

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка мобільного додатку KNUTE App»

Студента 2 курсу, 5м групи,
спеціальності
121 «Інженерія програмного
забезпечення»
спеціалізації
«Інженерія програмного
забезпечення»

підпис студента

Проява Владислав
Вікторович

Науковий керівник
доктор технічних наук,
професор

підпис керівника

Цюцюра Світлана
Володимирівна

Гарант освітньої програми
доктор технічних наук,
професор

підпис керівника

Криворучко Олена
Володимирівна

КИЇВ – 2019

Зміст	
ВСТУП.....	4
РОЗДІЛ 1. ПЛАТФОРМА ТА ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ ДОДАТКУ «KNUTE APP»	7
1.1 Основні відомості про Android та програмне забезпечення.....	7
1.2 Структура операційної системи Android	10
1.3 Особливості розробки крос-платформних додатків	11
1.4 Висновок до розділу 1	14
РОЗДІЛ 2. ТЕХНОЛОГІЇ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ...15	
2.1 Функція додатку.....	15
2.2 Технології розробки додатку.....	15
2.2.1 Розробка додатків на Ionic: Переваги та недоліки	15
2.2.2 Різниця між нативними та крос-платформними додатками.....	17
2.3 Висновок до розділу 2	18
РОЗДІЛ 3. ПРОЦЕС РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ «KNUTE APP»	20
3.1 Інструменти для розробки додатку	20
3.2 Характеристика додатку	23
3.3 Структура додатку	24
3.4 Компонент додатку	25

					КНТЕУ 121-05-18.МР			
					«Розробка мобільного додатку «KNUTE App»»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Зміст	2	43
Зав. каф.	Криворучко О.В.					Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		
Керівник	Цюцюра С.В.							
Гарант	Криворучко О.В.							
Розробив	Проява В.В.				Зміст			

3.5	Елементи сторінки	31
3.6	Висновок до розділу 3	39
	ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	40
	СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ	42
	ДОДАТКИ	44
	ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ	44
	КЕРІВНИЦТВО КОРИСТУВАЧА	47

					КНТЕУ 121-05-18.МР			
					«Розробка мобільного додатку «KNUTE App»»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Зміст	3	43
Зав. каф.	Криворучко О.В.					Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		
Керівник	Цюцюра С.В.							
Гарант	Криворучко О.В.							
Розробив	Проява В.В.				Зміст			

ВСТУП

Ринок мобільних додатків стрімко розвивається, гаджети стають невід’ємною частиною людського життя. При цьому, частка звичайних мобільних телефонів зменшується, що свідчить про збільшення долі смартфонів. Вже зараз кожен третій українець користується розумним гаджетом. Технічні можливості смартфонів постійно розвиваються, що дає підґрунтя для розвитку ринку мобільних додатків і потреби в нових спеціалістах – розробників мобільних додатків, попит на яких перевищує пропозицію.

Ринок техніки сьогодні все більше і більше тяжіє до портативних пристроїв. Адже для сучасної людини важливо постійно мати доступ до Інтернету, електронної пошти, соціальних мереж. У цьому аспекті портативні пристрої значно перевершують стаціонарні ПК. Головні переваги – розмір, тривалість роботи без підзарядки і, звичайно ж, багатофункціональність.

Таким чином, варто визнати, що майбутнє ПК за портативною, легкою та функціональною технікою – планшетах, електронних книжках, нетбуках і смартфонах.

І продовжуючи дослідження роботи університету, було прийнято рішення продовжити розробку мобільного додатку, однак, тепер без обмеження однією кафедрою. Тепер, мобільний додаток буде призначений для користування всіх співробітників університету.

Відповідно, після переосмислення роботи, також змінилась і технологія розробки, із-за чого, додаток був повністю переписаний з нуля на основі

					<i>КНТЕУ 121-05-18.МР</i>				
					«Розробка мобільного додатку «KNUTE App»»	Стадія	Аркуш	Аркушів	
Зм.	Аркуш	№ докум.	Підпис	Дата		В	4	43	
Зав. каф.	Криворучко О.В.					Вступ	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		
Керівник	Цюцюра С.В.								
Гарант	Криворучко О.В.								
Розробив	Проява В.В.								

технології Ionic framework, що являється покращеною версією Cordova.

Apache Cordova - це платформа для розробки мобільних додатків з відкритим вихідним кодом. Вона дозволяє використовувати стандартні веб-технології, такі як HTML5, CSS3 і JavaScript для крос-платформної розробки, уникаючи рідної мови розробки для кожної з мобільних платформ. Програми запускаються всередині оболонки, націленої на кожну платформу, і використовують стандартні API для доступу до датчиків пристрою, даних і стану мережі.

Такиим чином, додаток більше не буде обмежений лише однією платформою Android, що є плюсом для тих співробітників, які користуються смартфонами з іншою операційною системою, такі як IOS, Windows Phone і тому подібні.

Окрім крос-платформності, розробка з допомогою Ionic має як переваги так і недоліки, про які піде мова в другому розділі роботи.

Актуальність полягає у дослідженні роботи київського національного торговельно-економічного університету, та розробки програмного забезпечення для мобільних пристроїв, яке б допомогло персоналу в роботі, зберігаючи всю необхідну інформацію в смартфоні.

Мета дослідження: Розробка додатку для співробітників КНЕТУ, що дозволить керувати інформацією за допомогою смартфона.

Об'єкт дослідження: Розробка додатку під різні операційні системи.

Предмет дослідження: створення гібридного мобільного додатку за допомогою фреймворку Ionic та Angular.

Задачі дослідження: Дослідити роботу співробітників університету; Створення додатку на основні крос-платформних технологій; Розробка мовою програмування TypeScript;

Наукова новизна дослідження:

1. Розробка мобільного додатку з використанням технологій для

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		5

web-розробки.

2. Розробка одного додатку відразу на декілька платформ.

Метод і технології розробки: В роботі використовуються методи об'єктно-орієнтованого програмування, аналіз операційних систем, програмування на мові TypeScript, робота з сховищами даних, розробка інтерфейсу користувача, робота з фреймворками.

Результат роботи: створення мобільного додатку для співробітників університету з інтуїтивно зрозумілим інтерфейсом.

В даній роботі описана розробка мобільного додатку “KNUTE App” з використанням фреймворку Ionic та технологій web-розробки, що дозволяє швидко розробляти додатки на декілька платформ, не потребує глибоких пізнань технологій розробки на декілька платформ.

					<i>KNUTE 121-05-18.MP</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		6

РОЗДІЛ 1.

ПЛАТФОРМА ТА ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ ДОДАТКУ «KNUTE APP»

1.1 Основні відомості про Android та програмне забезпечення

Android - це операційна система для смартфонів, планшетів, електронних книг, цифрових плеєрів, годинників, ігрових консолей, нетбуків, смартбордів, окулярів Google, телевізорів, домашніх роботів та інших пристроїв. Виходячи з ядра Linux та фактичної реалізації віртуальної машини Java від Google. Спочатку система була розроблена компанією Android, Inc., яка потім була придбана компанією Google. Згодом компанія Google ініціювала створення Alliance Handset Alliance (ОНА), який зараз займається підтримкою та подальшим розвитком платформи. Android дозволяє створювати додатки Java, які керують пристроєм через розроблені бібліотеки Google. Інтерактивний набір програм Android дозволяє переносити бібліотеки та компоненти додатків, написані мовою програмування C та іншими мовами високого рівня. У 86% смартфонів, проданих у другому кварталі 2014 року, була встановлена операційна система Android. На конференції розробників у травні 2017 року Google оголосила, що в історії Android активовано більше 2 мільярдів Android-пристроїв.

Android, Inc. була заснована в Пало-Альто, Каліфорнія, у жовтні 2003 року Енді Рубіном (Енді Рубін, співзасновник Danger Company), Річардом Мінером (співзасновник Wildfire Communications, Inc.), колишнім віце-

					КНТЕУ 121-05-18.МР				
					«Розробка мобільного додатку «KNUTE App»»	Стадія	Аркуш	Аркушів	
Зм.	Аркуш	№ докум.	Підпис	Дата		P1	7	43	
Зав. каф.	Криворучко О.В.					Розділ 1. Платформа та інструменти для розробки додаток «KNUTE App»	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		
Керівник	Цюцюра С.В.								
Гарант	Криворучко О.В.								
Розроб	Проява В.В.								

президентом Сірс компанії T-Mobile, а Кріс Біл - очолював розробку та розробку інтерфейсу на WebTV), щоб, за словами Рубіна, розвинути "більш інтелектуальні мобільні пристрої, які краще знають про місце проживання власника та його переваги". Ранні наміри компанії полягали у розробці передової операційної системи для цифрових фотоапаратів, однак було ясно, що ринок пристроїв був недостатнім, і вони зосередили свої зусилля на розробці операційної системи для смартфонів, щоб конкурувати з Symbian і Windows Mobile (на той момент Apple ще не випустив iPhone). Незважаючи на минулі досягнення засновників та ранніх співробітників, Android, Inc. працював таємно, показуючи лише те, що вона працювала на програмне забезпечення для мобільних телефонів. Детальніше про розвиток операційної системи Android описано в таблиці 1.1.

Таблиця 1.1

Дата	Нововведення
Липень 2005 р	Придбання Google компанії Android, Inc.
5 листопада 2007 року	Створення Open Handset Alliance (ОНА) та анонс відкритої мобільної платформи Android
12 листопада 2007 року	Альянс показав першу версію пакету для розробників Android «Early Look» SDK і емулятор Android.
23 вересня 2008	Офіційно вийшла перша версія ОС, а також перший повноцінний пакет розробника SDK 1.0, Release 1.
2009 рік	Було показано чотири оновлення платформи.
2010 рік	Google представила версію Android 2.2 під назвою «Froyo», а в кінці 2010 року - Android 2.3 «Gingerbread».
22 лютого 2011	Була офіційно презентована орієнтована на інтернет-планшети платформа - Android 3.0 «Honeycomb».
19 жовтня 2011	Реліз Android 4.0 «Ice Cream Sandwich»
Червень 2012	Реліз оновлення під назвою «Jelly Bean» з номером 4.1 який змінився на 4.2 внаслідок невеликого оновлення в кінці жовтня того року.

					КНТЕУ 121-05-18.МР	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		8

Продовження таблиці 1.1

Липень 2013	Реліз версії 4.3
31 жовтня 2013	Google показали наступну версію операційної системи Android 4.4, яка отримала назву «KitKat». Ця версія Android оптимізована для роботи на більш широкому наборі пристроїв, що мають 512 МБ ОЗУ в якості рекомендованого мінімуму.
25 червня 2014	Google представили Android L, доступний для розробників, користувачів смартфонів Nexus, а також деяких інших смартфонів. Слідом була анонсована операційна система для пристроїв Android Wear. Також на Google I/O були представлені версії Android Auto (для автомобілів) і Android TV (для телевізорів), тим самим Android перестав бути операційною системою тільки для мобільних пристроїв.
15 жовтня 2014	Було офіційно анонсована Android 5.0 Lollipop. Головне оновлення системи - новий дизайн Material Design. Також, була додана нова функція. Якщо Android-пристрій встановлений пароль або графічний ключ, і якщо поблизу знаходяться годинник господаря пристрою з Android Wear, то пристрій автоматично розблокується.
9 грудня 2014	Google змінила середовище розробки, засновану на Eclipse (adt-bundle), на Android Studio .
29 травня 2015	Google представила Android M. За словами Google головна мета нової операційної системи - поліпшити користувацький досвід спілкування зі смартфоном, зробити взаємодію інтуїтивно і простіше.
17 серпня 2015	Офіційно стало відомо, що ця версія отримала назву Android 6.0 Marshmallow.
19 травня 2016	На конференції розробників Google I/O була представлена нова версія Android N, а 1 липня Google офіційно оголосила назву цієї версії Android - Android 7.0 Nougat. Розсилка фінальної версії операційної системи стартувала 22 серпня 2016 року.
Березень 2017	Був представлений Android O для розробників. Призначена для користувача версія була випущена 21 серпня 2017 роки як 8.0.0 Oreo.

					КНТЕУ 121-05-18.МР	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		9

1.2 Структура операційної системи Android

Після встановлення на пристрій, кожна програма знаходиться в особистому, безпечному, середовищі ПЗ:

1. ОС Android це багатокористувацька система на основі Linux, в якій, кожна програма це інший користувач.
2. За замовчуванням, система призначає кожній ID (унікальний ідентифікатор) для користувача Linux (ідентифікатор використовується лише системою, та видимий для програми). Система видає дозволи для всіх файлів, таким чином, що доступ до них, може надаватися лише даному ідентифікатору, наданому додатком.
3. Кожен процес має власну віртуальну машину, що ізольована від інших програм. Як правило, кожна програма запускається у процесі Linux.
4. Android запускає процес, коли необхідно запустити будь-який компонент програми, необхідний для запуску. Потім він знищує процес, якщо він більше не потрібний, або коли система потребує відновлення пам'яті для інших програм. Таким чином, система Android реалізує принципи найменшої привілеї. Тобто кожна програма, за умовчанням, має доступ лише до тих компонентів, які вона потребує, і не більше. Це створює безпечне середовище, в якому програма не може отримати доступ до тієї частини системи, на яку не має дозволу.

Проте існують способи обміну даних з іншими програмами та доступу до системних служб:

1. Ви можете розширити ідентифікатор Linux між двома додатками, щоб вони могли отримати доступ до файлів один одного. Щоб зберегти системні ресурси, програми з тим самим ідентифікатором

					КНТЕУ 121-05-18.МР	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		10

можуть бути запущені в одному процесі та поділяють одну і ту ж віртуальну машину (Virtual Machine, VM).

2. Додаток може запитувати дозвіл на доступ до персональних даних пристрою, наприклад, контактів користувача, SMS-повідомлень, розділу додаткової пам'яті (SD-карта), камери, Bluetooth і багато іншого. Всі дозволи до програми повинні надаватися користувачем під час встановлення.

