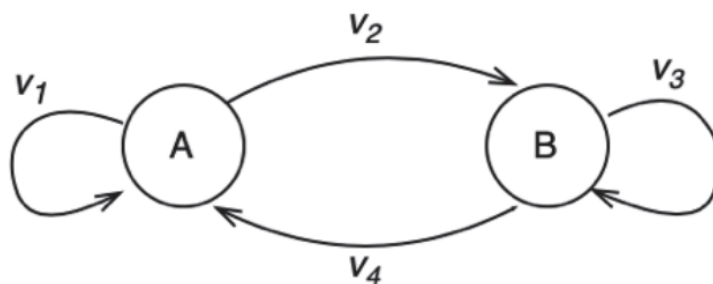


Рисунок 1.3 – Граф ланцюга Маркова з двома станами

A, B – стани процесу;

$v_1, v_2, v_3, v_4$  – ймовірності переходу між станами.

Перевагами моделей на основі ланцюгів Маркова:

- простота в моделюванні випадкових процесів;
- застосовуються в різноманітних сферах життя;

- широко використовуються у фінансовому прогнозуванні та Data Science.

Метод має недолік у вигляді короткої пам'яті, яка не дозволяє враховувати попередні стани в деяких типах задач.

Моделі опорних векторів (SVM) використовуються для задач класифікації шляхом побудови гіперплощини більш високої розмірності, яка максимально віддалена від паралельних прямих, що проходять через опорні вектори. Опорні вектори - це точки у критичній зоні розмежування класів. SVM застосовується в різних галузях, де потрібно проводити класифікацію, наприклад, у медицині, фінансах та інформаційних технологіях [22]. Модель опорних векторів можна зобразити у вигляді схеми для двох класів (рис. 1.4).

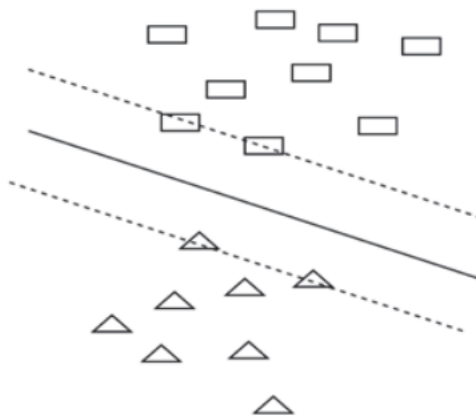


Рисунок 1.4 – Гіперплощини для двох класів на основі опорних векторів

Метод опорних векторів (SVM) застосовують для класифікації даних та вирішення задач розпізнавання образів, створення спам-фільтрів та фінансового прогнозування [23-25]. Його перевагами є простота реалізації та висока точність у випадку задач з двома класами даних. Однак, метод SVM не може прогнозувати майбутні значення для часових рядів, а лише напрям тренду. Класифікаційно-регресійні дерева (CART) використовуються для виведення прогнозованого значення на основі вхідних параметрів. Вони можуть бути деревами регресії або



класифікації та мають форму деревоподібного графа з вузлами прийняття рішень та ребрами (рис. 1.5).

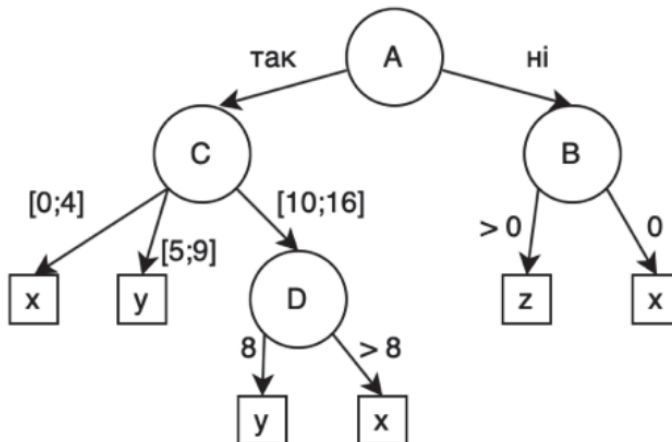


Рисунок 1.5 – Просте дерево рішень

CART є популярним методом в системах підтримки прийняття рішень, аналізу даних та машинному навчанні, який застосовується в різних галузях, таких як банківська справа, промисловість, медицина і економіка. Цей метод відрізняється простотою і гнучкістю моделі, високою точністю прогнозування, використанням невеликої кількості інформації та малим обсягом пам'яті. Однак, недоліками є пошук оптимальної структури дерева та можливість перенавчання.

Останні дослідження націлені на розробку комбінованих моделей та методів, що дозволяють компенсувати недоліки однієї моделі іншою. Комбінації моделей нейронних мереж з нечіткою логікою, з регресією та з ARIMA є одними з перших, що були розроблені. Результати таких прогнозів залежать від сфери застосування та вигляду часового ряду, але вони демонструють більшу точність порівняно з використанням окремих моделей. Комбінації моделей були представлені в дослідженні [26]. Цей підхід широко використовується для прогнозування фінансових показників та торгівлі на біржі.

Отже модель ARIMA є дуже популярною серед валютних трейдерів, що працюють на практиці [27]. Вона базується на тому, що ринок має пам'ять, тому попередня поведінка курсу конкретної валютної пари може бути використана

для прогнозування її поведінки в майбутньому. Цей метод включає побудову цінових графіків та визначення типових графічних зображень.

Враховуючи специфіку поставленого завдання можна зробити висновок, що модель ARIMA є ідеальним варіантом реалізації алгоритму, спираючись на спрощену таблицю 1.1 порівняння моделей для реалізації аналізу часових рядів.

Таблиця 1.1 Порівняння моделей для реалізації аналізу часових рядів

Модель	Опис	Переваги	Недоліки
ARIMA	Autoregressive Integrated Moving Average	Проста у використанні, може давати точні результати для коротких періодів	Може не давати точних результатів для довгих періодів, потребує деяких попередніх знань
LSTM	Long Short-Term Memory	Може давати точні результати для довгих періодів, може працювати з більш складними даними	Вимагає багато часу та обчислювальних ресурсів, може бути складно використовувати
Prophet	Facebook Prophet	Проста у використанні, може враховувати сезонність та інші фактори, які впливають на курс валют	Може давати недостатньо точні результати для коротких періодів
ARIMA-GARCH	Autoregressive Integrated Moving Average - Generalized Autoregressive Conditional Heteroskedasticity	Може враховувати як коливання курсу валют, так і зміну його ризику	Вимагає попередніх знань та навичок у використанні, може бути складно використовувати для некваліфікованих користувачів

### 1.3 Постановка задачі

Метою роботи є розробка програмного забезпечення на основі методу аналізу часових рядів для відстеження змін коливання валютних курсів. Дане дослідження виконується в рамках в рамках кваліфікаційної роботи магістра за спеціальністю 122 «Комп'ютерні науки».

Розроблений програмний продукт має задовольняти наступні вимоги:

- прогнозування курсу валют на два місяці;
- виведення діаграм зміни курсу протягом місяця;

- перегляд зміни значень декількох валют;
- нотифікація на пошту користувача про зміну курсу валюти;
- оцінка та обґрунтування інвестицій у певну валюту на основі тенденції коливання її значень.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- 1) виконати аналіз проблемної області та визначити актуальність;
- 2) провести порівняння методів прогнозування та аналіз аналогів програмних продуктів;
- 3) обрати технології на розробку програмного забезпечення;
- 4) реалізувати модель інформаційної системи.

