

Test Scenario

ឈានអីឡូនិក	រៀលនៃពាណិជ្ជកម្ម	6530200142
ក្រុងពាណិជ្ជកម្ម	ស្ថាបន្ទូរ	6530200444
អាជីវកម្ម	ស្ថាបន្ទូរ	6530200410
អាជីវកម្ម	ស្ថាបន្ទូរ	6530200496
ឈាន រាជធានី	សំណងជនុយដ្ឋាន	6530200327
អាយុរិបាយទេសចរណ៍	ចំណាំ	6530200711

โครงการ: Bank														
รหัสสถานการณ์การทดสอบ: TSB001		ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์พันธุ์ ตั้ง												
ฟังก์ชันที่ทดสอบ: add Account (Account account) ใน Bank.java		วันที่เขียน: 10/10/2024												
เงื่อนไขก่อนทดสอบ: ต้องมี Object Bank และ Account ต้องมี Account ที่จะเพิ่มลงใน Bank														
เงื่อนไขหลังทดสอบ: Account ที่เพิ่มเข้าไปอยู่ใน ArrayList ใน Bank														
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank และ Account (name, accountNumber, balance) เป็น string, string, double ตามลำดับ 2. ใช้งาน func addAccount() 3. ตรวจสอบ account อยู่ใน ArrayList ใน Bank														
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ								
TCB01	สร้าง Account ในแล้วเพิ่มลงใน ArrayList ใน Bank	Account("bob", "001", 1000)	Account bob Account ID 001 อยู่ใน ArrayList Bank	Account bob อยู่ใน ArrayList Bank	ผ่าน									
TCB02	สร้าง Account ที่มีอยู่แล้วใน ArrayList ใน Bank	Account("bob", "001", 1000)	ต้องมี Account bob ใน ArrayList ค่าไม่ถูกเปลี่ยน	Account ("bob", "001", 100) อยู่ใน ArrayList	ไม่ผ่าน									

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB002			ผู้เขียนสถานการณ์การทดสอบ: พี่ๆ วงศ์พัฒนา คุ้ง							
ฟังก์ชันที่ทดสอบ: remove Account(Account account) ใน Bank.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องมี Account อยู่แล้วใน ArrayList ของ Bank										
เงื่อนไขหลังทดสอบ: Account หาย ออกจาก ArrayList ของ Bank										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank ด้วย Account ไปรูปแบบ (name, accountNumber, balance) เป็น string, string, double ตามลำดับ 2. ใช้งาน func addAccount() 4. ตรวจสอบว่าบัญชีถูกลบออกจาก ArrayList ของ Bank 3. ใช้งาน func remove Account()										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TCB01	ลงทะเบียนบัญชี 1 ตัวใน ArrayList	Account("bob", "001", 1000)	ปัจจุบันมีบัญชี 1 ตัวใน ArrayList	ปัจจุบันไม่มีบัญชีใน ArrayList	ผ่าน					
TCB02	ลงทะเบียนบัญชี 2 ตัวใน ArrayList และลบ 1 ตัว	Account("jcop", "002", 2000)	บัญชีมีจำนวน 2 ตัวใน ArrayList	บัญชีมีจำนวน 1 ตัวใน ArrayList	ผ่าน					

โครงการ: Bank									
รหัสสถานการณ์การทดสอบ: TSB003			ผู้เขียนสถานการณ์การทดสอบ: นฤทธิ์ พัฒนา ตั้ง						
พัฒนาชั้นทดสอบ: deposit Money(Account account, double amount) by Bank.java				วันที่เขียน: 10/10/2024					
เงื่อนไขก่อนทดสอบ: ต้องมี Account อยู่แล้วใน ArrayList ของ Bank amount ต้องมากกว่า 0									
เงื่อนไขหลังทดสอบ: balance ต้องมี เป็น เท่ากับ จำนวนที่ฝาก (amount) + เงินเดิม (Balance)									
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank ด้วย constructor (name, accountNumber, balance) ที่รับ string, string, double ตามลำดับ 2. ใช้งาน func addAccount() 3. ใช้งาน func depositMoney() 4. ตรวจสอบยอดเงินใน Account ArrayList ของ Bank									
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ			
TCB01	ฝากเงิน จำนวนปกติ เท่า 1000.0	Account("bob", "001", 1000) n=amount = 1000.0	Balance มากกว่า 1000.0 1000.0 Balance มากกว่า 2000.0	Balance มากกว่า 1000.0 1000.0 Balance มากกว่า 2000.0	ผ่าน				
TCB02	ฝากเงิน จำนวน 0 เท่า 0.0	Account("bob", "001", 1000) n=amount = 0.0	Balance มากกว่า 1000.0 Balance มากกว่า 1000.0	Balance มากกว่า 1000.0 Balance มากกว่า 1000.0	ผ่าน				
TCB03	ฝากเงิน จำนวน ติดลบ เท่า -1000.0	Account("bob", "001", 1000.0) n=amount = -1000.0	balance คงที่ 0.0	Balance มากกว่า 1000.0 Balance มากกว่า 0.0	ไม่ผ่าน				