1.3 Особливості розробки крос-платформних додатків

Завдяки стрімко поширенню мобільного Інтернету і розробці ІТ-додатків, встановлених на нових, інтуїтивно зрозумілих для користувача пристроях, мобільний додаток стає хорошим тоном, візиткою, комерційним інструментом і додатковим джерелом доходу. Тому розробка нового покоління мобільних додатків, що використовують web-ресурси і сучасні ІТ-технології, стає популярною і потенційно вигідною.

На сьогоднішній день найпопулярнішими і сучасними рішеннями для крос-платформенної розробки мобільних додатків є: PhoneGap, Xamarin, Unity. Дані фреймворки на сьогоднішній день займають 80% ринку крос-платформенної розробки для мобільних пристроїв.

Фреймворк PhoneGap дозволяє створювати мобільні додатки Використовуючи стандартні вебтехнологов (HTML5, JavaScript і CSS3). В результаті це призвело до швидкого зростання популярності фреймворка, з його допомогою можна обійтися без розробки на таких мовах програмування як: Java для Android, Objective-C для iOS і C #.

Основні переваги PhoneGap:

1. просте API, що дозволяє швидко почати розробку;
2. можливість використання будь-яких існуючих JavaScript бібліотек (jQuery, Prototype, Sencha Touch).
3. Велика кількість готових плагінів

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		11

4. У даного фреймворка є і недоліки, такі як:
5. візуалізація призначеного для користувача інтерфейс за допомогою вбудованого браузера. Це створює труднощі в отриманні зворотного зв'язку по порівнянні з нативним додатком;
6. необхідність написання своїх плагінів, так як існуючі плагіни в більшості випадків виявляються застарілими.

Наступний Розглянутий фреймворк для крос-платформенной розробки - Xamarin. Xamarin дозволяє створювати одну єдину логіку додатка із застосуванням C # і .NET. Функціонально платформа Xamarin представляє ряд субплатформ. Ці субплатформи відіграють велику роль - через них додатки можуть направляти запити до прикладних інтерфейсів на пристроях. Також в Xamarin визначається візуальний інтерфейс, прив'язується логіка на C #, і все це працює на Android, iOS і Windows Phone.

Переваги Xamarin:

1. розвивається співтовариство;
2. TestCloud, що використовується розробниками, використовується для тестування додатків автоматично;
3. багаторазове використання вже написаного коду;
4. динамічна верстка для iOS;
5. можливість доповнення довільними властивостями стандартних контролів по рахунок Custom Renderers.

Недоліки Xamarin:

1. складність реалізації деяких інтерфейсних патернів на monodroid і monotouch, так як рішення за замовчуванням для тієї чи іншої функції спираються на непередумані до кінця можливості платформи, які можуть просто не працювати в Xamarin;
2. виникнення проблем з боку платформи mono, monotouch і

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		12

monodroid, так як додаток повинен задовольняти особливим вимогам стабільності; - неможливість розташування Android сторінок як частини вже існуючого Activity / Fragment;

3. немає можливості реалізації Деяких контролів (RadioButton, CheckBox, Hyperlink).

Останній Розглянутий інструмент - Unity. Мультиплатформовий інструмент для розробки 2D- і 3D-додатків та ігор Unity, також один з кращих інструментів для демонстрації 3D-контента. Створені за допомогою Unity програми працюють під операційними системами Windows, OS X, Linux, Android, Apple iOS, Windows Phone, BlackBerry, а також на ігрових приставках Wii, PlayStation 3 і Xbox 360.

Переваги:

1. наявність безкоштовних плагінів;
2. можливість створення власних шейдерів і зміни шляху, яким Unity здійснює візуалізації.

Недоліки:

1. складність у використанні для початківців розробників;
2. відсутність оптимізації компіляторів для ARM процесорів на деяких мобільних пристроях.

Для додатка «KNUTE App» був обраний фреймворк PhoneGap з надбудовою Ionic, який дозволяє використовувати HTML5, CSS3 та TypeScript замість JavaScript, який має більш розширений функціонал, та є логічним продовженням JS.

Проте, слід зазначити, що немає ідеального рішення, кожен фреймворк має свої плюси і мінуси. Але завдяки Ionic, є можливість розробляти один додаток на декілька платформ одразу, та не витратити час на оптимізацію додатку під його роботи на різних мобільних операційних системах (Android, iOS і Windows Phone).

					<i>KNUTE 121-05-18.MP</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		13

1.4 Висновок до розділу 1

Android – як основна платформа для розробки та експлуатації мобільного додатку, добре підходить для використання, так як є відкритою системою для розробників і дозволяє встановлювати додатки з інших джерел, окрім офіційного магазину, як це зроблено на IOS пристроях. Це дозволяє встановлювати ранні версії додатку та тестувати, чи коректно вони працюють на даному пристрої та на системі андроїд в цілому.

Проте, незважаючи на переваги Android систем, на сьогоднішній день є також багато користувачів операційної системи IOS, під яку теж розроблять додатки. Тому, щоб покрити як можна більшу кількість співробітників, без великої кількості затрат, було обрано метод розробки додатку на крос-платформній основі.

В якості інструменту був обраний фреймворк Ionic на основі PhoneGap, який дозволяє створювати мобільні додатки на основі web-технологій. А саме HTML, SCSS та TypeScript. Цей набір технологій дає змогу розробляти додатки швидко, має надзвичайно велику кількість документації в свободному доступі, а головною перевагою являється змогу дивитися на зміни в додатку відразу в вікні браузера, без необхідності кожен раз робити збірку проекту, та встановлювати її на пристрій. Що робить процес надзвичайно швидким.

І хоча на даний час, у PhoneGap є гідні конкуренти, які дозволяють розробляти додатки на декілька платформ, саме цей фреймворк має найменшу кількість мінусів і постійно підтримується командою розробників та ентузіастів, які також приймають участь у розробці фреймворку.

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		14

РОЗДІЛ 2.

ТЕХНОЛОГІЇ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

2.1 Функція додатку

Додаток «KNUTE App» розроблений для співробітників університету як допоміжний засіб в роботі. З його допомогою, працівники можуть мати швидкий доступ до довідника контактів всіх співробітників, і з його допомогою, швидко зв'язатися з потрібною людиною.

Є можливість додавати окремі контакти в список обраних, щоб мати до них швидкий доступ, коли, наприклад, необхідно часто контактувати з ними. І при необхідності, можна створити новий контакт.

2.2 Технології розробки додатку

Для створення даного додатку, були використані нові технології розробки додатків для мобільних пристроїв. Основою додатку є фреймворк Ionic, який використовує набір CSS і HTML компоненти, створений на основі Apache Cordova використовуючи мову програмування TypeScript.

Apache Cordova дозволяє створювати універсальні мобільні додатки, що працюють на різних мобільних платформах, з використанням стандартних веб-технологій (HTML5, CSS3 і JavaScript). Використання Apache Cordova дозволяє створювати застосунки, що функціонують на широкому спектрі мобільних платформ, включаючи Tizen, webOS, Android, Apple iOS, Blackberry, Samsung Bada і Windows Phone.

2.2.1 Розробка додатків на Ionic: Переваги та недоліки

Ionic - це технологія, що дозволяє розробляти повноцінні програми

					KHTEU 121-05-18.MP			
					«Розробка мобільного додатку «KNUTE App»»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		P2	15	43
Зав. каф.		Криворучко О.В.						
Керівник		Цюцюра С.В.						
Гарант		Криворучко О.В.						
Розробив		Проява В.В.			Розділ 2. «Додаток «KNUTE App»»	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		

для iOS і Android. Для цього не потрібно мати глибокі знання в кожній з платформ. Звичайно ж, є деякі обмеження, але в цілому необхідно бути знайомим з Angular (популярний веб-фреймворк), щоб почати розробку програми. Для застосування стилів можна використовувати SCSS - це додасть додатку потрібний вид.

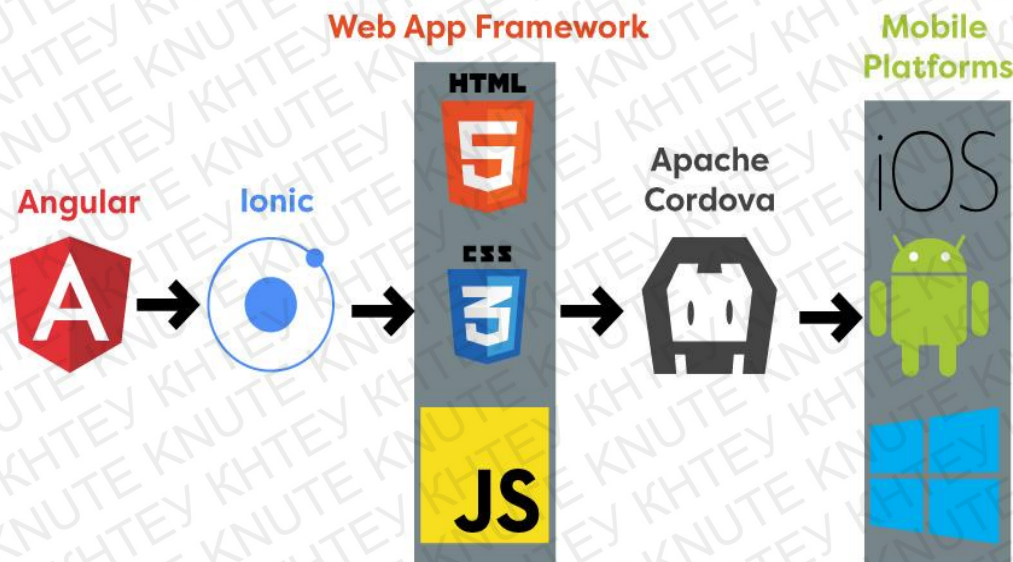


Рис.2.1 Структура розробки додатку на Ionic

У Ionic є вбудована бібліотека стандартних елементів, які можна використовувати аналогічно елементам Bootstrap: картки, кнопки, перемикачі, сегменти, попап, поля введення, списки, сітка з рядків і колонок і т.д. За замовчуванням ці елементи змінюються так, щоб виглядати як нативні на iOS і Android, але їх вид можна змінити і при необхідності.

Також з Ionic доступно безліч плагінів, які дозволяють використовувати залізо смартфона (Ionic Native / Cordova). Але необхідно стежити, щоб платформи активно підтримували обрані плагіни. У більшості випадків такі плагіни працюють добре, але іноді може виникнути помилка або конфлікт навіть з широко використовуваними плагінами по типу входу через Facebook або Firebase Analytics. Такі

проблеми зазвичай вирішуються або чищенням і ребілдом проекту, або оновленням плагіну.

У Ionic є ще один великий плюс - швидкість розробки. Оскільки він оснований на Angular, проект на Ionic можна запускати в браузері і бачити, як буде виглядати додаток під час розробки. Для того щоб побачити таке прев'ю, не обов'язково встановлювати додаток на смартфон або емулятор. Це істотно допомагає економити час при зміні UI. А при роботі над функціями, які вимагають перевірки на смартфоні (наприклад, зробити фото), збірка білду і установка його на смартфон займе всього кілька хвилин. На Android встановлювати додатки можна прямо з командного рядка, а на iOS білд потрібно відкрити в Xcode.

2.2.2 Різниця між нативними та крос-платформними додатками

Нещодавно з'явилося кілька фреймворків, що дозволяють розробляти справжні нативні додатки з використанням Angular (NativeScript) або React (React Native). Вони мають важливу перевагу перед гібридним підходом Ionic - ніщо не може зрівнятися з продуктивністю нативних додатків. Однак нативні додатки мають не менш важливий мінус. Використовуючи Ionic, ви працюєте з HTML- і SCSS-файлами, а для нативної розробки знадобляться інші навички для роботи з розміткою і стилями.

Переваги при розробці крос-платформних додатків

1. Швидка розробка і мінімальний час виходу на ринок.
2. Можна вести основну частину розробки в браузері (крім нативної функціональності смартфона - тут знадобиться використовувати смартфон для дебага).
3. Можна розробляти додаток для iOS і Android одночасно (з деякими обмеженнями - такими як особливості платформ, що стосуються стилів і плагінів).

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		17

4. Навички в Angular, HTML, CSS і JavaScript - це практично все, що знадобиться для початку розробки. Немає необхідності знати Java і Swift або Objective-C.
5. Безліч UI-компонентів доступні і прості у використанні - картки, кнопки, перемикачі, сегменти, попап, поля введення, списки, сітка з рядків і колонок і т. Д.
6. Безліч плагінів, що дозволяють використовувати функції смартфонів, такі як: камера, сканер відбитків пальців, NFC, геолокація, відправка аналітики на Firebase, оповіщення та глибинні посилання.

Недоліки при розробці крос-платформних додатків

1. Нативні плагіни можуть стати проблемою, якщо якісь із використовуваних плагінів конфліктують або якщо в одному з них баг. Наприклад, недавно в плагіні для логіна за допомогою Facebook був баг: якщо юзер хоч раз зробив log out, то більше логін вже не працював. А плагін FCM (Firebase Cloud Messaging) не працює з firebase-analytics. Але є плагін під назвою Firebase, яким можна замінити їх обидва. Дебаг може бути досить складним: іноді важко зрозуміти, звідки приходить помилка, оскільки повідомлення про помилки можуть бути неінформативними.
2. Білд може ламатися без причин, щось виявляється пошкодженим в оригінальній папці.

В якості середовища для розробки використовувалось IDE (Integrated Development Environment) Visual Studio Code, що використовується для розробки web-продуктів (сайти, портали, форуми та ін.).

2.3 Висновок до розділу 2

На даний час, Ionic є одним із найпоширеніших фреймворків для розробки мобільних додатків на декілька платформ. Завдяки його основі, яка складається з web-інструментів, таких як HTML, CSS, TS, він є легкий

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		18

в освоєні, і дозволяє швидко розробляти прототипи або готові додатки, спостерігаючи за процесом прямо в вікні браузера. Це дуже зручно, особливо коли потрібно внести незначні правки в проект по дизайну, і не потрібно витрачати час на встановлення збірки.

І хоча Ionic має свої мінуси, які можуть відлякнати потенційних розробників, все ж таки його плюси та переваги нівелюють ці мінуси. Так як головними перевагами є низький поріг входу та готова бібліотека UI компонентів, що дозволяє за доволі короткий час робити красиві додатки.