Предметом дослідження є методологія прогнозування курсу валют з використанням інтегрованої моделі авторегресії.

Наукова новизна полягає в тому, що на відміну від існуючих аналогів інформаційних систем, описане у даній роботі програмне рішення дозволить спрогнозувати максимально точні значення, представити їх у вигляді інтерактивних діаграм, порівнювати тенденції змін значень кількох валют за певний проміжок часу та надавати обґрунтування до інвестицій.

Практичною значимістю буде те, що застосування розроблюваного web-додатку зменшить кількість необхідних ресурсів для аналізу коливання курсів валют та скоротить час на прийняття рішень щодо інвестування.

## 2 ВИБІР МЕТОДУ РІШЕННЯ

### 2.1 Вибір мови програмування

Згідно з даними, що викладені у ТЮВЕ [28] найпопулярнішими мовами програмування на сьогодні є Python, C, Java, C++ та C# (рис. 2.1).






Apr 2023	Apr 2022	Change	Programming Language	Ratings	Change
1	1		 Python	14.51%	+0.59%
2	2		 C	14.41%	+1.71%
3	3		 Java	13.23%	+2.41%
4	4		 C++	12.96%	+4.68%
5	5		 C#	8.21%	+1.39%

Рисунок 2.1 – Топ 5 мов програмування за даними ТЮВЕ

Усі вищезазначені мови програмування використовуються не тільки для розробки програм для вирішення прикладних задач, а й для розробки різноманітних алгоритмів із застосуванням штучного інтелекту. Вибір інструментарію для розробки програмних продуктів безпосередньо залежить від багатьох показників. Таких як швидкодія, вимоги до системи, зручність інтеграції та методи реалізації.

C/C++ [29] є однією з традиційних мов програмування, що часто використовується для динамічного балансування навантаження, адаптивного кешування та розробки великих фреймворків великих даних та бібліотек. Багато бібліотек глибинного навчання були розроблені з використанням C++. Завдяки безумовним перевагам ефективного управління пам'яттю, високої продуктивності та можливості системного програмування, вона є унікальним інструментом для розробки швидко масштабованих бібліотек машинного навчання.

C# [30] є однією з найбільш потужних, динамічних та популярних мов програмування в сфері ІТ. Зараз її використовують для розробки різноманітних

додатків, від невеликих настільних до великих систем та веб-сервісів, що обслуговують мільйони користувачів щодня. Для роботи з машинним навчанням використовується спеціальна бібліотека - ML.NET. Це безкоштовна та відкрита платформа машинного навчання, створена для розробників .NET. За допомогою ML.NET можна розробляти та інтегрувати моделі машинного навчання у програми .NET, не маючи попереднього досвіду в цій галузі. Платформа ML.NET містить інструменти у Visual Studio та міжплатформовий CLI, який підтримує розпізнавання функцій Microsoft, таких як Windows Hello, Bing Ads, PowerPoint Design Ideas та багато інших.

**Java** [31] є однією з найпоширеніших і популярних мов програмування, заснованою на класах та об'єктно-орієнтованому підході, з мінімальними залежностями від конкретної реалізації. Це загально призначена мова програмування, яка дозволяє розробникам писати програми один раз та запускати їх на будь-якій платформі. Java використовується для створення рішень у галузі машинного навчання, нейронних мереж, алгоритмів пошуку, генетичного програмування та мульти-робототехнічних систем. Об'єктно-орієнтованість та масштабованість, є важливими властивостями для ІІ-проектів, і саме тому Java є найкращим вибором для цих завдань. Бізнес вже широко використовує ІІ-технології, тому популярність Java збільшується, оскільки вона дозволяє створювати єдину версію програми, яка буде працювати на декількох платформах.

**Python** [32] є мовою програмування високого рівня загального призначення, що підкреслює читабельність коду завдяки використанню значних відступів. Ця мова має об'єктно-орієнтований підхід та мовні конструкції, що допомагають програмістам писати чіткий логічний код для проектів різного розміру. Python є динамічною мовою зі збиранням сміття і підтримує кілька парадигм програмування, включаючи структуроване, об'єктно-орієнтоване та функціональне програмування. Існує безліч модулів та бібліотек, що дозволяють виконувати різні завдання. Python є ключовим інструментом у розробці

штучного інтелекту та машинного навчання, завдяки наявності фреймворків, таких як scikit-learn, а також є потужним інструментом D3.js.

Критерії порівняння мов програмування для розробки прогнозування зміни курсу валют можуть включати наступні (Таблиця 2.1):

- легкість вивчення та простота синтаксису;
- наявність стандартних бібліотек для математичних операцій та обробки даних;
- швидкодія виконання програми;
- наявність сторонніх бібліотек та фреймворків для машинного навчання та обробки даних;
- наявність інструментів для візуалізації даних та результатів прогнозування.

Таблиця 2.1 Порівняння мов програмування за критеріями

Мова програмування/ Критерій	Python	C	Java	C++	C#
Легкість вивчення	Висока	Середня	Середня	Низька	Середня
Стандартні бібліотеки	Так	Ні	Так	Ні	Так
Швидкодія	Середня	Висока	Середня	Висока	Середня
Бібліотеки та фреймворки	Scikit-learn, Pandas, NumPy	OpenCV, Caffe	Weka, Mahout	TensorFlow, OpenCV	Accord.NET, ML.NET
Інструменти візуалізації	Matplotlib, Seaborn	Gnuplot	JFreeChart	VTK	LiveCharts

За цією таблицею можна зробити висновок, що Python є найкращим вибором для розробки мікросервісу алгоритму прогнозування курсів валют завдяки своїй високій легкості вивчення та простоті синтаксису, наявності стандартних бібліотек для математичних операцій та обробки даних, багатим

вибором сторонніх бібліотек та фреймворків для машинного навчання та обробки даних, а також інструментів для візуалізації даних та результатів прогнозування.

## 2.2 Вибір фреймворку

Існує багато PHP-фреймворків, кожен з яких має свої переваги та недоліки. Ось порівняння кількох популярних фреймворків PHP:

Laravel є одним з найбільш популярних фреймворків PHP [33] (рис.2.2). Він має велику спільноту, яка розробляє додаткові пакети для фреймворку. Laravel також має дуже зручну документацію, яка допомагає швидко розібратися в фреймворку. Одним з найбільших переваг Laravel є його здатність швидко розгорнути веб-додатки.

Symfony також є дуже популярним фреймворком PHP, який відрізняється високим рівнем модульності та розширюваності. Він також має дуже гарну документацію та спільноту, що розробляє додаткові пакети.

CodeIgniter - це легкий фреймворк PHP, який відрізняється високою продуктивністю та швидкістю. Він також має дуже малу вагу та швидко встановлюється.

CakePHP - це фреймворк PHP, який дуже зручний для вирішення проблем, що стосуються безпеки. Він також має вбудований ORM, який допомагає зменшити кількість коду, потрібного для звернення до бази даних.

Існує декілька причин, чому варто обрати Laravel для розробки веб-додатків:

- Зручна та проста робота з базами даних: Laravel забезпечує зручний та інтуїтивно зрозумілий інтерфейс для роботи з базами даних, а також підтримує різні системи управління базами даних, такі як MySQL, PostgreSQL, SQLite та SQL Server.
- Широкі можливості для розширення функціональності: Laravel має велику кількість готових пакетів та бібліотек, що дозволяє значно скоротити час розробки та забезпечити високу якість коду.

