Слайд 1 (титулка):

Презентую курсову роботу за темою «Додаток, який створює сильні та безпечні паролі», виконував студент групи групи ТВ-32, Семенов Максим Михайлович.

Слайд 2 (актуальність теми):

В сучасному цифровому світі проблема безпеки облікових даних є надзвичайно актуальною. Метою даної роботи було створення веб-додатку, який би дозволяв не тільки генерувати надійні паролі, але й безпечно їх зберігати, використовуючи клієнтське шифрування та унікальний майстер-ключ користувача. Об'єктом розробки стала веб-програма на Node.js та Express, з клієнтською частиною на чистому JavaScript та базою даних MySQL.

Слайд 3 (постановка задачі):

Для досягнення мети було поставлено ряд завдань: реалізувати доступ до сховища за майстер-ключем, розробити модуль генерації паролів, забезпечити клієнтське шифрування даних перед збереженням, створити API для взаємодії клієнта та сервера, а також розробити інтуїтивно зрозумілий користувацький інтерфейс.

Слайд 4 (аналіз предметної області):

Аналіз існуючих рішень показав, що багато з них або покладаються на довіру до сторонніх сервісів, або є складними у використанні. Дана програма, названа "Генератор Паролів", прагне запропонувати альтернативу з клієнтським шифруванням. Для реалізації було обрано Node.js та Express.js за їх гнучкість, JavaScript для клієнтської логіки, та MySQL як надійну СУБД.

Слайд 5 (проєктування системи 1):

Програма розроблена за клієнт-серверною архітектурою з монолітним бекендом. Ключовим методом є клієнтське шифрування на основі динамічної мапи заміни символів, що генерується з майстер-ключа. Також реалізовано алгоритм генерації паролів за параметрами користувача.

Слайд 6 (проєктування системи 2):

Клієнтська частина відповідає за інтерфейс та шифрування, серверна – за API та взаємодію з базою даних. На слайді представлені діаграма компонентів та діаграма послідовності для типового сценарію.

Слайд 7 (проєктування системи 3):

Тут ви можете бачити модель діаграми прецидентів (або Use Case), послідовність генерації паролю користувачем, та діаграму класів «Генератора Паролів».

Слайд 8 (проєктування системи 4):

Серед основних функціональних вимог – генерація, безпечне зберігання, перегляд та видалення паролів. Нефункціональні вимоги акцентують увагу на безпеці даних (клієнтське шифрування, дані в БД тільки зашифровані), продуктивності та зручності використання.

Слайд 9 (реалізація системи 1):

Процес розробки включав аналіз, проектування, розробку серверної та клієнтської частин, інтеграцію та тестування. Використано HTML, CSS, JavaScript на клієнті, Node.js, Express.js, драйвер mysql2 на сервері, та MySQL для бази даних.

Слайд 10 (реалізація системи 2):

На цьому та наступних слайдах представлені скріншоти ключових екранів користувацького інтерфейсу. Тут - екран редагування паролів, та також меню їх генерації.

Слайд 11 (реалізація системи 3):

Тут зображено у наступному порядку: приклад успішно згенерованого паролю, разом із прикладом помилки при невірному вводі, та меню зі списком створених паролів.

Слайд 12 (реалізація системи 4):

Програма має опції додавання свого паролю без генерації, та видалення вже доданих, меню для яких видно на цьому слайді.

Слайд 13 (тестування та верифікація):

Проведено модульне тестування крипто-сервісу за допомогою Jest, яке показало коректну роботу основних операцій. Також виконано ручне функціональне тестування всіх ключових сценаріїв користувача та перевірка API. Система показала стабільну роботу та відповідність заявленим вимогам. "Генератор Паролів" пропонує підхід з клієнтським шифруванням, що є його особливістю порівняно з деякими аналогами.

Слайд 14 (впровадження та супровід):

Розгортання серверної частини передбачає використання віртуальної машини з Node.js та MySQL. Клієнтська частина сервірується тим же сервером. Документація включає опис для користувачів та розробників. Плани на майбутнє включають посилення криптографії та розширення функціоналу.

Слайд 15 (висновки):

В ході курсової роботи було розроблено веб-програму "Генератор Паролів", що реалізує функції генерації надійних паролів та безпечного їх зберігання з клієнтським шифруванням. Мета роботи досягнута, всі поставлені завдання виконані. Програма має практичну цінність як інструмент для створення та управління паролями, а перспективи розвитку включають покращення безпеки та розширення функціоналу.