Проте, із мінусів потрібно відмітити деякі важливі аспекти. Плагіни для фреймворку не завжди можуть оновлюватися відповідно до останньої версії самого фреймворку, і тому бувають ситуації, коли уже зроблений проект, після внесення змін, може перестати працювати. Але це вирішується доволі швидко та просто, перезбіркою проекту, або встановленням попередньої версії фреймворку.

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		19

РОЗДІЛ 3.

ПРОЦЕС РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ «KNUTE APP»

3.1 Інструменти для розробки додатку

Для розробки додатку на Ionic, використовуються такі інструменти як: HTML5, SCSS3 та TypeScript. Зазвичай, ці мови використовують для створення Web-додатків, але фреймворк Ionic дозволяє використовувати гнучкість та простоту HTML та SCSS в розробці мобільних додатків. Це значно спрощує розробку дизайну додатку, та дозволяє простіше створювати майже будь-який зовнішній вигляд додатку порівняно з нативними мовами платформ.

Hypertext Markup Language (HTML) (Мова розмітки гіпертекстових документів) — стандартна мова розмітки для створення веб-сторінок і веб-додатків. З Cascading Style Sheets (CSS) і JavaScript, вона утворює тріаду основних технологій для World Wide Web.

Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци,

					<i>КНТЕУ 121-05-18.МР</i>			
					«Розробка мобільного додатку «KNUTE App»»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		РЗ	20	43
Зав. каф.		Криворучко О.В.			Розділ 3. Процес розробки мобільного додатку «KNUTE App»»	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		
Керівник		Цюцюра С.В.						
Гарант		Криворучко О.В.						
Розроб		Проява В.В.						

списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегами, написаними з використанням кутових дужок. Теги, такі як і безпосередньо вводять вміст на сторінку. Інші теги, такі як `<img />` `<input />` `<p>` оточують і надають інформацію про текст документа і можуть включати інші теги як під-елементи. Браузери не показують теги HTML, але використовують їх для інтерпретації вмісту сторінки.

HTML може вбудовувати програми, написані на мові сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і компоновання вмісту. World Wide Web Consortium (W3C), які супроводжують як HTML і CSS стандартів, заохочує використання CSS над явним презентаційним HTML з 1997 року.

HTML впроваджує засоби для:

1. створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
2. отримання інформації із Всесвітньої мережі через гіперпосилання;
3. створення інтерактивних форм;
4. включення зображень, звуку, відео, та інших об'єктів до тексту.

Cascading Style Sheets (Каскадні таблиці стилів) — спеціальна розбитка, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		21

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

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

TypeScript — мова програмування, представлена Microsoft восени 2012; позиціонується як засіб розробки веб-застосунків, що розширює можливості JavaScript.

Розробником мови TypeScript є Андерс Гейлсберг (англ. Anders Hejlsberg), який створив раніше C#, Turbo Pascal і Delphi.

Код експериментального компілятора, котрий транслює код TypeScript в представлення JavaScript, поширюється під ліцензією Apache, розробка ведеться в публічному репозиторії через сервіс CodePlex. Специфікації мови відкриті і опубліковані в рамках угоди Open Web Foundation Specification Agreement (OWFa 1.0).

TypeScript є зворотно сумісним з JavaScript. Фактично, після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або використовувати спільно з серверною платформою Node.js.

Переваги над JavaScript:

1. можливість явного визначення типів (статична типізація),
2. підтримка використання повноцінних класів (як в традиційних об'єктно-орієнтованих мовах),
3. підтримка підключення модулів.

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		22

За задумом ці нововведення мають підвищити швидкість розробки, прочитність, рефакторинг і повторне використання коду, здійснювати пошук помилок на етапі розробки та компіляції, а також швидкодію програм.

Планується, що в силу повної зворотної сумісності адаптація існуючих застосунків на нову мову програмування може відбуватися поетапно, шляхом поступового визначення типів. Підтримка динамічної типізації зберігається — компілятор TypeScript успішно обробить і не модифікований код на JavaScript. Основний принцип мови — весь існуючий код на JavaScript сумісний з TypeScript, тобто в програмах на TypeScript можна використовувати стандартні JavaScript-бібліотеки і раніше створені напрацювання. Більш того, можна залишити існуючі JavaScript-проекти в незмінному вигляді, а дані про типізації розмістити у вигляді анотацій, які можна помістити в окремі файли, які не заважатимуть розробці і прямому використанню проекту (наприклад, подібний підхід зручний при розробці JavaScript-бібліотек).

На момент релізу представлені файли для сприйняття розширеного синтаксису TypeScript для Vim і Emacs, а також плагін для Microsoft Visual Studio.

Одночасно з виходом специфікації розробники підготували файли з деклараціями статичних типів для деяких популярних JavaScript-бібліотек, серед яких jQuery.

3.2 Характеристика додатку

Для розробки додатку «KNUTE App» були використані web-технології та фреймворк Ionic на основі ядра PhoneGap. Ця технологія дозволяє створювати гібридні додатки одразу для всіх операційних систем, оскільки вимоги до web на всіх платформах одні. Цей метод значно зменшує витрати на розробку,

					КНТЕУ 121-05-18.МР	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		23

оскільки не потребує створення додатку конкретно під одну операційну систему.

Звичайно, цей метод не без недоліків. Основний недолік даного підходу – швидкість роботи, порівняно з нативними додатками конкретно під платформу. Проте, для даного додатку, швидкість роботи не є критичною, та не потребує складних обчислень та функціоналу, який краще реалізовано саме в нативних додатках.

TypeScript виникла з недоліків JavaScript для розробки масштабних додатків як у Microsoft, так і серед їх зовнішніх клієнтів. Проблеми, що стосуються роботи зі складним кодом JavaScript, спричинили потребу в користувальницьких інструментах для полегшення розробки компонентів мови.

Розробники TypeScript шукали рішення, яке не порушило б сумісність зі стандартом та його міжплатформною підтримкою. Знаючи, що поточна стандартна пропозиція ECMAScript обіцяє майбутню підтримку програмування на основі класів, TypeScript базувався на цій пропозиції. Це призвело до компілятора JavaScript з набором синтаксичних розширень мови, суперсети на основі пропозиції, що перетворює розширення в звичайний JavaScript. У цьому сенсі TypeScript був попереднім переглядом того, чого можна очікувати від ECMAScript 2015. Унікальним аспектом, який не в пропозиції, а доданий до TypeScript, є обов'язковий статичний набір тексту, що дозволяє статичний аналіз мови, що полегшує інструментарій та підтримку IDE.

3.3 Структура додатку

Кожний компонент (сторінка) складається з трьох файлів різного змісту.

1. HTML – тип файлу, в якому прописується структура сторінки, які елементи там присутні, де і як дані будуть відображені

					КНТЕУ 121-05-18.МР	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		24

користувачеві.

2. SCSS – тип файлу, в якому розробник стилізує елементи. Кожному елементу (тегу) в файлі html присвоюється клас, який потім викликається в даному файлі. Так, можна задати певні розміри будь якому елементу, додати відступи, та розмістити так, як це виконано в макеті сторінки.
3. TS – тип файлу typescript, в якому прописується весь функціонал сторінки. Тут ініціалізуються функції, які відповідають за виконання певних дій. Наприклад, виведення товару на сторінці в змінних, які додані в файлі html, перехід на інші сторінки, додавання товару в корзину та все інше, що відноситься до функціональної частини.

3.4 Компонент додатку

Як було зазначено вище, мобільний додаток «KNUTE App» складається з компонентів, до якого входять три елементи – html, scss, ts. Всі ці елементи тісно переплітаються між собою, і без якогось з них, додаток будет виглядати гірше, а то й зовсім не працювати. Цю взаємодію я хотів би продемонструвати на прикладі сторінки Мої контакти, тим самим продемонструвати навички, які були отримані під час роботи над цим проектом.

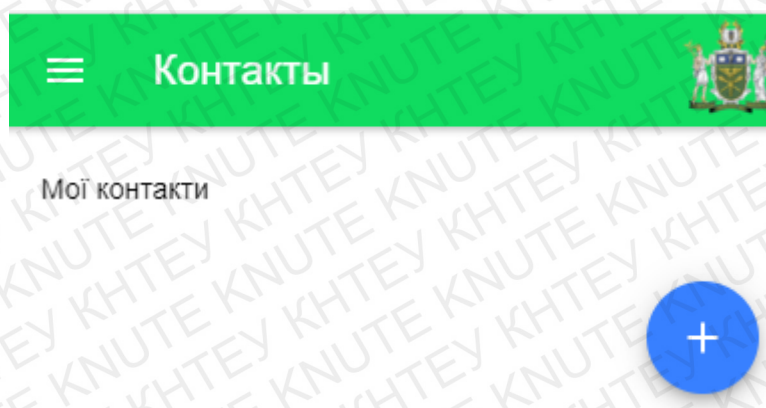


Рис.3.1. Звичайний вигляд сторінки «Мої контакти»

					KHTEY 121-05-18.MP	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		25

На зображенні показана пуста сторінка, на якій мають відображатися контакти, що додав користувач сам.

Для того, щоб відкрити форму для заповнення даних, потрібно натиснути на синю кнопку в правому кутку екрану. Після цього, відкриється форма для вводу інформації.

Рис.3.2. Відкрита форма для вводу інформації

Форма складається з наступних елементів:

- Імя

					<i>КНТЕУ 121-05-18.МР</i>		Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата			26

- Поле, в яке вводять імя користувача. Можна вводити як повне прізвище, імя та по-батькові, так і лише імя.
- Фото
 - Для того, щоб фото відображалось в додатку, потрібно скопіювати посилання із будь якого інтернет-ресурсу, де є фото користувача. Це може бути як офіційний сайт КНТЕУ, так і соціальні мережі.
- Посада
 - Поле, в якому зазначається посада контакту
- Телефон
 - Поле для введення номера телефона. Поле має особливі опції, тому після збереження запису, на телефон можна натиснути, і смартфон сам відкриє додаток для здійснення зв'язків, де вже буде додано номер телефону. Це дозволяє швидко здійснювати зв'язки потрібним користувачам.
- Пошта
 - Поле, де потрібно вводити електронну пошту користувача. Поле має особливі опції, тому після збереження адреси електронної пошти, користувач може натиснути на назву, після чого відкриється поштовий додаток в смартфоні, де в полі «Кому» уже буде додана адреса контакту із додатку.
- Кабінет
 - Поле для зазначення в якому кабінеті знаходиться робоче місце контакту
- День народження
 - Поле, в якому потрібно зазначити дату народження контакту. Поле має особливі опції, тому після збереження контакту, в дату коли у контакта буде день народження, мобільний

					КНТЕУ 121-05-18.МР		Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата			27

додаток надішле користувачу нагадування.

- Інформація
 - Поле, в якому користувач може зазначити будь яку важливу інформацію про співробітника, якого він додає до списку.

Коли всі поля заповнені, користувач повинен натиснути на кнопку «Додати запис», щоб додаток зберіг введену інформацію.

Так виглядають заповнені поля та сам процес додавання контакту:

The screenshot shows a mobile application interface with a green header bar containing a menu icon and the title 'Контакты'. Below the header, there is a section titled 'Мої контакти' (My contacts). A contact card is displayed with the following fields: 'Імя' (Name) with the value 'Харченко Олександр Анатолійович', 'Фото' (Photo) with a URL, 'Посада' (Position) with the value 'Декан ФОАІС', 'Телефон' (Phone) with the value '531-49-56', 'Пошта' (Email) with the value 'a.kharchenko@knute.edu.ua', 'Кабінет' (Cabinet) with the value 'Д-105', 'День народження' (Date of birth) with the value '01/10/2019', and 'Інформація' (Information) with the value 'Декан'. At the bottom of the card is a blue button labeled 'ДОДАТИ ЗАПИСІ' (Add record). A blue circular button with a white 'X' is located at the bottom right of the screen.

Рис.3.3. Заповнені поля з даними

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		28

Після натискання кнопки «Додати запис!», запис буде додано та збережено на сторінці «Мої контакти».

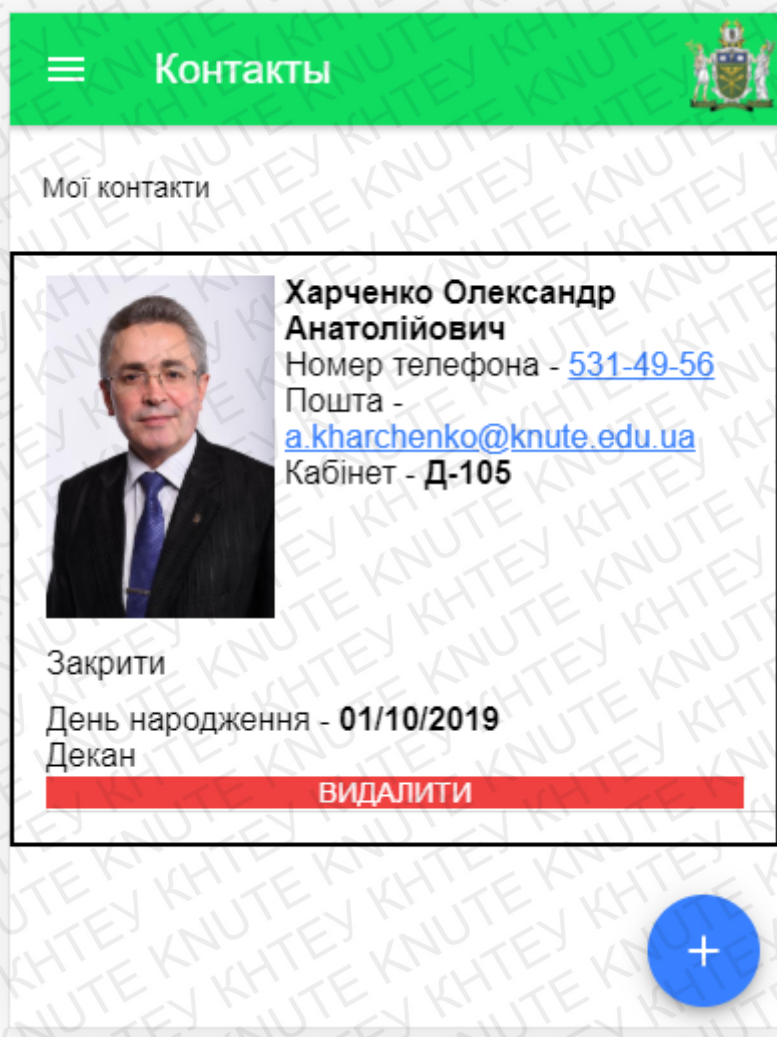


Рис.3.4 Збережений запис контакту

На зображенні вище показано приклад того, як виглядає доданий контакт. Він зберігається в пам'яті телефону, тому користувач може завжди мати доступ до інформації.