- Підтримка моделей MVC: Laravel базується на паттерні MVC, що дозволяє розподілити логіку додатку між моделями, контролерами та представленнями. Це робить розробку більш організованою та підтримує високу розширюваність додатку.
- Зручний та простий маршрутизатор: Laravel має простий та зручний маршрутизатор, який дозволяє легко налаштувати шляхи до різних частин додатку, що спрощує розробку та підтримку коду.
- Безпека: Laravel має вбудовані засоби захисту від хакерських атак, такі як CSRF-токени, хешування паролів та захист від SQL-ін'єкцій. Крім того, Laravel регулярно оновлюється та випускається патчі безпеки, що забезпечує високий рівень безпеки для додатків, побудованих на цьому фреймворку.

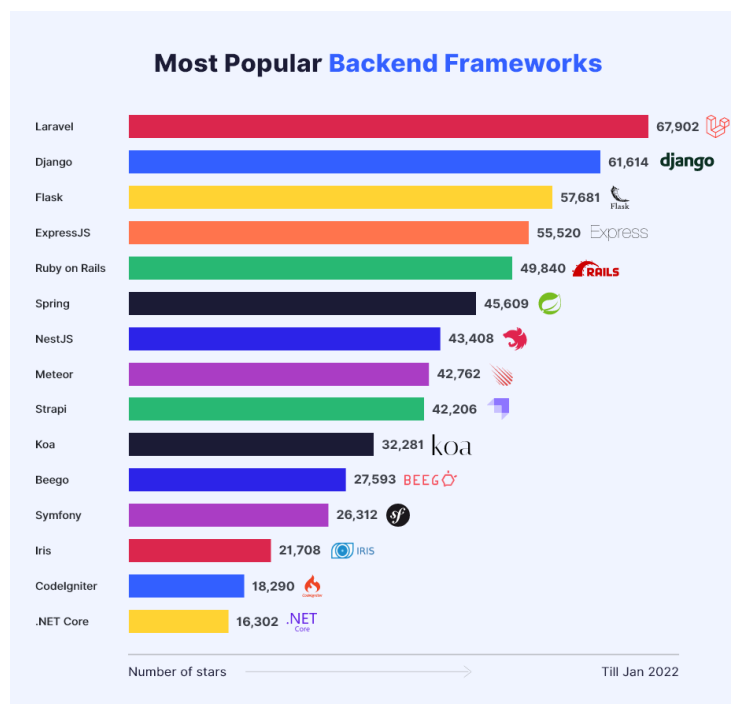


Рисунок 2.2 – Список найпопулярніших PHP фреймворків

Отже, для розробки інформаційної технології прогнозу курсу валют було обрано інструменти: PHP фреймворк Laravel для реалізації бек-енд частини web-додатку, мова програмування Python для розробки мікросервісу прогнозування курсу валют, технологія vue.js для створення графічних даних програми та СУБД MySQL для реалізації бази даних.



### 3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

#### 3.1 Формування та обробка вхідних даних

З метою розробки працюючого алгоритму прогнозування курсів валют, необхідно підготувати набір даних, що містить інформацію про курси валют протягом великого проміжку часу, зокрема, з початку 2000-х років. Ці дані повинні бути збережені в базі даних з наступними колонками:

- date - дата, на яку був зафіксований курс валюти;
- actual - фактичний курс валюти на дату, вказану в колонці "date";
- predicted - порожнє поле, яке буде заповнюватися алгоритмом;
- currency - валюта, до якої належить курс, вказана у відповідній колонці.

	☰ Дата ↕	☰ Офіційний курс гривні, грн ↕
1	03.02.2000	5.4693
2	04.02.2000	5.4748
3	05.02.2000	5.5035
4	06.02.2000	5.5035
5	07.02.2000	5.5035
6	08.02.2000	5.5386
7	09.02.2000	5.5613
8	10.02.2000	5.582100000000000005
9	11.02.2000	5.5975
10	12.02.2000	5.6029
11	13.02.2000	5.6029
12	14.02.2000	5.6029
13	15.02.2000	5.6025
14	16.02.2000	5.595800000000000006

Рисунок 3.1 - Дані про актуальні курси на конкретну дату в форматі CSV

	id	date	actual	predicted	currency
1	2120	2016-05-11	25.2	<null>	USD
2	2121	2016-05-12	25.33	<null>	USD
3	2122	2016-05-13	25.31	<null>	USD
4	2123	2016-05-14	25.31	<null>	USD
5	2124	2016-05-15	25.31	<null>	USD
6	2125	2016-05-16	25.45	<null>	USD
7	2126	2016-05-17	25.39	<null>	USD
8	2127	2016-05-18	25.27	<null>	USD
9	2128	2016-05-19	25.26	<null>	USD
10	2129	2016-05-20	25.26	<null>	USD
11	2130	2016-05-21	25.26	<null>	USD

Рисунок 3.2 - Перенесені дані в базу даних до прогнозування

Ця інформація дозволить системі прогнозувати курси різних валют, а також робити порівняння фактичного та прогнозованого курсів. Для забезпечення точності прогнозування необхідно використовувати якомога більшу кількість даних, що охоплюють різні економічні ситуації та періоди часу.

Спочатку дані знаходяться в CSV файлі, ми їх зчитуємо.

```
# get data
def GetData(fileName):
    return read_csv(fileName, header=0, parse_dates=[0], index_col=0).values
```

Рисунок 3.3 - Зчитування даних з CSV файлу

Після чого в циклі перебираємо кожний актуальний курс на конкретну дату та розраховуємо прогнозований курс.

```

NumberOfElements = len(ActualData)

# Use 70% of data as training, rest 30% to Test model
TrainingSize = int(NumberOfElements * 0.7)
TrainingData = ActualData[0:TrainingSize]
TestData = ActualData[TrainingSize:NumberOfElements]
TestDates = ActualDates[TrainingSize:NumberOfElements]

# new arrays to store actual and predictions
Actual = [x for x in TrainingData]
Predictions = list()

# in a for loop, predict values using ARIMA model
for timepoint in range(len(TestData)):
    ActualValue = TestData[timepoint]
    # forecast value
    Prediction = StartARIMAForecasting(Actual, 3, 1, 0)

    print('Date=%s, Actual=%f, Predicted=%f' % (TestDates[timepoint], ActualValue, Prediction))
    # add it in the list
    Predictions.append(Prediction)
    Actual.append(ActualValue)

insert_query = "INSERT INTO exchange_rates (date, actual, predicted, currency) VALUES (%s, %s, %s, %s)"

date = TestDates[timepoint][0]
date_obj = datetime.strptime(date, "%d.%m.%Y")
date_formatted = date_obj.strftime("%Y-%m-%d")
av = ActualValue[0]
|
record = (date_formatted, av, Prediction, "USD")

```

Рисунок 3.4 - Прогнозування курсу

Після чого в таблиці бази даних зберігаються актуальні курси разом з прогнозованими, таким чином прогнозувати курс можна і на майбутній проміжок часу, головне щоб були тренувальні дані з попередніми курсами.