Також у контакта є кнопка «Видалити», яка дозволяє видалити непотрібний контакт.

Для того, щоб стилізувати ці елементи, і надати їм гарного виду, використовується бібліотека CSS стилів, яка дозволяє кастомізувати елементи. В даному випадку, блокам з текстом були присвоєні класи, які

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		29

потім викликаються в елементі .scss. В даному випадку, код має наступний ВИГЛЯД.

```
src > app > home > home.page.scss > ...
1  .content {
2    padding-top: 10px; //Відступ зверху 10 пікселів
3  }
4
5  .itema {
6    border: 2px solid; //Чорна рамка товщиною 2 пікселі
7    padding-bottom: 15px; //Відступ знизу 15 пікселів
8    margin-bottom: 15px; //Відступ між елементами 15 пікселів
9  }
10
11 .contentBlock {
12   width: 100%; //Ширина 100% відносно блоку, в якому знаходиться елемент
13 }
14
15 .imageBlock {
16   box-sizing: border-box; //Відображення зображення на ширину всього блоку
17   display: table-cell; //Відображення зображення на ширину всього блоку
18   vertical-align: top; //Вирівнювання по вертикалі
19   padding: 10px 0; //Відступ з усіх сторін 10 пікселів
20   width: 30%;
21 }
22
23 .textBlock {
24   box-sizing: border-box; //Відображення зображення на ширину всього блоку
25   display: table-cell; //Відображення зображення на ширину всього блоку
26   vertical-align: top; //Вирівнювання по вертикалі
27   padding: 0 5px; //Відступ з усіх сторін 5 пікселів
28   width: 50%; //Ширина відносно батьківського блоку 50%
29   padding-top: 10px; //Відступ зверху 10 пікселів
30 }
31
32 .img {
33   max-width: 100%; //Максимальна ширина на блок 100%
34   border: 0; //Рамки 0 пікселів
35   width: 100%; //Ширина відносно батьківського блоку 100%
36 }
37
38 .moreTxt {
39   height: 100%; //Висота - 100%
40   padding-top: 10px; //Відступ зверху 10 пікселів
41 }
42
43 .addItem {
44   height: 100%; //Висота - 100%
45   padding-top: 37px; //Відступ зверху 37 пікселів
46   border: 2px solid; //Чорна рамка товщиною 2 пікселі
47 }
48
```

Рис.3.5. Стили для відображення контакту

Завдяки стилям, елементи виглядають красиво, на відміну від звичайного тексту. Без стилів, ці елементи мали б наступний вигляд.

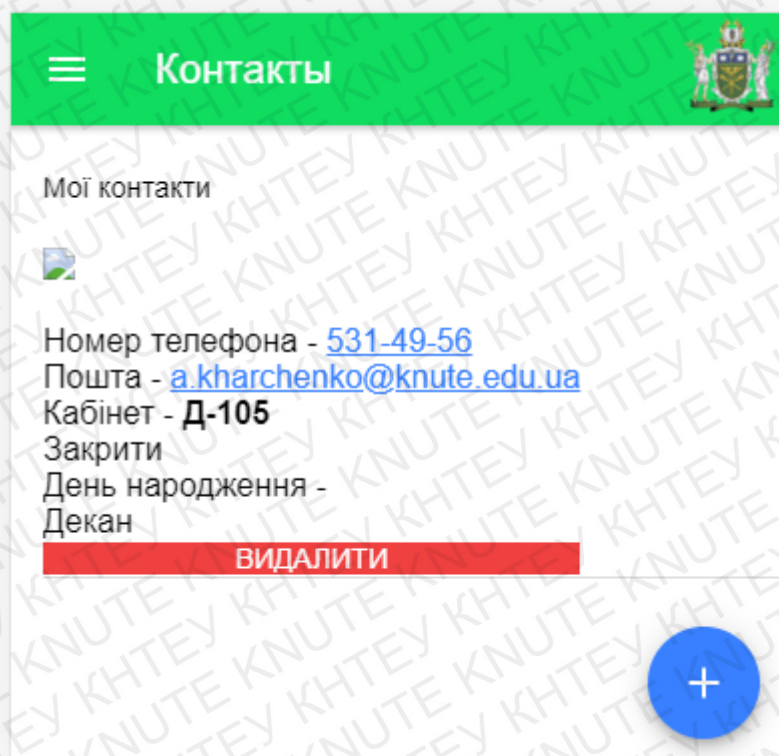


Рис.3.6. Сторінка без стилів

Одразу видно, що елементи сприймаються не так, як на те розраховує розробник або дизайнер. Нічого не центровано, елементи не відрізняються один від одного, і зображення не завантажуються зовсім. Тому, дуже корисно використовувати стилі, щоб виділяти потрібні елементи в додатку.

3.5 Елементи сторінки

Сторінка «Мої контакти» складається з трьох файлів формату .html, .scss, .ts. В кожному з них описані компоненти, які пов'язані між собою, і дозволяють сторінці функціонувати так, як це було продумано. В кожному з файлів знаходиться свій тип коду.

В файлі типу html, знаходиться розмітка сторінки. Там описано те, як блоки мають відображатися на сторінці, що за чим іде, та яка інформація

будет відображатися в тих чи інших елементах.

```
src > app > home > <> home.page.html > <> ion-header
1 <ion-header>
2   <ion-toolbar color="success">
3     <ion-buttons slot="start">
4       <ion-menu-button></ion-menu-button>
5     </ion-buttons>
6     <a slot="end" href="#" onclick="window.open('https://www.knteu.kiev.ua/', '_system'); return false;">
7       
8     </a>
9   <ion-title>
10     Контакти
11   </ion-title>
12 </ion-toolbar>
13 </ion-header>
14
15 <ion-content>
16
17   <ion-list #mylist>
18
19     <ion-list-header>
20       <ion-label>Мої контакти</ion-label>
21     </ion-list-header>
22
23     <ion-item-sliding *ngFor="let item of items" class="category-block">
24       <ion-list class="content">
25         <ion-item class="itema">
26           <div class="contentBlock">
27             <div class="imageBlock">
28               
29             </div>
30             <div class="textBlock">
31               <b>{{item.name}}</b><br>
32               Номер телефона - <a href="tel:{{item.phone}}">{{item.phone}}</a><br>
33               Пошта - <a href="mailto:{{item.mail}}">{{item.mail}}</a><br>
34               Кабінет - <b>{{item.place}}</b><br>
35             </div>
36             <div class="moreBtn" (click)="item.expanded = !item.expanded">
37               <div *ngIf="!item.expanded">Детальніше</div>
38               <div *ngIf="item.expanded">Закрити</div>
39             </div>
40             <div class="moreTxt" *ngIf="item.expanded">
41               <div>
42                 День народження - <b>{{item.birthday | date: "dd/MM/yyyy"}}</b><br>
43               </div>
44               <div>
45                 {{item.info}}<br>
46               </div>
47             </div>
48             <ion-item-option color="danger" (click)="doConfirm(item)">Видалити</ion-item-option>
49           </div>
50         </ion-item>
51       </ion-list>
52     </ion-item-sliding>
53   </ion-list>
54 </ion-content>
55
56 </ion-list>
```

Рис. 3.7. Частина html файлу сторінки «Мої контакти»

На рисунку зображено блок хедеру, з 1 по 13 рядки, в якому знаходиться логотип університету, з посиланням на сайт вищого навчального закладу, а також назва самої сторінки.

В блоці нижче, знаходяться блоки з змінними, які будуть відображати

					KHTEU 121-05-18.MP	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		32

інформацію контактів, які додав користувач самостійно. Складаються такі блоки з наступних компонентів:

- Html тег – в данному прикладі `<div>`. Він являється контейнером, або ж блоком, для первих даних, так як в Html вся структура сторінок будується на блоках.
- Змінна з інформацією, де зберігаються данні. В данному прикладі – `item.photo`.
- І при необхідності SCSS клас, який дозволяє використовувати стилі до змінних.
- Функція `item.expanded` дозволяє створити відкриваючий список по натисканню на відповідну кнопку. Це дозволяє економити простір на екрані, та не навантажувати його непотрібною інформацією.
- Рядок `«(click)="doConfirm(item)»` означає функцію, яка спрацює по кліку, тобто по натиску на певну кнопку. Сам сценарій функції описаний в іншому файлі формату `ts`.
- Всі інші теги на сторіці являється допоміжними елементами, що допомагають зробити структуру сторінки більш зручною в відображенні та в використанні.

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		33

```

src > app > home > home.page.html > ion-header
54 </ion-item-sliding>
55
56 </ion-list>
57
58 <ion-fab vertical="bottom" horizontal="end" slot="fixed">
59   <ion-fab-button (click)="expanded = !expanded">
60     <ion-icon *ngIf="!expanded" name="add"></ion-icon>
61     <ion-icon *ngIf="expanded" name="close"></ion-icon>
62   </ion-fab-button>
63 </ion-fab>
64
65 <div class="addItem" *ngIf="expanded">
66
67   <ion-item>
68     <ion-label position="floating">Ім'я</ion-label>
69     <ion-input [(ngModel)]="name" [(ngModel)]="newItem.name"></ion-input>
70   </ion-item>
71
72   <ion-item>
73     <ion-label position="floating">Фото</ion-label>
74     <ion-input [(ngModel)]="newItem.photo"></ion-input>
75   </ion-item>
76
77   <ion-item>
78     <ion-label position="floating">Посада</ion-label>
79     <ion-input [(ngModel)]="newItem.position"></ion-input>
80   </ion-item>
81
82   <ion-item>
83     <ion-label position="floating">Телефон</ion-label>
84     <ion-input [(ngModel)]="newItem.phone"></ion-input>
85   </ion-item>
86
87   <ion-item>
88     <ion-label position="floating">Пошта</ion-label>
89     <ion-input [(ngModel)]="newItem.mail"></ion-input>
90   </ion-item>
91
92   <ion-item>
93     <ion-label position="floating">Кабінет</ion-label>
94     <ion-input [(ngModel)]="newItem.place"></ion-input>
95   </ion-item>
96
97   <ion-item>
98     <ion-label position="floating">День народження</ion-label>
99     <ion-datetime display-format="DD/MM/YYYY" [(ngModel)]="birthday" [(ngModel)]="newItem.birthday"></ion-datetime>
100   </ion-item>
101
102   <ion-item>
103     <ion-label position="floating">Інформація</ion-label>
104     <ion-textarea [(ngModel)]="newItem.info"></ion-textarea>
105   </ion-item>
106
107   <ion-button expand="full" (click)="addItem()">Додати запис!</ion-button>
108
109 </div>
110
111 </ion-content>

```

Рис. 3.8. Інші елементи файлу home.page.html «Мої контакти»

В даній частині файлу html знаходиться форма для заповнення даних. Для цього використовується новий масив, в який записуються вже нові елементи newItem.item які вводить користувач самостійно. І після натискання кнопки (click)="addItem()", дані перезаписуються в масив item.item, який відображається вже на головній сторінці.

					KHTEY 121-05-18.MP	Аркуш
Зм.	Аркуш	№ документу	Підпис	Дата		34

Структура сторінок з типом файлу TS значно відрізняється, адже там уже використовується повноцінна мова програмування зі своєю логікою та синтаксисом. В файлі `home.page.ts` реалізована функція виведення збережених контактів на екран додатку, та додавання нових контактів, які користувач вводить самостійно. .

```

1  import { Component, ViewChild } from '@angular/core';
2  import { StorageService, Item } from '../services/storage.service';
3  import { Platform, ToastController, IonList, AlertController } from '@ionic/angular';
4  import { LocalNotifications } from '@ionic-native/local-notifications/ngx';
5
6
7  @Component({
8    selector: 'app-home',
9    templateUrl: 'home.page.html',
10   styleUrls: ['home.page.scss'],
11 })
12 export class HomePage {
13
14   items: Item[] = [];
15
16   newItem: Item = <Item>{};
17
18   birthday: string;
19   name: string;
20
21   @ViewChild('mylist')mylist: IonList;
22
23   constructor(private storageService: StorageService,
24     private plt: Platform,
25     public alertCtrl: AlertController,
26     private toastController: ToastController,
27     private localNotifications: LocalNotifications) {
28
29     this.plt.ready().then(() => {
30       this.loadItems();
31     });
32   }
33
34   // CREATE
35   addItem() {
36     this.newItem.modified = Date.now();
37     this.newItem.id = Date.now();
38     this.storageService.addItem(this.newItem).then(item => {
39       this.newItem = <Item>{};
40       this.showToast('Запис додано!');
41       this.loadItems(); // Or add it to the array directly
42     });
43     var day = new Date(this.birthday);
44     var name1 = new String(this.name);
45     console.log(day);
46
47     this.localNotifications.schedule({
48       text: 'Сьогодні у ' + name1 + 'день народження',
49       sound: 'file://hangout_message.ogg',
50       trigger: {at: day},
51       led: 'FF0000',
52     })
53   }
54 }

```

1

2

3

4

5

6

Рис. 3.9. Синтаксис файлу `home.page.ts`

					KHTEY 121-05-18.MP	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		35

1. Блок з підключеними плагінами в Ionic.
 - 1.1. В даному блоці відбувається підключення готових плагінів із бібліотеки Ionic, які дозволяють використовувати весь функціонал фреймворку.
2. Блок з підключенням сторінок html та css.
 - 2.1. Дане підключення відбувається для того, щоб файли могли взаємодіяти та використовувати функціонал що описаний в них. Без цього підключення, наприклад, не будуть працювати функції сторінки.
3. Блок з ініціалізацією даних.
 - 3.1. В даному блоці відбувається ініціалізація масивів, які використовуються головною сторінкою, а також інших змінних.
4. Блок конструктор
 - 4.1. В блоці відбувається ініціалізація змінних для плагінів, щоб в кодї функції до них можна було швидко звертатися.
5. Функція додавання контакту
 - 5.1. Ця функція спрацьовує при натисканні на кнопку «Додати контакт». В ній всі данні записуються у масив, який в свою чергу виводить данні на сторінку.
6. Функція сповіщень.
 - 6.1.Ця функція перевіряє дату народження контакта, яка була вказана при його створенні та теперішню. І якщо дати співпадають, то додаток надсилає сповіщення користувачу про те, що у певного контакту з його списку сьогодні день народження.

```

55 // READ
56 loadItems() {
57     this.storageService.getItems().then(items => {
58         this.items = items;
59     });
60 }
61
62 // UPDATE
63 updateItem(item: Item) {
64     item.title = `UPDATED: ${item.title}`;
65     item.modified = Date.now();
66
67     this.storageService.updateItem(item).then(item => {
68         this.showToast('Item updated!');
69         this.mylist.closeSlidingItems();
70         this.loadItems();
71     });
72 }
73
74 // DELETE
75 deleteItem(item: Item) {
76     this.storageService.deleteItem(item.id).then(item => {
77         this.showToast('Запис видалено!');
78         this.mylist.closeSlidingItems();
79         this.loadItems();
80     });
81 }
82
83 async doConfirm(item: Item) {
84     const confirm = await this.alertCtrl.create({
85         header: 'Видалити запис?',
86         message: 'Запис буде видалено із Вашого списку контактів?',
87         buttons: [
88             {
89                 text: 'Hi',
90                 handler: () => {
91                 }
92             },
93             {
94                 text: 'Так',
95                 handler: () => {
96                     this.deleteItem(item);
97                 }
98             }
99         ]
100     });
101     await confirm.present();
102 }
103
104 // Helper
105 async showToast(msg) {
106     const toast = await this.toastController.create({
107         message: msg,
108         duration: 2000
109     });
110     toast.present();
111 }
112
113 }

```

7

8

9

10

11

Рис. 3.10. Продовження файлу home.page.ts

7. Функція Read

					KHTEY 121-05-18.MP	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		37

- 7.1. Функція, яка циклом виводить всю інформацію на сторінку.
8. Функція Update
- 8.1. Функція, яка перевіряє данні та оновлює їх, коли був доданий якийсь новий елемент.
9. Функція Delete
- 9.1. Функція, яка дозволяє користувачеві видалити контакт із списку, якщо він уже не потрібен.
10. Функція doConfirm
- 10.1. Даний блок відповідає за впливаючи вікно з попередженням, чи дійсно користувач хоче видалити даний контакт із списку.
11. Функція Helper
- 11.1. Функція, яка просто відображає інформацію внизу екрану про те, що користувач зробив якусь дію. Наприклад додав контакт до списку, або видалив його.
- І для кращого розуміння архітектури додатку, була розроблена UML-діаграма, яка зображує зв'язки між компонентами.

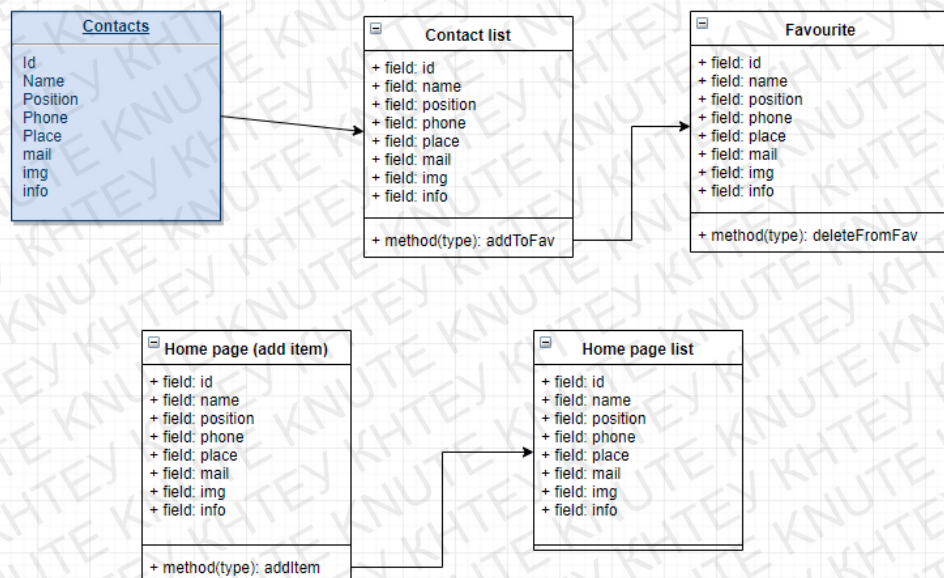


Рис.3.11. UML-діаграма проекту

3.6 Висновок до розділу 3

Для розробки додатку «KNUTE App», був використаний фреймворк Ionic, який дозволяє використовувати технології web-розробки для створення додатків на декілька мобільних платформ.

В роботі використані мови програмування - html, scss, typescript, які дають можливість без глибокого пізнання специфічних для кожної платформи мов програмування, розробляти додатки відразу на декілька операційних систем, що значно економить час на розробку.

Під час розробки додатку було використано основи UX (user experience) патернів, що допомагає створити ергономічний та зручний дизайн, який вміщає всю необхідну інформацію на невеликому екрані, та дає розуміння того, як правильно потрібно розміщувати елементи, який колір вони повинні мати, та як користувач реагує на певні кольори, що є дуже важливим фактором в мобільних додатках.

					<i>KNUTE 121-05-18.MP</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		39

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Під час написання випускної кваліфікаційної роботи, був проведений аналіз ринку мобільних додатків. По результатам, мобільних додатків саме для університетів надзвичайно мало. Тому, зайняти цю нішу дозволило б почати розвиток більш тісної взаємодії університету та студентів. Кожен мав би змогу зв'язатися з потрібним йому співробітником університету, без зайвих пошуків у великому довіднику університету, або ж через інших людей. Вся ця інформація може знаходитися у кишені. Розклад занять, зв'язок з викладачем, без використання його особистого номеру телефону або електронної пошти, створення чатів студентів та викладачів, де всі мали б можливість поширювати додатковий матеріал по лекціям або практикам.

Сам додаток «KNUTE App» розроблено за допомогою web-технологій, а саме: HTML5, CSS3, Angular7 та фреймворк Ionic на основі PhoneGap, який взаємодіє з усіма попередніми технології. В якості мови програмування, використаний стандартний для цього стеку технологій TypeScript, який є логічним продовженням JavaScript, та розширює його можливості.

Також, проведена робота по розумінню того, як працює університет всередині, та які процеси налагодженні між співробітниками. Чого не вистачає, та який функціонал вони б хотіли бачити в додатку.

На даному етапі, додаток знаходиться в розробці, і в роботі представлено демо версію додатку, з частиною реалізованого функціоналу. Зараз, додаток працює локально, тому всі данні зберігаються на пристрої користувача. І при втраті пристрою, дані неможна буде відновити. Проте, якщо цю ідею будуть

					КНТЕУ 121-05-18.МР			
					«Розробка мобільного додатку «KNUTE App»»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		ВП	40	43
Зав. каф.		Криворучко О.В.			Висновки та пропозиції	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		
Керівник		Цюцюра С.В.						
Гарант		Криворучко О.В.						
Розроб		Проява В.В.						

розвивати, то в майбутньому можливе підключення до серверу університету, за допомогою якого можна буде реалізувати інші соціальні функції.

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		41

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Rahat Khanna, Sani Yusuf, Hoc Phan. Ionic : Hybrid Mobile App Development - India. Packt Publishing Ltd, 2017. - 544 с.
2. Mahesh Panhale. Beginning Hybrid Mobile Application Development. – Great Britain. Apress, 2015. – 500 с.
3. Bobby Gill, Jordan Gurrieri. Appsters: A Beginner's Guide to App Entrepreneurship. Blue Label Labs, 2012. – 240 с.
4. Jeff McWherter, Scott Gowell. Professional Mobile Application Development. John Wiley & Sons, 2012. – 432 с.
5. Chris Griffith. Mobile App Development with Ionic 2: Cross-Platform Apps with Ionic, Angular, and Cordova. "O'Reilly Media, Inc.", 2017. – 294 с.
6. Fu Cheng. Build Mobile Apps with Ionic 2 and Firebase: Hybrid Mobile App Development. Apress, 2017 – 252 с.
7. Mike van Drongelen, Adam Dennis, Richard Garabedian, Alberto Gonzalez, Aravind Krishnaswamy. Lean Mobile App Development: Apply Lean startup methodologies to develop successful iOS and Android apps. Packt Publishing Ltd, 2017 – 378 с.
8. Fu Cheng. Build Mobile Apps with Ionic 4 and Firebase: Hybrid Mobile App Development. Apress, 2018 – 435 с.
9. Matt Gifford. PhoneGap Mobile Application Development Cookbook. Packt Publishing Ltd, 2012 - 320 с.
10. Kerri Shotts. PhoneGap 2.X Mobile Application Development. Packt

					<i>KHTEY 121-05-18.MP</i>			
					«Розробка мобільного додатку «KNUTE App»»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		СВД	42	43
Зав. каф.		Криворучко О.В.			Список використаних літературних джерел	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		
Керівник		Цюцюра С.В.						
Гарант		Криворучко О.В.						
Розроб		Проява В.В.						

- Publishing Ltd, 2013 - 388 с.
11. Kerri Shotts. PhoneGap 3.X Mobile Application Hotshot. Packt Publishing Ltd, 2014 - 450 с.
 12. Mark Lasso. Mobile App Development with HTML5. LearnToProgram, Incorporated, 2015 - 350 с.
 13. Kerri Shotts. Mastering PhoneGap Mobile Application Development. Packt Publishing Ltd, 2016 - 392 с.
 14. John Estrella, Rossetta Sornabala. Agile Project Management for Mobile Application Development. CreateSpace Independent Publishing Platform, 2017 - 136 с.
 15. Проява В.В., Цюцюра С.В., Можливості функціоналу додатку «KNUTE App», КНТЕУ, 2019 – 9 с.
 16. Офіційний сайт редактора HTML. [Електронний ресурс]. — Режим доступу: <https://html.com/>
 17. Офіційний сайт CSS. [Електронний ресурс]. — Режим доступу: <https://www.w3.org/Style/CSS/Overview.en.html>
 18. Офіційний сайт Angular. [Електронний ресурс]. — Режим доступу: <https://angular.io/>
 19. Офіційний сайт Ionic Framework. [Електронний ресурс]. — Режим доступу: <https://ionicframework.com/>
 20. Сайт для публікації новин, аналітичних статей, думок, пов'язаних з інформаційними технологіями. [Електронний ресурс]. — Режим доступу: <https://habrahabr.ru>
 21. Веб-сервіс для хостингу ІТ-проектів [Електронний ресурс]. - Режим доступу: <http://www.github.com>

					<i>КНТЕУ 121-05-18.МР</i>	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		43

ДОДАТКИ

Додаток А

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

Для тестування даного проекту, була обрана методика «Manual testing», тобто метод ручного тестування. Цей спосіб є менш затратим та потребує менше навиків, проте дає хороші результати.

Для тестування мобільного додатку, були використанні наступні види тестування:

1. Функціональне тестування

1.1. Були протестовані всі функції.

2. Дослідницьке тестування

2.1. Була запрошена людина з цільової аудиторії, без досвіду тестування, якій надали тестову версію проекту, для того, що б перевірити, чи зрозумілий дизайн та функціонал додатку.

3. Регресивне тестування

3.1. Після того, як відбувалися глобальні зміни в коді додатку, проводилося повторне тестування всіх функцій та компонентів, щоб забезпечитися в тому, що після змін та виправлення багів, весь функціонал працює коректно.

Щоб забезпечити якість проекту і після тестового періоду, тестувальником були розроблені Тест Кейси, які покривають весь функціонал додатку. Завдяки їм, навіть людина без досвіду в тестуванні, може слідувати написаним тест кейсам, та провести перевірку всього функціоналу, після внесення змін при розробці.

Тест кейс має наступні складові:

1. Порядковий номер тест кейсу
2. Описання тестового кейсу
3. Кроки виконання
4. Отриманий результат

5. Очікуваний результат

Тест кейси

1. ТК_01

2. Перевірка завантаження додатку на платформі android

3. Кроки виконання:

3.1. Натиснути на іконку на робочому столі пристрою

3.2. Звернути увагу на екран завантаження

3.3. Перейти на сторінку «Мої контакти»

4. Додаток завантажився та автоматично перейшов на сторінку «Мої контакти»

5. При завантаженні, додаток має автоматично переходити на сторінку «Мої контакти»

1. ТК_02

2. Перевірка завантаження додатку на платформі IOS

3. Кроки виконання:

3.1. Натиснути на іконку на робочому столі пристрою

3.2. Звернути увагу на екран завантаження

3.3. Перейти на сторінку «Мої контакти»

4. Додаток завантажився та автоматично перейшов на сторінку «Мої контакти»

5. При завантаженні, додаток має автоматично переходити на сторінку «Мої контакти»

1. ТК_03

2. Перевірка завантаження додатку на платформі IOS

3. Кроки виконання:

3.1. Натиснути на іконку на робочому столі пристрою

3.2. Звернути увагу на екран завантаження

3.3. Перейти на сторінку «Мої контакти»

4. Додаток завантажився та автоматично перейшов на сторінку «Мої контакти»
5. При завантаженні, додаток має автоматично переходити на сторінку «Мої контакти»

1. ТК_04

2. Перевірка функції додавання нового контакту

3. Кроки виконання:

3.1. Відкрити додаток

3.2. Перейти на сторінку «Мої контакти»

3.3. Натиснути на іконку «+» в правому кутку екрану

3.4. У відкритому вікні ввести відповідні дані у поля

3.5. Натиснути кнопку «Додати запис!»