2120	2016-05-11	25.2	25.16	USD
2121	2016-05-12	25.33	25.2	USD
2122	2016-05-13	25.31	25.34	USD
2123	2016-05-14	25.31	25.32	USD
2124	2016-05-15	25.31	25.33	USD
2125	2016-05-16	25.45	25.31	USD
2126	2016-05-17	25.39	25.47	USD
2127	2016-05-18	25.27	25.4	USD
2128	2016-05-19	25.26	25.28	USD
2129	2016-05-20	25.26	25.24	USD
2130	2016-05-21	25.26	25.24	USD
2131	2016-05-22	25.26	25.26	USD

Рисунок 3.5 - Дані після прогнозування

## 3.2 Опис програмної реалізації

Задля забезпечення реалізації вимог до програмного продукту було використано наступні інструменти:

- Pandas - це відкрите програмне забезпечення для маніпулювання та аналізу даних у мові програмування Python [34]. Воно надає структури даних для ефективної роботи з даними, такі як таблиці, рядки та стовпці.
- `read_csv` - це функція бібліотеки Pandas, яка дозволяє читати дані з файлів CSV та створювати DataFrame, який є основним типом даних у бібліотеці Pandas.
- Matplotlib - це бібліотека для візуалізації даних у мові програмування Python [35]. Вона дозволяє створювати різні типи графіків, діаграм, гістограм та інших візуальних елементів.
- Pyplot - це модуль бібліотеки Matplotlib, який надає інтерфейс для створення графіків та інших візуальних елементів.
- `statmodels.tsa.arima.model` - це модуль бібліотеки Statsmodels, який містить інструменти для аналізу часових рядів та побудови моделей ARIMA (авторегресійно-інтегрованої середньої) для прогнозування часових рядів.
- ARIMA - це модель для аналізу та прогнозування часових рядів, яка враховує авторегресію, диференціювання та ковзне середнє.
- `mysql.connector` - це бібліотека мови програмування Python, яка дозволяє з'єднатися з базою даних MySQL та взаємодіяти з нею [36]. Вона надає інтерфейс для виконання запитів до бази даних та отримання результатів.
- `datetime` - це модуль мови програмування Python, який дозволяє працювати з датами та часом. Він містить класи для роботи зі стандартними форматами дат і часу, а також функції для роботи зі змінними типу дата/час.

```

from pandas import read_csv
from matplotlib import pyplot
from statsmodels.tsa.arima.model import ARIMA
import mysql.connector
from datetime import datetime

```

Рисунок 3.6 – Імпорт бібліотек

Для реалізації алгоритму необхідно створити функцію під назвою "StartARIMAForecasting" (рис 3.7), яка приймає чотири аргументи: Actual, P, D та Q. Actual - це часовий ряд, для якого ми хочемо зробити прогноз, а P, D та Q - це параметри моделі ARIMA, які визначають порядок авторегресії (P), рівня стаціонарності (D) та порядок ковзного середнього (Q) для моделі.

Перш ніж виконати прогноз, функція створює об'єкт моделі ARIMA з використанням параметрів P, D та Q та передає до моделі наш часовий ряд Actual. Після цього функція навчає модель нашому часовому ряду та знаходить найкращі параметри ARIMA моделі. Після навчання моделі функція виконує прогноз на один крок вперед та повертає прогнозоване значення. Отже, цей код створює просту функцію для прогнозування одного кроку вперед за допомогою ARIMA моделі з заданими параметрами P, D та Q.

```

def StartARIMAForecasting(Actual, P, D, Q):
    model = ARIMA(Actual, order=(P, D, Q))
    model_fit = model.fit()
    prediction = model_fit.forecast()[0]
    return prediction

```

Рисунок 3.7 – Функція StartARIMAForecasting

Функція відображення графіку прогнозованих значень та реальних даних зображена на рисунку 3.8. Вона приймає два параметри: `testData` - це набір реальних даних, а `prediction` - це набір прогнозованих даних.

Спочатку у функції закоментовано рядок з обчисленням середньоквадратичної помилки (MSE), яка відображає, наскільки добре модель прогнозує дані. Далі на графіку відображаються реальні дані (`testData`) та прогнозовані дані (`prediction`), при цьому останній відображається червоним кольором. Також функція використовує `pyplot` (модуль `matplotlib`) для відображення графіку на екрані.

```
def ShowPlot(testData, prediction):
    # Print MSE to see how good the model is
    # Error = MeanSquaredError(TestData, Predictions)
    # print('Test Mean Squared Error (smaller the better fit): %.3f' % Error)
    # plot
    pyplot.plot(testData)
    pyplot.plot(prediction, color='red')
    pyplot.show()
```

Рисунок 3.8 – Функція ShowPlot

Функція `Predict()` призначена для прогнозування курсу валют за допомогою моделі ARIMA на основі історичних даних, отриманих з CSV-файлу `dollar_exchange.csv` (рис. 3.9-3.10).

Функція виконує наступні кроки:

1. Отримує історичні дані курсу валюти та дати з CSV-файлу `dollar_exchange.csv`.
2. Розбиває історичні дані на дві частини: 70% для навчання моделі та 30% для перевірки прогнозування.
3. Створює два списки: `Actual`, що містить значення курсу валюти для навчання моделі, та `Predictions`, що буде заповнюватись передбаченими

значеннями курсу валюти для перевірки прогнозування.

4. Виконує цикл, який для кожної дати в перевірочній вибірці прогнозує значення курсу валюти на наступний період за допомогою ARIMA-моделі, використовуючи значення зі списку Actual для навчання моделі.

5. Додає прогнозоване значення до списку Predictions та додатково додає актуальне значення курсу валюти до списку Actual, щоб використовувати його в наступному кроці циклу.

6. Виводить дату, актуальне значення курсу валюти та прогнозоване значення для кожного елемента в перевірочній вибірці, а також виконує запис прогнозованих значень в базу даних MySQL.

7. Викликає функцію ShowPlot(), яка будує графік курсу валюти на основі актуальних значень та прогнозів.

```
def Predict():  
    # Get exchange rates  
    ActualData = GetData('dollar_exchange.csv')  
    ActualDates = GetDates('dollar_exchange.csv')  
  
    # Size of exchange rates  
    NumberOfElements = len(ActualData)  
  
    # Use 70% of data as training, rest 30% to Test model  
    TrainingSize = int(NumberOfElements * 0.7)  
    TrainingData = ActualData[0:TrainingSize]  
    TestData = ActualData[TrainingSize:NumberOfElements]  
    TestDates = ActualDates[TrainingSize:NumberOfElements]  
  
    # new arrays to store actual and predictions  
    Actual = [x for x in TrainingData]  
    Predictions = list()
```

Рисунок 3.9 – Функція Predict

```

# in a for loop, predict values using ARIMA model
for timepoint in range(len(TestData)):
    ActualValue = TestData[timepoint]
    # forecast value
    Prediction = StartARIMAForecasting(Actual, 3, 1, 0)

    print('Date=%s, Actual=%f, Predicted=%f' % (TestDates[timepoint], ActualValue, Prediction))
    # add it in the list
    Predictions.append(Prediction)
    Actual.append(ActualValue)

insert_query = "INSERT INTO exchange_rates (date, actual, predicted, currency) VALUES (%s, %s, %s, %s)"

date = TestDates[timepoint][0]
date_obj = datetime.strptime(date, "%d.%m.%Y")
date_formatted = date_obj.strftime("%Y-%m-%d")
av = ActualValue[0]

record = (date_formatted, av, Prediction, "USD")

# processing query
cursor.execute(insert_query, record)

# commitment of changes in database
cnx.commit()

ShowPlot(TestData, Predictions)

```

Рисунок 3.10 – Функція Predict

Для формування діаграми було розроблено Vue.js компонент, який малює графік, що відображає фактичні та прогнозовані обмінні курси для обраної валюти у заданому діапазоні дат.

У шаблоні компонента ми маємо VueDatePicker, який дозволяє користувачеві вибрати діапазон дат, а також список, в якому можна вибрати валюту для відображення графіка. Нарешті компонент включає елемент canvas, на якому буде відображатися графік.