4. Після натискання кнопки додання запису, новий контакт з'являється на сторінці

5. Після натискання кнопки додання запису, новий контакт має з'явитися на сторінці

1. ТК_05

2. Перевірка функції видалення контакту

3. Кроки виконання:

3.1. Відкрити додаток

3.2. Перейти на сторінку «Мої контакти»

3.3. Натиснути на напис «Детальніше»

3.4. У відкритому вікні натиснути на кнопку «Видалити»

4. Після натискання кнопки видалити, запис видалається

5. Після натискання кнопки додання запису, запис має видалитися

КЕРІВНИЦТВО КОРИСТУВАЧА

Для того, щоб зайти в додаток, потрібно встановити його через спеціальний арк файл (якщо для android системи). Після цього, достатньо просто натиснути на іконку нового додатку, і той сам запуститься.

Після цього, користувач опиниться на головній сторінці – «Мої контакти». Тут він може додавати свої контакти за допомогою спеціальної форми. Щоб відкрити форму, потрібно натиснути на іконку «+».

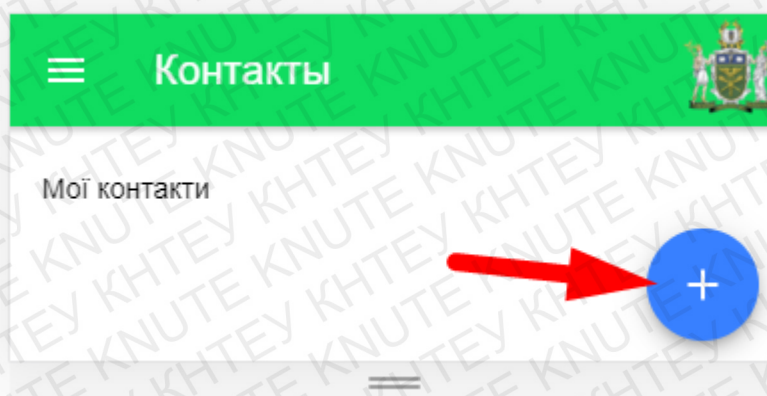


Рис.4.1. Кнопка додавання нового контакту

Після цього, відкриється форма для заповнення даних, в яку потрібно ввести відповідну інформацію про співробітника.

Контакты

Мої контакти

Ім'я

Фото

Посада

Телефон

Пошта

Кабінет

День народження

Інформація

ДОДАТИ ЗАПИСІ

Рис.4.2. Пуста форма для додавання контакту

Однак, для того щоб додати фото співробітника, потрібно скопіювати посилання з іншого ресурсу, де це фото є.



Рис.4.3. Варіант копіювання посилання на зображення

Після цього, скопійоване посилання потрібно вставити в поле «Фото».

Ім'я	Харченко Олександр Анатолійович
Фото	https://knteu.kiev.ua/image/MTU3OQ==/1a714c
Посада	Декан
Телефон	531-49-56
Пошта	a.kharchenko@knteu.edu.ua
Кабінет	Д-103
День народження	20/10/2019
Інформація	Декан

ДОДАТИ ЗАПИС

Рис.4.4. Форма з заповнення інформацією

Після чого, потрібно натиснути на кнопку «Додати запис», і запис буде додано на сторінку «Мої контакти».

 Харченко Олександр Анатолійович Номер телефона - 531-49-56 Пошта - a.kharchenko@knteu.edu.ua Кабінет - Д-103
--

Закрити

День народження - 20/10/2019

Декан

ВИДАЛИТИ

Рис.4.5. Новий запис на сторінці

Запис також можна видалити, натиснувши на кнопку «Видалити» і підтвердивши дію в спеціальному діалоговому вікні.

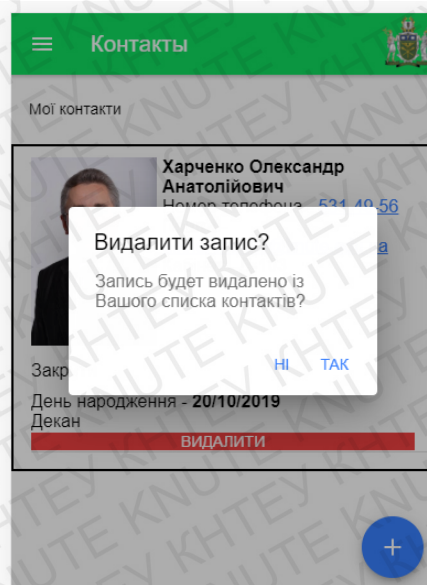


Рис.4.6. Діалогове вікно підтвердження видалення контакту

На сторінці «Довідник», знаходиться список співробітників, яких уже внесено в пам'ять додатку. Вони розсортовані по списку та відділах, в яких працюють.



Рис.4.7. Довідник з уже записаними контактами

Ці записи можна вибірково додати до списку обраних контактів, щоб потрібні контакти завжди були у швидкому доступі.



Рис.4.8. Обрані контакти

Обрані контакти також можна видалити, якщо вони більше не потрібні. Робиться це натисканням на кнопку «Видалити обрані контакти».

Лістинг коду сторінки «Обрані контакти»**Файл favourite.module.ts**

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { Routes, RouterModule } from '@angular/router';

import { IonicModule } from '@ionic/angular';

import { FavouritePage } from './favourite.page';

const routes: Routes = [
  {
    path: '',
    component: FavouritePage
  }
];

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    RouterModule.forChild(routes)
  ],
  declarations: [FavouritePage]
})
export class FavouritePageModule {}

```

Файл Favourite.page.html

```

<ion-header>
  <ion-toolbar color="success">
    <ion-buttons slot="start">
      <ion-menu-button></ion-menu-button>
    </ion-buttons>
    <ion-title>Обрані контакти</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-list>

    <ion-item-sliding *ngFor="let item of selectedItems; let i = index" lines="inset">
      <ion-item class="item">
        <div class="contentBlock">
          <div class="imageBlock">
            
          </div>
          <div class="textBlock">
            <b>{{item.name}}</b><br>
            Посада - <b>{{item.position}}</b><br>
            Номер телефона - <a href="tel:{{item.phone}}">{{item.phone}}</a><br>
            Пошта - <a href="mailto:{{item.mail}}">{{item.mail}}</a><br>
            Кабінет - <b>{{item.place}}</b><br>
          </div>
          <div class="moreBtn" (click)="item.expanded = !item.expanded">

```

```

<div *ngIf="!item.expanded">Детальніше</div>
<div *ngIf="item.expanded">Закрити</div>
</div>
<div class="moreTxt" *ngIf="item.expanded">
  {{item.info}}<br>
</div>
</div>
</ion-item>
</ion-item-sliding>

</ion-list>

<ion-button color="danger" expand="full" (click)="delete(item)">Видалити обрані контакти
<ion-icon slot="end" name="trash"></ion-icon>
</ion-button>

</ion-content>

```

Файл Favourite.page.scss

```

.logo {
  width: 100%;
}

.content {
  padding-top: 10px;
  padding-left: 5px;
  padding-right: 5px;
}

.item {
  //border: 2px solid;
  padding-bottom: 15px;
  margin-bottom: 15px;
}

.category-banner {
  border-left: 8px solid var(--ion-color-secondary); // Лінія з лівої сторони блоку товщиною 8 пікселів синього кольору
  background: var(--ion-color-light); // фоновий світлий колір
  height: 40px; //Висота блоку 40 пікселів
  padding: 10px; //Відступи з усіх сторін по 10 пікселів
}

.contentBlock {
  width: 100%;
}

.imageBlock {
  box-sizing: border-box;
  display: table-cell;
  vertical-align: top;
  padding: 10px 0;
  width: 30%;
}

.textBlock {
  box-sizing: border-box;
  display: table-cell;
  vertical-align: top;
}

```

```
padding: 0 5px;
width: 50%;
padding-top: 10px;
}
```

```
.img {
  max-width: 100%;
  border: 0;
  width: 100%;
}
```

```
.moreBtn {
  text-decoration: underline;
  padding-top: 10px;
}
```

```
.moreTxt {
  height: 100%;
  padding-top: 10px;
  padding-bottom: 10px;
}
```

Файл Favourite.page.specs.ts

```
import { CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { FavouritePage } from './favourite.page';
```

```
describe('FavouritePage', () => {
  let component: FavouritePage;
  let fixture: ComponentFixture<FavouritePage>;
```

```
  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ FavouritePage ],
      schemas: [CUSTOM_ELEMENTS_SCHEMA],
    })
    .compileComponents();
  }));
```

```
  beforeEach(() => {
    fixture = TestBed.createComponent(FavouritePage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });
```

```
  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Favoutire.page.ts

```
import { Component, OnInit } from '@angular/core';
import { EmployeeService } from 'src/app/services/employee.service';
import { ToastController, AlertController } from '@ionic/angular';
```

```
@Component({
  selector: 'app-favourite',
```

```

templateUrl: './favourite.page.html',
styleUrls: ['./favourite.page.scss'],
})
export class FavouritePage implements OnInit {

  //Змінна в якій зберігаються всі товари
  items2 = this.employeeService.getFav();

  // Масив, в якому зберігаються всі ДОДАНІ до обраного товари
  selectedItems = [];

  constructor(
    private employeeService: EmployeeService,
    public alertCtrl: AlertController,
    private toastController: ToastController
  ) { }

  ngOnInit() {

    let selected = {};
    for (let obj of this.items2) {
      if (selected[obj.id]) {
        selected[obj.id].count++;
      } else {
        selected[obj.id] = { ...obj, count: 1 };
      }
    }
    // Вивід всіх елементів корзини в вигляді об'єктів (для того, що б можна було видаляти їх)
    this.selectedItems = Object.keys(selected).map(key => selected[key])
    console.log('items: ', this.selectedItems);

  }

  delete(item) {
    this.employeeService.deleteFavourite(item);
    this.showToast("Записи буде видалено після оновлення сторінки!")
    let index = this.selectedItems.indexOf(item);
    if (index > -1) {
      this.selectedItems.splice(index, 1);
    }
  }

  async showToast(msg) {
    const toast = await this.toastController.create({
      message: msg,
      duration: 2000
    });
    toast.present();
  }

}

```

Лістинг коду сторінки «Мої контакти»**Home.module.ts**

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { IonicModule } from '@ionic/angular';
import { RouterModule } from '@angular/router';
```

```
import { HomePage } from './home.page';
```

```
@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    RouterModule.forChild([
      {
        path: "",
        component: HomePage
      }
    ])
  ],
  declarations: [HomePage]
})
export class HomePageModule {}
```

Home.page.html

```
<ion-header>
  <ion-toolbar color="success">
    <ion-buttons slot="start">
      <ion-menu-button></ion-menu-button>
    </ion-buttons>
    <a slot="end" href="#" onclick="window.open('https://www.knteu.kiev.ua/', '_system'); return false;">
      
    </a>
    <ion-title>
      Контакти
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>

  <ion-list #mylist>

    <ion-list-header>
      <ion-label>Мої контакти</ion-label>
    </ion-list-header>

    <ion-item-sliding *ngFor="let item of items" class="category-block">
      <ion-list class="content">
        <ion-item class="itema">
          <div class="contentBlock">
            <div class="imageBlock">
              
            </div>
```

```

<div class="textBlock">
  <b>{{item.name}}</b><br>
  Номер телефона - <a href="tel:{{item.phone}}">{{item.phone}}</a><br>
  Пошта - <a href="mailto:{{item.mail}}">{{item.mail}}</a><br>
  Кабінет - <b>{{item.place}}</b><br>
</div>
<div class="moreBtn" (click)="item.expanded = !item.expanded">
  <div *ngIf="!item.expanded">Детальніше</div>
  <div *ngIf="item.expanded">Закрити</div>
</div>
<div class="moreTxt" *ngIf="item.expanded">
  <div>
    День народження - <b>{{item.birthday | date: "dd/MM/yyyy"}}</b><br>
  </div>
  <div>
    {{item.info}}<br>
  </div>
  <div>
    <ion-item-option color="danger" (click)="doConfirm(item)">Видалити</ion-item-option>
  </div>
</div>
</ion-item>
</ion-list>
</ion-item-sliding>
</ion-list>

<ion-fab vertical="bottom" horizontal="end" slot="fixed">
  <ion-fab-button (click)="expanded = !expanded">
    <ion-icon *ngIf="!expanded" name="add"></ion-icon>
    <ion-icon *ngIf="expanded" name="close"></ion-icon>
  </ion-fab-button>
</ion-fab>

<div class="addItem" *ngIf="expanded">
  <ion-item>
    <ion-label position="floating">Ім'я</ion-label>
    <ion-input [(ngModel)]="name" [(ngModel)]="newItem.name"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Фото</ion-label>
    <ion-input [(ngModel)]="newItem.photo"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Посада</ion-label>
    <ion-input [(ngModel)]="newItem.position"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Телефон</ion-label>
    <ion-input [(ngModel)]="newItem.phone"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="floating">Пошта</ion-label>
    <ion-input [(ngModel)]="newItem.mail"></ion-input>
  </ion-item>
</div>

```

```

</ion-item>

<ion-item>
  <ion-label position="floating">Кабінет</ion-label>
  <ion-input [(ngModel)]="newItem.place"></ion-input>
</ion-item>

<ion-item>
  <ion-label position="floating">День народження</ion-label>
  <ion-datetime display-
format="DD/MM/YYYY" [(ngModel)]="birthday" [(ngModel)]="newItem.birthday"></ion-datetime>
</ion-item>

<ion-item>
  <ion-label position="floating">Інформація</ion-label>
  <ion-textarea [(ngModel)]="newItem.info"></ion-textarea>
</ion-item>

<ion-button expand="full" (click)="addItem()">Додати запис!</ion-button>

</div>

</ion-content>

```

Home.page.scss

```

.content {
  padding-top: 10px; //Відступ зверху 10 пікселів
}

.itema {
  border: 2px solid; //Чорна рамка товщиною 2 пікселі
  padding-bottom: 15px; //Відступ знизу 15 пікселів
  margin-bottom: 15px; //Відступ між елементами 15 пікселів
}

.contentBlock {
  width: 100%; //Ширина 100% відносно блоку, в якому знаходиться елемент
}

.imageBlock {
  box-sizing: border-box; //Відображення зображення на ширину всього блоку
  display: table-cell; //Відображення зображення на ширину всього блоку
  vertical-align: top; //Вирівнювання по вертикалі
  padding: 10px 0; //Відступ з усіх сторін 10 пікселів
  width: 30%;
}

.textBlock {
  box-sizing: border-box; //Відображення зображення на ширину всього блоку
  display: table-cell; //Відображення зображення на ширину всього блоку
  vertical-align: top; //Вирівнювання по вертикалі
  padding: 0 5px; //Відступ з усіх сторін 5 пікселів
  width: 50%; //Ширина відносно батьківського блоку 50%
  padding-top: 10px; //Відступ зверху 10 пікселів
}

.img {
  max-width: 100%; //Максимальна ширина на блок 100%
  border: 0; //Рамки 0 пікселів
}

```

```

    width: 100%;    //Ширина відносно батьківського блоку 100%
  }

  .moreTxt {
    height: 100%;    //Висота - 100%
    padding-top: 10px; //Відступ зверху 10 пікселів
  }

  .addItem {
    height: 100%;    //Висота - 100%
    padding-top: 37px; //Відступ зверху 37 пікселів
    border: 2px solid; //Чорна рамка товщиною 2 пікселі
  }

```

Home.page.specs.ts

```

import { CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { HomePage } from './home.page';

describe('HomePage', () => {
  let component: HomePage;
  let fixture: ComponentFixture<HomePage>;

  beforeEach(async() => {
    TestBed.configureTestingModule({
      declarations: [ HomePage ],
      schemas: [CUSTOM_ELEMENTS_SCHEMA],
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(HomePage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

Home.page.ts

```

import { Component, ViewChild } from '@angular/core';
import { StorageService, Item } from '../services/storage.service';
import { Platform, ToastController, IonList, AlertController } from '@ionic/angular';
import { LocalNotifications } from '@ionic-native/local-notifications/ngx';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  items: Item[] = [];

```

```

newItem: Item = <Item>{};

birthday: string;
name: string;

@ViewChild('mylist')mylist: IonList;

constructor(private storageService: StorageService,
private plt: Platform,
public alertCtrl: AlertController,
private toastController: ToastController,
private localNotifications: LocalNotifications) {

    this.plt.ready().then(() => {
        this.loadItems();
    });
}

// CREATE
addItem() {
    this.newItem.modified = Date.now();
    this.newItem.id = Date.now();
    this.storageService.addItem(this.newItem).then(item => {
        this.newItem = <Item>{};
        this.showToast('Запис додано!')
        this.loadItems(); // Or add it to the array directly
    });
    var day = new Date(this.birthday);
    var name1 = new String(this.name);
    console.log(day);

    this.localNotifications.schedule({
        text: 'Сьогодні у ' + name1 + ' день народження',
        sound: 'file://hangout_massage.ogg',
        trigger: {at: day},
        led: 'FF0000',
    })
}

// READ
loadItems() {
    this.storageService.getItems().then(items => {
        this.items = items;
    });
}

// UPDATE
updateItem(item: Item) {
    item.title = `UPDATED: ${item.title}`;
    item.modified = Date.now();

    this.storageService.updateItem(item).then(item => {
        this.showToast('Item updated!');
        this.mylist.closeSlidingItems();
        this.loadItems();
    });
}

// DELETE

```

```

deleteItem(item: Item) {
  this.storageService.deleteItem(item.id).then(item => {
    this.showToast('Запис видалено!');
    this.mylist.closeSlidingItems();
    this.loadItems();
  });
}

async doConfirm(item: Item) {
  const confirm = await this.alertCtrl.create({
    header: 'Видалити запис?',
    message: 'Запис буде видалено із Вашого списку контактів?',
    buttons: [
      {
        text: 'Ні',
        handler: () => {}
      },
      {
        text: 'Так',
        handler: () => {
          this.deleteItem(item);
        }
      }
    ]
  });
  await confirm.present();
}

// Helper
async showToast(msg) {
  const toast = await this.toastController.create({
    message: msg,
    duration: 2000
  });
  toast.present();
}

```

Лістинг коду сторінки «Список контактів»**List.module.ts**

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { IonicModule } from '@ionic/angular';
import { RouterModule } from '@angular/router';
```

```
import { ListPage } from './list.page';
```

```
@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    RouterModule.forChild([
      {
        path: '',
        component: ListPage
      }
    ])
  ],
  declarations: [ListPage]
})
export class ListPageModule {}
```

List.page.html

```
<ion-header>
  <ion-toolbar color="success">
    <ion-buttons slot="start">
      <ion-menu-button></ion-menu-button>
    </ion-buttons>
    <ion-title>
      Довідник
    </ion-title>
    <ion-button class="icon" slot="end" (click)="openFav()">
      <ion-icon slot="icon-only" name="star"></ion-icon>
    </ion-button>
  </ion-toolbar>
</ion-header>

<ion-content>

  <div class="object" *ngFor="let cat of items">

    <ion-row no-padding class="category-banner">
      <ion-col text-left button tappable (click)="cat.expanded = !cat.expanded" align-self-center">
        {{cat.category}}
      </ion-col>
    </ion-row>

    <ion-list *ngIf="cat.expanded" class="content">
      <ion-item class="item" *ngFor="let emp of cat.employee">
        <div class="contentBlock">
          <div class="imageBlock">
```

```

        
    </div>
    <div class="textBlock">
        <b>{{emp.name}}</b><br>
        Посада - <b>{{emp.position}}</b><br>
        Номер телефона - <a href="tel:{{emp.phone}}">{{emp.phone}}</a><br>
        Пошта - <a href="mailto:{{emp.mail}}">{{emp.mail}}</a><br>
        Кабінет - <b>{{emp.place}}</b><br>
        <ion-button color="success" (click)="addToFav(emp)">Додати в обране
        <!-- <ion-icon color="danger" slot="icon-only" name="heart"></ion-icon> -->
    </ion-button>
    </div>
    <div class="moreBtn" (click)="emp.expanded = !emp.expanded">
    <div *ngIf="!emp.expanded">Детальніше</div>
    <div *ngIf="emp.expanded">Закрити</div>
    </div>
    <div class="moreTxt" *ngIf="emp.expanded">
        {{emp.info}}<br>
    </div>
    </div>
</ion-item>
</ion-list>
</div>
</ion-content>

```

List.page.scss

```

.logo {
    width: 100%;
}

.content {
    padding-top: 10px;
    padding-left: 5px;
    padding-right: 5px;
}

.item {
    //border: 2px solid;
    padding-bottom: 15px;
    margin-bottom: 15px;
}

.category-banner {
    border-left: 8px solid var(--ion-color-secondary); // Лінія з лівої сторони блоку товщиною 8 пікселів синього кольору
    background: var(--ion-color-light); // фоновий світлий колір
    height: 60px; //Висота блоку 40 пікселів
    padding: 10px; //Відступи з усіх сторін по 10 пікселів
}

.contentBlock {
    width: 100%;
}

.imageBlock {
    box-sizing: border-box;
    display: table-cell;
}

```

```

    vertical-align: top;
    padding: 10px 0;
    width: 30%;
  }

  .textBlock {
    box-sizing: border-box;
    display: table-cell;
    vertical-align: top;
    padding: 0 5px;
    width: 50%;
    padding-top: 10px;
  }

  .img {
    max-width: 100%;
    border: 0;
    width: 100%;
  }

  .moreBtn {
    text-decoration: underline;
    padding-top: 10px;
  }

  .moreTxt {
    height: 100%;
    padding-top: 10px;
    padding-bottom: 10px;
  }

```

List.page.specs.ts

```

import { CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
import { ComponentFixture, TestBed, async } from '@angular/core/testing';

import { ListPage } from './list.page';

describe('ListPage', () => {
  let component: ListPage;
  let fixture: ComponentFixture<ListPage>;
  let listPage: HTMLElement;

  beforeEach(async() => {
    TestBed.configureTestingModule({
      declarations: [ ListPage ],
      schemas: [CUSTOM_ELEMENTS_SCHEMA],
    })
    .compileComponents();
  });

  beforeEach(async () => {
    fixture = await TestBed.createComponent(ListPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

```

```

it('should have a list of 10 elements', () => {
  listPage = fixture.nativeElement;
  const items = listPage.querySelectorAll('ion-item');
  expect(items.length).toEqual(10);
});

```

```

});

```

List.page.ts

```

import { Component, OnInit } from '@angular/core';
import { EmployeeService } from 'src/app/services/employee.service';
import { Router } from '@angular/router';
import { ToastController, AlertController } from '@ionic/angular';

```

```

@Component({
  selector: 'app-list',
  templateUrl: 'list.page.html',
  styleUrls: ['list.page.scss']
})

```

```

export class ListPage implements OnInit {

```

```

  items = [];
  favour = [];

```

```

  constructor(
    private employeeService: EmployeeService,
    private router: Router,
    public alerCtrl: AlertController,
    private toastController: ToastController
  ) {}

```

```

  ngOnInit() {
    this.items = this.employeeService.getEmployee();
    this.favour = this.employeeService.getFav();
  }

```

```

  //Функция додавання товару в обране

```

```

  addToFav(emp) {
    this.employeeService.addProductToFav(emp);
    this.showToast('Запис додано!')
  }

```

```

  //Функція переходу до обраного

```

```

  openFav() {
    this.router.navigate(['favourite'])
  }

```

```

  async showToast(msg) {
    const toast = await this.toastController.create({
      message: msg,
      duration: 2000
    });
    toast.present();
  }

```

```

}

```

Employee.service.specs.ts

```

import { TestBed } from '@angular/core/testing';

```

```
import { EmployeeService } from './employee.service';

describe('EmployeeService', () => {
  beforeEach(() => TestBed.configureTestingModule({}));

  it('should be created', () => {
    const service: EmployeeService = TestBed.get(EmployeeService);
    expect(service).toBeTruthy();
  });
});
```

Employee.service.ts

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class EmployeeService {

  private data = [
    {
      category: 'Керівництво',
      expanded: true,
      employee: [
        {
          id: 0,
          name: 'МАЗАРАКІ Анатолій Антонович ',
          position: 'Ректор',
          phone: '1002',
          place: 'A-246',
          mail: 'rector@knute.edu.ua',
          img: 'https://knute.edu.ua/image/MjAwNg==/7c2fcc0a27155057d3aa60100787e4ec.jpg',
          info: 'Доктор економічних наук, професор, академік Національної академії педагогічних наук України, заслужений діяч науки і техніки України, лауреат Державної премії України в галузі науки і техніки, кавалер ордена князя Ярослава Мудрого, лауреат Премії Кабінету Міністрів України за розроблення і впровадження інноваційних технологій, голова наукової секції "Економіка" МОН України, голова редколегій журналів "Вісник КНТЕУ", "Товари та ринки", "Зовнішня торгівля: економіка, фінанси, право", член редколегії журналу "Економіка України".'
        },
        {
          id: 1,
          name: 'ФЕДОРОВА Катерина Михайлівна',
          position: 'Приймальня ректора',
          phone: '1002',
          place: 'A-246',
          mail: 'rector@knteu.kiev.ua',
          img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
          info: ""
        },
        {
          id: 3,
          name: 'БАЗАРОВА Марина Юріївна',
          position: 'Приймальня',
          phone: '1023 , 531-49-32',
          place: 'A-258',
          mail: 'marina@knteu.kiev.ua',
          img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
          info: ""
        }
      ]
    }
  ]
}
```

```

    },
    {
      id: 4,
      name: 'ШАПОВАЛІ Світлана Леонідівна',
      position: 'Проректор з науково-педагогічної роботи',
      phone: '1012',
      place: 'A-250',
      mail: 'shapoval@knteu.kiev.ua',
      img: 'https://knteu.edu.ua/file/MjA=/a789ea738da729321f1ecf58860c4ae0.jpg',
      info: 'кандидат технічних наук, доцент.'
    },
    {
      id: 5,
      name: 'РЯБОКЛЯЧ Олена Валеріївна',
      position: 'Приймальня',
      phone: '1012 , 531-47-40',
      place: 'A-250',
      mail: '',
      img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
      info: ''
    },
    {
      id: 6,
      name: 'МЕЛЬНИЧЕНКО Світлана Володимирівна',
      position: 'Проректор з наукової роботи',
      phone: '1013',
      place: 'A-250',
      mail: 'melnichenko@knteu.kiev.ua',
      img: 'https://knteu.edu.ua/file/MjA=/51feb864b45a9714817ea52bfb1ce58e.jpg',
      info: 'доктор економічних наук, професор.'
    },
    {
      id: 7,
      name: 'СОРОКА Тетяна Миколаївна',
      position: 'Приймальня',
      phone: '1013 , 531-47-30',
      place: 'A-250',
      mail: 'soroka@knteu.kiev.ua',
      img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
      info: ''
    },
    {
      id: 8,
      name: 'САЙ Валерій Миколайович',
      position: 'Проректор з науково-педагогічної роботи та міжнародних зв'язків',
      phone: '1009',
      place: 'A-250',
      mail: 'melnichenko@knteu.kiev.ua',
      img: 'https://knteu.edu.ua/file/MjA=/0f286d8633ccbd424da5cea67f0c432c.jpg',
      info: 'кандидат економічних наук, доцент, дипломатичний радник І класу, координатор проєктів E RASMUS+, ERASMUS MUNDUS, JEAN MONNET, TEMPUS.'
    },
    {
      id: 9,
      name: 'ЧУДІНОВИЧ Марія Василівна',
      position: 'Приймальня',
      phone: '1009 , 531-49-51',
      place: 'A-229',