Блок скриптів компонента починається з імпорту необхідних залежностей, включаючи vue-chartjs, який надає Line Chart компонент, а також axios і vue-underscore.

Далі встановлюються початкові значення для діапазону дат, вибраного користувачем, а також параметри графіка, включаючи дані набору графіків і параметри плагінів.

У методах визначається рендеринг графіка та оновлення за зміни



параметрів вибору дат та валюти. При оновленні графіка компонент виконує GET-запит до API для отримання обмінних курсів для вибраної валюти та діапазону дат. Потім фактичні та прогнозовані курси розбираються та використовуються для оновлення набору даних графіка. Після цього викликається метод рендерингу графіка для відображення оновлених даних.

```

export default {
  components: { Line },
  //props: ["diagram-data", "labels-prop", "group", "department"],
  created() {
    const startDate = new Date();
    const endDate = new Date(new Date().setDate(startDate.getDate() + 7));
    this.dateRange.value = [startDate, endDate];
    axios.get( url: '/webapi/exchange/get', config: {params:
      {
        currency: "USD",
        startDate: "2023-01-01",
        endDate: "2023-06-07",
      }
    }).then((res) => {
      let exchanges = res.data;
      let actual = _.pluck(exchanges, 'actual');
      let predicted = _.pluck(exchanges, 'predicted');
      this.chartData.labels = _.pluck(exchanges, 'date');
      this.chartData.datasets[0].data = actual;
      this.chartData.datasets[1].data = predicted;
    });
  },
},

```

Рисунок 3.11 – Компонент формування діаграми

Програмний код API-контролера, що відповідає за отримання актуальних та прогнозованих курсів валют, за вказаними датами та валютою, зображений на рисунку 3.12.

```

class ExchangeRatesController
{
    public function get(Request $request) {
        $currency = $request->get( key: 'currency');
        $startDate = $request->get( key: 'startDate');
        $endDate = $request->get( key: 'endDate');
        $exchangeRates = ExchangeRate::query()
            ->where( column: 'currency', $currency)
            ->where( column: 'date', operator: '>', $startDate)
            ->where( column: 'date', operator: '<', $endDate)
            ->get();

        return response()->json(ExchangeRatesResource::collection($exchangeRates));
    }
}

```

Рисунок 3.12 – API-контролер для отримання даних

### 3.3 Аналіз результатів

На початку роботи з інформаційною технологією користувач має можливість ознайомитися з даними актуального та прогнозованого курсів валют євро, долара та швейцарського франку на обраний проміжок часу. Для більшої зручності відвідувачів сайту налаштування проміжку дат для прогнозування включає години та хвилини (рис. 3.13). На діаграмі зеленим кольором позначено актуальний курс, а червоним – прогнозований (рис.3.14).

May 2019							Jun 2019						
Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
29	30	1	2	3	4	5	27	28	29	30	31	1	2
6	7	8	9	10	11	12	3	4	5	6	7	8	9
13	14	15	16	17	18	19	10	11	12	13	14	15	16
20	21	22	23	24	25	26	17	18	19	20	21	22	23
27	28	29	30	31	1	2	24	25	26	27	28	29	30

05/09/2019, 14:19 - 06/30/2019, 14:19

Cancel Select

Рисунок 3.13 – Компонент з вказанням діапазону дат

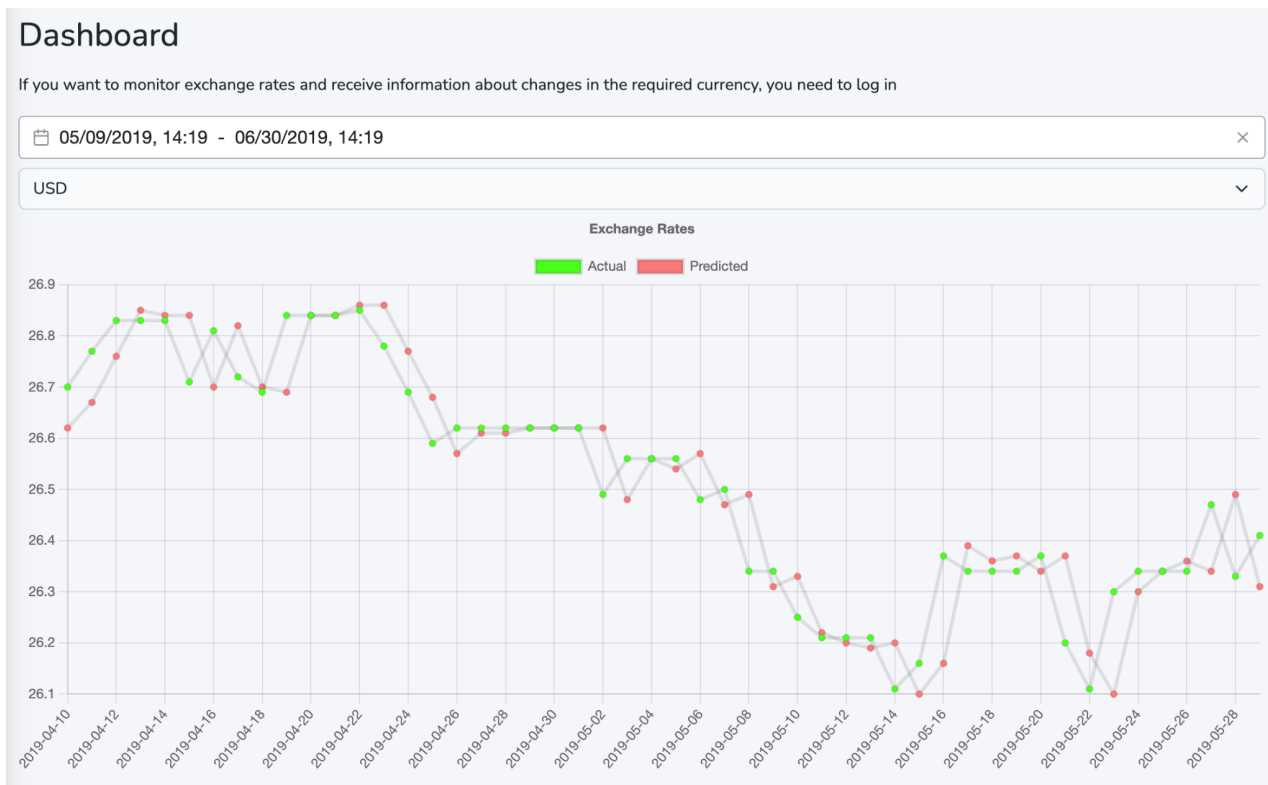


Рисунок 3.14 – Діаграма актуального і прогнозованого курсу вказаної валюти за вказаний період часу

Для того щоб мати можливість доступу до нотифікації про зміну обраного курсу валюти та оцінки змін з обґрунтуванням інвестицій у певну валюту на основі тенденції коливання її значень користувач має бути авторизованим у системі (рис. 3.15).

Login

Email Address

Password

Remember Me

[Forgot Your Password?](#)

Рисунок 3.15 – Сторінка авторизації

Для того щоб перейти до сторінки з планування повідомлення про зміну курсу потрібно перейти за пунктом меню Profile (рис. 3.16).

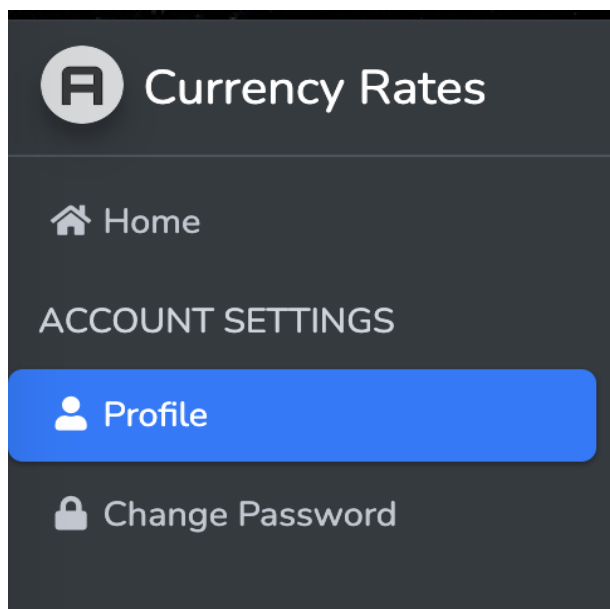
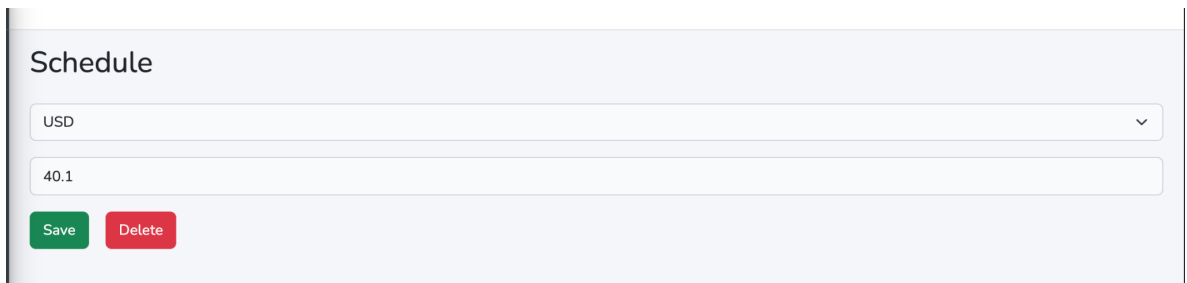


Рисунок 3.16 – Головне меню інформаційної системи

На сторінці налаштування нотифікації треба обрати бажану валюту для відстеження зміни її значень та граничне значення курсу при якому буде надходити повідомлення на пошту користувача. Також на цій сторінці можна переглянути сплановані раніше нотифікації (рис. 3.17-3.18).

Рисунок 3.17 – Додавання нотифікацій, коли курс обраної валюти буде спрогнозовано до вказаного значення



Schedule

USD

40.1

Save Delete

Рисунок 3.18 – Сторінка планування конкретної нотифікації

Щоб перейти до сторінки зі зміною пароля потрібно клацнути по пункту меню Change password (рис. 3.19). Пізніше користувачу буде запропоновано ввести старий та новий паролі (рис. 3.20).

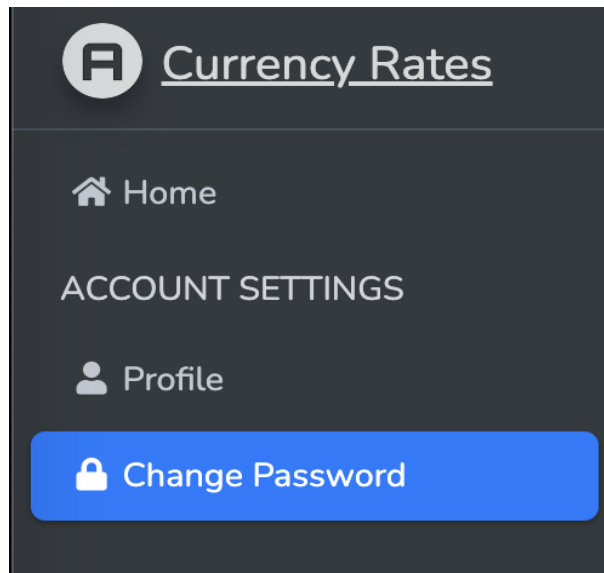
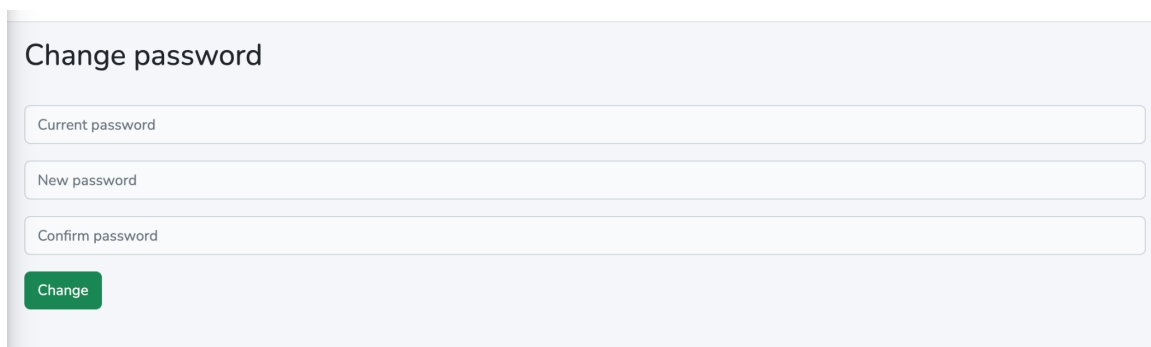


Рисунок 3.19 – Пункт меню Change password



Change password

Current password

New password

Confirm password

Change

Рисунок 3.20 – Сторінка зі зміною пароля

## ВИСНОВКИ

В результаті виконання роботи було створено інформаційну систему на основі методу аналізу часових рядів для відстеження зміни валютних курсів. Було розглянуто різноманітні можливості розробки зазначеного методу та обрано оптимальні інструменти для виконання поставленої мети. Також під час розробки проекту було враховано усі вимоги до програмного продукту.

У ході виконання кваліфікаційної роботи магістра було виконано наступні завдання:

1. Виконано аналіз проблемної області, визначено актуальність розробки системи.
2. Проведено порівняння методів прогнозування та їх моделей.
3. Визначено тестові матеріали на розробку.
4. Обрані технології на розробку програмного продукту.
5. Реалізовано роботу інформаційної системи, що відстежує коливання курсу валют та прогнозує значення на найближче майбутнє.
6. Виконано тестування системи на реальних даних.
7. Проаналізовано отримані результати та зроблено висновки щодо якості роботи інформаційної технології.