```

```

mail: "",
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ""
},
{
id: 10,
name: 'БОБК Галина Миколаївна',
position: 'Проректор з адміністративно-господарської роботи',
phone: '1387',
place: 'Гурт. № 4',
mail: 'vovk@knteu.kiev.ua',
img: 'https://knteu.edu.ua/image/MjAwNg==/c6c45f28c8d820d0e2d690e3504cc330.JPG',
info: 'кандидат економічних наук, доцент, дипломатичний радник І класу, координатор проєктів E RASMUS+, ERASMUS MUNDUS, JEAN MONNET, TEMPUS.'
},
{
id: 11,
name: 'ЯЩЕНКО Альона Валентинівна',
position: 'Приймальня',
phone: '14-43 , 531-48-43',
place: 'Гурт. №4',
mail: "",
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ""
},
{
id: 12,
name: 'ОХРИМЕНКО Володимир Іванович',
position: 'Помічник проректора з адміністративно-господарської роботи',
phone: '513-22-36',
place: 'Л-224',
mail: "",
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ""
}
},
{
category: 'Приймальна комісія',
employee: [
{
id: 13,
name: 'ОСИКА Віктор Анатолійович',
position: 'Відповідальний секретар',
phone: '1680 , 1220 , 531-47-94 , 513-77-63 , 531-31-80',
place: 'B-307 , A-337',
mail: 'osyka@knteu.kiev.ua',
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ""
},
{
id: 14,
name: 'Приймальня',
position: "",
phone: '1681 , 1682 , 1683 , 1684 , 531-31-81 , 531-31-84 (факс)',
place: 'B-306',

```

```

    mail: "",
    img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
    info: ""
  },
  {
    id: 15,
    name: 'Архів приймальної комісії',
    position: "",
    phone: '1685',
    place: 'B-305',
    mail: "",
    img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
    info: ""
  },
  {
    id: 16,
    name: 'Комп'ютерний центр приймальної комісії',
    position: "",
    phone: '1686',
    place: 'B-303',
    mail: "",
    img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
    info: ""
  },
  {
    id: 17,
    name: 'ПІКУЛІЦЬКА Юлія Олегівна',
    position: 'Уповноважена особа з питань прийняття та розгляду електронних заяв, адміністратор ЄД ЕБО',
    phone: '1450',
    place: 'B-301',
    mail: "",
    img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
    info: ""
  },
  {
    id: 18,
    name: 'Інформаційно-консультативний центр приймальної комісії',
    position: "",
    phone: '1043 , 1044 , 531-48-88',
    place: 'B-106',
    mail: "",
    img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
    info: ""
  },
  ]
},
{
  category: 'Вчена рада',
  employee: [
    {
      id: 0,
      name: 'МАЗАРАКІ Анатолій Антонович ',
      position: 'Ректор',
      phone: '1002',

```

```

        place: 'A-246',
        mail: 'rector@knute.edu.ua',
        img: 'https://knute.edu.ua/image/MjAwNg==/7c2fcc0a27155057d3aa60100787e4ec.jpg',
        info: 'Доктор економічних наук, професор, академік Національної академії педагогічних наук України, заслужений діяч науки і техніки України, лауреат Державної премії України в галузі науки і техніки, кавалер ордена князя Ярослава Мудрого, лауреат Премії Кабінету Міністрів України за розроблення і впровадження інноваційних технологій, голова наукової секції "Економіка" МОН України, голова редколегій журналів "Вісник КНТЕУ", "Товари та ринки", "Зовнішня торгівля: економіка, фінанси, право", член редколегії журналу "Економіка України".'
    },
    {
        id: 19,
        name: 'ТАРАСЮК Михайло Вікторович',
        position: 'Вчений секретар',
        phone: '1006 , 531-49-14',
        place: 'A-235',
        mail: '',
        img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
        info: ''
    },
    {
        category: 'Методична рада',
        employee: [
            {
                id: 0,
                name: 'МАЗАРАКІ Анатолій Антонович ',
                position: 'Ректор',
                phone: '1002',
                place: 'A-246',
                mail: 'rector@knute.edu.ua',
                img: 'https://knute.edu.ua/image/MjAwNg==/7c2fcc0a27155057d3aa60100787e4ec.jpg',
                info: 'Доктор економічних наук, професор, академік Національної академії педагогічних наук України, заслужений діяч науки і техніки України, лауреат Державної премії України в галузі науки і техніки, кавалер ордена князя Ярослава Мудрого, лауреат Премії Кабінету Міністрів України за розроблення і впровадження інноваційних технологій, голова наукової секції "Економіка" МОН України, голова редколегій журналів "Вісник КНТЕУ", "Товари та ринки", "Зовнішня торгівля: економіка, фінанси, право", член редколегії журналу "Економіка України".'
            },
            {
                id: 2,
                name: 'ПРИТУЛЬСЬКА Наталія Володимирівна',
                position: 'Перший проректор з науково-педагогічної роботи',
                phone: '1023',
                place: 'A-258',
                mail: 'pritulska@knteu.kiev.ua',
                img: 'https://knute.edu.ua/image/MjAwNg==/e28bc0d81bffe49f3260807c9c870e3.jpg',
                info: 'доктор технічних наук, професор, голова спеціалізованої вченої ради Д 26.055.02.'
            },
            {
                id: 20,
                name: 'БОЖКО Тетяна Василівна',
                position: 'Секретар',
                phone: '1100 , 531-47-33',
                place: 'A-512',
                mail: '',
                img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
            }
        ]
    }
]

```

```

    info: "
  },
]
},
{
  category: 'Методична рада',
  employee: [
    {
      id: 0,
      name: 'МАЗАРАКІ Анатолій Антонович ',
      position: 'Ректор',
      phone: '1002',
      place: 'A-246',
      mail: 'rector@knute.edu.ua',
      img: 'https://knute.edu.ua/image/MjAwNg==/7c2fcc0a27155057d3aa60100787e4ec.jpg',
      info: 'Доктор економічних наук, професор, академік Національної академії педагогічних наук України, заслужений діяч науки і техніки України, лауреат Державної премії України в галузі науки і техніки, кавалер ордена князя Ярослава Мудрого, лауреат Премії Кабінету Міністрів України за розроблення і впровадження інноваційних технологій, голова наукової секції "Економіка" МОН України, голова редколегій журналів "Вісник КНТЕУ", "Товари та ринки", "Зовнішня торгівля: економіка, фінанси, право", член редколегії журналу "Економіка України".'
    },
    {
      id: 21,
      name: 'БАРАБАШІ Наталія Степанівна ',
      position: 'Вчений секретар',
      phone: '1198 , 1099 , 531-42-18',
      place: 'Д-306 , А-329',
      mail: 'd26.055.01_knteu@knteu.edu.ua',
      img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
      info: "
    },
    {
      id: 2,
      name: 'ПРИТУЛЬСЬКА Наталія Володимирівна',
      position: 'Перший проректор з науково-педагогічної роботи',
      phone: '1023',
      place: 'A-258',
      mail: 'priteluska@knteu.kiev.ua',
      img: 'https://knute.edu.ua/image/MjAwNg==/e28bc0d81bfff49f3260807c9c870e3.jpg',
      info: 'доктор технічних наук, професор, голова спеціалізованої вченої ради Д 26.055.02.'
    },
    {
      id: 22,
      name: 'СИДОРЕНКО Олена Володимирівна',
      position: 'Вчений секретар ',
      phone: '1233',
      place: 'A-115 , А-329',
      mail: 'd26.055.02_knteu@knteu.edu.ua',
      img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
      info: "
    },
    {
      id: 23,
      name: 'ХАРЧЕНКО Ольга Григорівна',
      position: 'В.о. технічного секретаря',
      phone: '1098 , 531-49-72',
      place: 'A-329',

```

```

mail: 'H_olga@i.ua',
img: 'https://knute.edu.ua/image/OTg=/33c46985861dd3115ef2258da8f39622.jpg',
info: "
},
{
  id: 24,
  name: 'ЧУГУНОВ Ігор Якович ',
  position: 'Голова спеціалізованої вченої ради Д 26.055.03 (з економічних наук)',
  phone: '1202 , 531-47-68',
  place: 'A-218',
  mail: "",
  img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
  info: "
},
{
  id: 25,
  name: 'КУЧЕР Галина Вікторівна',
  position: 'Вчений секретар',
  phone: '1203',
  place: 'A-213',
  mail: 'd26.055.03_knteu@knteu.kiev.ua',
  img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
  info: "
},
{
  id: 26,
  name: 'КОРОЛЬЧУК Микола Степанович',
  position: 'Голова спеціалізованої вченої ради К 26.055.04 (з психологічних наук)',
  phone: '529-21-56',
  place: 'M-104',
  mail: "",
  img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
  info: "
},
{
  id: 27,
  name: 'МИРОНЕЦЬ Сергій Миколайович',
  position: 'Вчений секретар',
  phone: '1358 , 529-34-87',
  place: 'M-103',
  mail: 'msm@email.ua',
  img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
  info: "
},
{
  id: 28,
  name: 'КОЗАЧЕНКО Оксана Володимирівна',
  position: 'В.о. технічного секретаря',
  phone: '529-40-38',
  place: 'M-102',
  mail: 'oksana1216new@gmail.com',
  img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
  info: "
},
{

```

```

id: 29,
name: 'ЛАГУТИН Василь Дмитрович',
position: 'Голова спеціалізованої вченої ради К 26.055.05 (з економічних наук)',
phone: '1286 , 531-47-66',
place: 'A-428',
mail: '',
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ''
},
{
id: 30,
name: 'ШТУНДЕР Ірина Олександрівна',
position: 'Вчений секретар',
phone: '1289',
place: 'A-432',
mail: 'iriwa_w@ukr.net',
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ''
},
{
id: 31,
name: 'МЕЛЬНИК Тетяна Миколаївна',
position: 'Голова спеціалізованої вченої ради Д 26.055.06 (з економічних наук)',
phone: '1306 , 531-47-46',
place: 'A-420',
mail: '',
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ''
},
{
id: 32,
name: 'КОРОВАЙЧЕНКО Наталія Юріївна',
position: 'Вчений секретар',
phone: '1310 , 531-48-74',
place: 'A-415',
mail: '',
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ''
},
{
id: 33,
name: 'РОМАТ Євгеній Вікторович',
position: 'Голова спеціалізованої вченої ради К 26.055.07 (з державного управління)',
phone: '1296 , 531-48-37 , 513-23-45',
place: 'A-403',
mail: '',
img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
info: ''
},
{
id: 34,
name: 'МАЛИШ Наталія Андріївна',
position: 'Вчений секретар',
phone: '1289',
place: 'A-432',

```

```

    mail: "",
    img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
    info: ""
  },
]
},
{
  category: 'Профспілка працівників та студентів університету',
  employee: [
    {
      id: 35,
      name: 'БАЙ Сергій Іванович',
      position: 'Голова профкому',
      phone: '1301 , 531-47-76',
      place: 'А-326',
      mail: 'bay@knteu.kiev.ua',
      img: 'https://knteu.edu.ua/file/MjA=/51945b20a779e28bf1ca9269e85fffe9.jpg',
      info: ""
    },
    {
      id: 36,
      name: 'ГОЛІК Оксана Василівна',
      position: 'Заступник голови',
      phone: '1223 , 531-47-96',
      place: 'А-329',
      mail: "",
      img: 'https://knteu.edu.ua/file/MTIyMzc=/63f04e02e42c95e42873a5fba1da371b.JPG',
      info: ""
    },
    {
      id: 37,
      name: 'ГОЛОВІНА Дар'я Вікторівна',
      position: 'Заступник голови',
      phone: '1182 , 531-48-87',
      place: 'Д-408',
      mail: 'dvdkarpenko88@gmail.com',
      img: 'https://stat.knteu.kiev.ua/file/MjkzNg==/9f49b8dc3473054e516f33a89d01523f.JPG',
      info: ""
    },
    {
      id: 38,
      name: 'КЛИМОВА Надія Олександрівна',
      position: 'Заступник голови',
      phone: '1088 , 531-48-13',
      place: 'А-252',
      mail: 'nmk@knteu.kiev.ua',
      img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',
      info: ""
    },
    {
      id: 39,
      name: 'ЧИРИК Людмила Дмитрівна',
      position: 'Головний бухгалтер профкому',
      phone: '1019 , 531-47-35 , 513-64-90 (факс)',
      place: 'Д-104',
      mail: 'profspilka@knteu.kiev.ua',
      img: 'https://www.canyonparkutah.com/wp-content/uploads/2018/10/contact-placeholder-square-600x600.jpg',

```

```

        info: "
    },
  ]
}
}

private fav = [];

getEmployee() {
  return this.data;
}

//Вивід даних в обране
getFav() {
  return this.fav;
}

//Додавання товару в обране
addProductToFav(product) {
  this.fav.push(product);
}

//Функція видалення товару з обраного
deleteFavourite(product) {
  this.fav.splice(product);
}
}

Storage.service.specs.ts
import { TestBed } from '@angular/core/testing';

import { StorageService } from './storage.service';

describe('StorageService', () => {
  beforeEach(() => TestBed.configureTestingModule({}));

  it('should be created', () => {
    const service: StorageService = TestBed.get(StorageService);
    expect(service).toBeTruthy();
  });
});

Storage.service.ts
import { Injectable } from '@angular/core';
import { Storage } from '@ionic/storage';

export interface Item {
  id: number,
  title: string,
  value: string,
  modified: number
}

const ITEMS_KEY = 'my-items';

@Injectable({
  providedIn: 'root'
})
export class StorageService {

```

```

constructor(private storage: Storage) { }

// CREATE
addItem(item: Item): Promise<any> {
  return this.storage.get(ITEMS_KEY).then((items: Item[]) => {
    if (items) {
      items.push(item);
      return this.storage.set(ITEMS_KEY, items);
    } else {
      return this.storage.set(ITEMS_KEY, [item]);
    }
  });
}

// READ
getItems(): Promise<Item[]> {
  return this.storage.get(ITEMS_KEY);
}

// UPDATE
updateItem(item: Item): Promise<any> {
  return this.storage.get(ITEMS_KEY).then((items: Item[]) => {
    if (!items || items.length === 0) {
      return null;
    }

    let newItems: Item[] = [];

    for (let i of items) {
      if (i.id === item.id) {
        newItems.push(item);
      } else {
        newItems.push(i);
      }
    }

    return this.storage.set(ITEMS_KEY, newItems);
  });
}

// DELETE
deleteItem(id: number): Promise<Item> {
  return this.storage.get(ITEMS_KEY).then((items: Item[]) => {
    if (!items || items.length === 0) {
      return null;
    }

    let toKeep: Item[] = [];

    for (let i of items) {
      if (i.id !== id) {
        toKeep.push(i);
      }
    }

    return this.storage.set(ITEMS_KEY, toKeep);
  });
}

```