За результатами роботи інформаційної системи, можна зробити висновок, що застосування розробленого web-додатку зменшить кількість необхідних ресурсів для аналізу коливання курсів валют та скоротить час на прийняття рішень щодо інвестування.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Валютне регулювання у макроекономічній політиці [Електронний ресурс] – Режим доступу до ресурсу: <https://msn.khmnu.edu.ua/mod/page/view.php?id=159047>.
2. Методи прогнозування [Електронний ресурс] – Режим доступу до ресурсу: <https://studies.in.ua/soc-ekzam/3313-metodi-prognozuvannya.html>.
3. Паритет купівельної спроможності [Електронний ресурс] – Режим доступу до ресурсу: <https://library.if.ua/book/115/7751.html>.
4. Кількісна оцінка економічного ризику: методи визначення та обліку [Електронний ресурс] – Режим доступу до ресурсу: [https://osvita.ua/vnz/reports/econom\\_theory/21715/](https://osvita.ua/vnz/reports/econom_theory/21715/)
5. ЕКОНОМЕТРИЧНІ МОДЕЛІ [Електронний ресурс] – Режим доступу до ресурсу: [https://stud.com.ua/91187/investuvannya/ekonometrichni\\_modeli](https://stud.com.ua/91187/investuvannya/ekonometrichni_modeli).
6. Аналіз часових рядів [Електронний ресурс] – Режим доступу до ресурсу: [https://www.wik.uk-ua.nina.az/Аналіз\\_часових\\_рядів.html](https://www.wik.uk-ua.nina.az/Аналіз_часових_рядів.html).
7. Time Series Analysis and Forecasting | Data-Driven Insights (Updated 2023) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/>.
8. Yang M.Sc. Power System Short-term Load Forecasting: Thesis for Ph.d degree / M.Sc. Yang. – Darmstadt, Germany, Elektrotechnik und Informationstechnik der Technischen Universitat, 2006. – 139 p.
9. 5.3. Методи прогнозування макроекономічних процесів Сутність методів прогнозування [Електронний ресурс] – Режим доступу до ресурсу: <https://buklib.net/books/24507/>.
10. Кількісні (статистичні) методи прогнозування [Електронний ресурс] – Режим доступу до ресурсу: <https://msd.in.ua/kilkisni-statistichni-metodi>

prognozuvannya/.

11. Андрусенко Ю. О. АНАЛІЗ ОСНОВНИХ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ / Ю. О. Андрусенко. // Збірник наукових праць Харківського національного університету Повітряних Сил. – 2020. – С. 91–96.

12. РЕГРЕСІЙНИЙ АНАЛІЗ [Електронний ресурс] – Режим доступу до ресурсу: [https://stud.com.ua/140825/prirodoznavstvo/regresiyuiy\\_analiz](https://stud.com.ua/140825/prirodoznavstvo/regresiyuiy_analiz).

13. 3. Метод групового врахування аргументів [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/4494701/page:8/>.

14. Time Series Analysis - ARIMA models - The behavior of non stationary time series [Електронний ресурс] – Режим доступу до ресурсу: [https://www.xycoon.com/behavior\\_of\\_non\\_stationary\\_time\\_.htm](https://www.xycoon.com/behavior_of_non_stationary_time_.htm).

15. Forecast of the trend in incidence of acute hemorrhagic conjunctivitis in China from 2011-2019 using the Seasonal Autoregressive Integrated Moving Average and Exponential Smoothing models / L. Huan, L. Chenxi, S. Yingqi, Z. Xin // Journal of Infection and Public Health. – 2020. – № 13. – P. 287-294.

16. Литвиненко В.І. Метод прогнозування гетероскедастичних процесів з використанням синтезованих поліноміальних нейронних мереж / В.І. Литвиненко, О.А. Кожухівська, А.О. Фефелов // Вісник Національного університету “Львівська політехніка”. – 2015. – № 829. – С. 201-214

17. Bhadeshia H.K. Neural Networks in Materials Science / H.K. Bhadeshia // ISIJ International. –1999. – № 39(10). – P. 966-979.

18. ШТУЧНА НЕЙРОННА МЕРЕЖА (ANN) [Електронний ресурс] – Режим доступу до ресурсу: <https://ua.nesrakonk.ru/artificial-neural-networks-ann/>.

19. Intelligent Deep Learning Method for Forecasting the Health Evolution Trend of Aero-Engine With Dispersion Entropy-Based Multi-Scale Series Aggregation and LSTM Neural Network / J.Wei, Z. Nan, X. Xiaoming, X. Yanhe // IEEE Access. – 17 February 2020. – P. 34350-34361.

20. Parisi G.I. Continual lifelong learning with neural networks: A review /



- G.I. Parisi, R. Kemker, J.L. Part // *Neural Networks*. – 2019. – № 113. – P. 54-71.
21. Afrinaldi F. Exploring product lifecycle using Markov chain / F. Afrinaldi // *Procedia Manufacturing*. – 2020. – № 43. – P. 391-398.
  22. Burges C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery* 2:121–167, 1998
  23. El-Sayed M. Silhouette-Based Gender Recognition in Smart Environments Using Fuzzy Local Binary Patterns and Support Vector Machines / M. El-Sayed, El-Alfy // *Procedia Computer Science*. – 2017. – № 109. – P. 164-171.
  24. Evolving Support Vector Machines using Whale Optimization Algorithm for spam profiles detection on online social networks in different lingual contexts / M. Ala, A.Z. Hossam, F.Alqatawnaabc, M.A. Haddonaha // *Knowledge-Based Systems*. – 2018. – № 153. – P. 91-104.
  25. Hitam N.A. An Optimized Support Vector Machine (SVM) based on Particle Swarm Optimization (PSO) for Cryptocurrency Forecasting / N.A. Hitam // *Procedia Computer Science*. – 2019. – № 163. – P. 427-433.
  26. Wongsathan R. A hybrid ARIMA and Neural Networks model for PM-10 pollution estimation: The case of Chiang Mai city moat area / R. Wongsathan, I. Seedadan // *Procedia Computer Science*. – 2016. – № 86. – P. 273-276.
  27. Autoregressive Integrated Moving Average (ARIMA) Prediction Model [Електронний ресурс] – Режим доступу до ресурсу: <https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp>.
  28. TIobe Index for April 2023 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tiobe.com/tiobe-index/>.
  29. Огляд і основи мови програмування C++ [Електронний ресурс] – Режим доступу до ресурсу: [http://www.znannya.org/?view=Cplusplus\\_basics](http://www.znannya.org/?view=Cplusplus_basics).
  30. Що таке C#? Чи підходить мені ця мова програмування? Чому вона крута? [Електронний ресурс] – Режим доступу до ресурсу: <https://beetroot.academy/blog/courses/what-is-csharp>.

31. Чому Java – найпопулярніша мова програмування у світі [Електронний ресурс] – Режим доступу до ресурсу: <https://java.lviv.ua/chomu-java-najpopulyarnisha-mova-programuvannya-u-sviti>.

32. Що таке Python? [Електронний ресурс] – Режим доступу до ресурсу: <http://www.plug.org.ua/documentation/about-python>.

33. Why Choose Laravel as the Best PHP Framework for Developing Applications in 2023? [Електронний ресурс] – Режим доступу до ресурсу: <https://webmobtech.com/blog/why-choose-laravel-as-the-best-php-framework/>.

34. What Is Pandas in Python? Everything You Need to Know [Електронний ресурс] – Режим доступу до ресурсу: <https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/>.

35. Matplotlib: Visualization with Python [Електронний ресурс] – Режим доступу до ресурсу: <https://matplotlib.org/>.

36. MySQL Connectors [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/connectors/en/connectors-apis.html#:~:text=MySQL%20Connectors%20provide%20connectivity%20to,MySQL%20protocol%20or%20X%20Protocol>.

## ДОДАТОК

#ProfileController - відповідає за зміну пароля, відображення, створення, оновлення планувань для нотифікацій щодо зміни бажаного курсу

```
<?php
namespace App\Http\Controllers;
use App\Models\Schedule;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

class ProfileController extends Controller
{
    public function index(Request $request)
    {
        $user = $request->user();
        $schedules = Schedule::query()->where('user_id', $user->id)->get();
        return view('profile', ['schedules' => $schedules]);
    }

    public function settings()
    {
        return view('settings');
    }

    public function changePassword(Request $request)
    {
        $request->validate([
            'current_password' => 'required',
            'new_password' => 'required|string|min:8|confirmed',
        ]);

        $user = Auth::user();
        if (!Hash::check($request->current_password, $user->password)) {
            return redirect()->back()->withErrors(['current_password' => 'Current password is incorrect.'])-
                >withInput();
        }

        $user->password = Hash::make($request->new_password);
        $user->save();

        return redirect()->back()->with('success', 'Password changed successfully.');
```

```
    }

    public function saveSchedule(Request $request)
    {
        $attributes = $request->validate([
            'currency' => 'required|in:USD,EUR,CHF',
            'rate' => 'required',
        ]);
        $attributes['user_id'] = $request->user()->id;
        $schedule = new Schedule($attributes);
```

```

    $schedule->save();
    return redirect()->route("schedule.show", $schedule);
}

public function updateSchedule(Schedule $schedule, Request $request)
{
    $attributes = $request->validate([
        'currency' => 'required|in:USD,EUR,CHF',
        'rate' => 'required',
    ]);

    $schedule->update($attributes);
    return redirect()->route("schedule.show", $schedule);
}

public function showSchedule(Schedule $schedule)
{
    return view('schedule', ["schedule" => $schedule]);
}

public function deleteSchedule(Schedule $schedule)
{
    $schedule->delete();
    return redirect()->route('profile');
}
}

#Шаблон головної сторінки

@extends('adminlte::page')
@section('title', 'Dashboard')

@section('content_header')
    <h1>Dashboard</h1>
@stop

@section('content')
    @if(auth()->guest())
        <p>If you want to monitor
        exchange rates and receive
        information about changes in
        the required currency, you need
        to log in</p>
    @endif
    <currency-
    diagram></currency-diagram>
@stop

@section('css')
    <link rel="stylesheet"
    href="/css/admin_custom.css">

```

```

@stop

@section('js')

@vite(['resources/sass/app.scss'
, 'resources/js/app.js'])
@stop

#Компонент з діаграмою

<template>
  <VueDatePicker v-
model="dateRange" range
multi-calendars :month-change-
on-scroll="false"
@update:model-
value="onConditionChange"></
VueDatePicker>

  <select class="form-select mt-
2" v-model="currency"
@change="onConditionChange"
>
    <option value="USD"
selected>USD</option>
    <option
value="EUR">EUR</option>
    <option
value="CHF">CHF</option>
    <option
value="BPH">BPH</option>
  </select>
  <div>
    <canvas ref="chart"
style="height:
500px;"></canvas>
  </div>
</template>
<script>
import {Line} from "vue-
chartjs";
import {
  Chart as ChartJS,
  Title,
  Tooltip,
  Legend,
  BarElement,
  CategoryScale,
  LinearScale,
  PointElement,
  LineElement
} from 'chart.js'

```

```

ChartJS.register(Title, Tooltip,
Legend, BarElement,
CategoryScale, LinearScale,
PointElement, LineElement)
import axios from "axios";
import {_} from 'vue-
underscore';

export default {
  components: { Line },
  //props: ["diagram-data",
"labels-prop", "group",
"department"],
  created() {
    const startDate = new
Date();
    const endDate = new
Date(new
Date().setDate(startDate.getDat
e() + 7));
    this.dateRange.value =
[startDate, endDate];

    axios.get('/webapi/exchange/ge
t', {params:
      {
        currency: "USD",
        startDate: "2023-01-
01",
        endDate: "2023-06-
07",
      }
    }).then((res) => {
      let exchanges = res.data;
      let actual =
_.pluck(exchanges, 'actual');
      let predicted =
_.pluck(exchanges, 'predicted');
      this.chartData.labels =
_.pluck(exchanges, 'date');

      this.chartData.datasets[0].data
= actual;

      this.chartData.datasets[1].data
= predicted;
    });
  },
  data() {
    return {
      currency: "USD",
      dateRange: {},
      chartData: {

```

```

      datasets: [
        {
          label: 'Actual',
          backgroundColor:
'#4aff1e',
        },
        {
          label: 'Predicted',
          backgroundColor:
'#f87979',
        }
      ],
    },
    chartOptions: {
      responsive: true,
      maintainAspectRatio:
false,
      plugins: {
        title: {
          display: true,
          text: 'Exchange
Rates',
        },
      },
    },
  },
}
},
mounted() {
  this.renderChart()
},
methods: {
  renderChart() {
    this.chart = new
ChartJS(this.$refs.chart, {
      type: 'line',
      data: this.chartData,
      options:
this.chartOptions,
    });
  },
  onConditionChange() {
    let dateStart = "2023-01-
01"
    let dateEnd = "2023-06-
07"

    if (this.dateRange.length
=== 2) {
      dateStart =
`${this.dateRange[0].getFullYear()}-
${this.dateRange[0].getMonth()}`
    }
  }
}
}

```

```

    }-
    `${this.dateRange[0].getDate()}`
      dateEnd =
    `${this.dateRange[1].getFullYear()}`-
    `${this.dateRange[1].getMonth()}`-
    }-
    `${this.dateRange[1].getDate()}`
      }

```

```

    axios.get('/webapi/exchange/get', {params:
      {
        currency:
    this.currency,
        startDate:
    dateStart,
        endDate: dateEnd,
      }
    }).then((res) => {
      this.chart.destroy();
      let exchanges =
    res.data;
      let actual =
    _.pluck(exchanges, 'actual');
      let predicted =
    _.pluck(exchanges, 'predicted');
      this.chartData.labels =
    _.pluck(exchanges, 'date');

    this.chartData.datasets[0].data
    = actual;

    this.chartData.datasets[1].data
    = predicted;

      this.renderChart()
    });
  }
}
}
</script>

```

```

<style scoped>
</style>

```

```

#Об'явлені маршрути
<?php

```

```

use
Illuminate\Support\Facades\Route;

```



```

/*
|-----
|
| Web Routes
|-----
|
| Here is where you can register
web routes for your application.
These
| routes are loaded by the
RouteServiceProvider and all of
them will
| be assigned to the "web"
middleware group. Make
something great!
|
*/

```

```
Auth::routes();
```

```
Route::get('/',
[App\Http\Controllers\HomeCo
ntroller::class, 'index'])-
>name('home');
```

```
Route::middleware('auth')-
>group(function() {
    Route::get('/profile',
[App\Http\Controllers\ProfileCo
ntroller::class, 'index'])-
>name('profile');
```

```
    Route::get('/profile/settings',
[App\Http\Controllers\ProfileCo
ntroller::class, 'settings'])-
>name('profile.settings');
```

```
Route::post('/profile/settings/ch
ange',
[App\Http\Controllers\ProfileCo
ntroller::class,
'changePassword'])-
>name('profile.settings.change')
;
```

```
    Route::post('/schedule/store',
[App\Http\Controllers\ProfileC
ontroller::class,
'saveSchedule'])-
>name('schedule.store');
```

```
Route::get('/schedule/{schedule
}/show',
[\App\Http\Controllers\ProfileC
ontroller::class,
'showSchedule'])-
>name('schedule.show');
```

```
Route::post('/schedule/{schemul
e}/update',
[\App\Http\Controllers\ProfileC
ontroller::class,
'updateSchedule'])-
>name('schedule.update');
```

```
Route::delete('/schedule/{sched
ule}/delete',
[\App\Http\Controllers\ProfileC
ontroller::class,
'deleteSchedule'])-
>name('schedule.delete');
});
```

```
Route::group([
    'prefix' => 'webapi',
    'as' => 'webapi.'
], function() {
    Route::get('exchange/get',
[\App\Http\Controllers\Api\Exc
hangeRatesController::class,
'get']->name('exchange.get');
});
```