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TCB004			ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์สุข ศรีวุฒิ							
พัฒนาชั้นทดสอบ: withdrawMoney(Account account, double amount) ใน Bank.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องมี Account อยู่แล้วใน ArrayList ของ Bank amount ต้องมากกว่า 0 และไม่เกินยอดคงเหลือ (Balance)										
เงื่อนไขหลังทดสอบ: Balance ต้องมีเงินคงเหลือ จำนวนเดือน เงินเดือน (Balance) - ยอด (amount)										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank ด้วย Account 3 ตัว (name, accountNumber, balance) เป็น string, string, double ตามลำดับ 2. ใช้งาน func addAccount() 4. ตรวจสอบยอดเงินใน Account ArrayList ของ Bank 3. ใช้งาน func withdrawMoney()										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TCB01	ถอนเงิน จำนวนมากกว่า 0 หรือลบ Balance	Account("bob", "001", 1000) n=amount = 1000.0	Balance มีเงินคงเหลือ 1000.0 Balance ลบ = 0.0	Balance มีเงินเหลือ 1000.0 Balance ลบ = 0.0	ผ่าน					
TCB02	ถอนเงิน จำนวน 0 ออกจากบัญชี	Account("bob", "001", 1000) n=amount = 0.0	Balance มีเงินเท่าเดิม Balance ลบ = 1000.0	Balance มีเงินเท่าเดิม Balance ลบ = 1000.0	ผ่าน					
TCB03	ถอนเงิน จำนวนน้อยกว่า Balance	Account("bob", "001", 1000.0) n=amount = 2000.0	ถอนเงิน超出了余额	Balance มีเงินคงเหลือ Balance ลบ = -1000.0	ไม่ผ่าน					

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TCB005			ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์พัฒนา ตุ้ย							
ฟังก์ชันที่ทดสอบ: getAccount() ใน Bank.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องมี Account อยู่แล้วใน ArrayList ของ Bank อย่างน้อย 1 Account										
เงื่อนไขหลังทดสอบ: ต้องมี 2 รายการ Account อยู่ใน ArrayList ของ Bank										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank และ Account โดยตั้ง (name, accountNumber, balance) เป็น string, string, double ตามลำดับ 2. ให้调用 func getAccounts() 3. ตรวจสอบว่า 2 รายการ Account อยู่ใน ArrayList อยู่แล้ว										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TCB01	ตรวจสอบ Account 2 รายการ ใน ArrayList	Account("bob", "001", 1000) Account("scop", "002", 2000)	ลื้นค่า รายการ 2 รายการ เพิ่มเท่านั้น	ลื้นค่า รายการ 2 รายการ เพิ่มเท่านั้น	ผ่าน					
TCB02	ตรวจสอบ Account ไม่มีรายการ ใน ArrayList	ไม่มีรายการ 0	ลื้นค่า ร่างโครงสร้าง null	ลื้นค่า ร่างโครงสร้าง null	ผ่าน					

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB006			ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์สุวรรณ ดีกรี							
ฟังก์ชันที่ทดสอบ: getName() ใน Account.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องมี Object Account ก่อนจะทดสอบต่อไป										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank และ Account โดย parameter (name, accountNumber, balance) คือ string, string, double ตามลำดับ 2. ให้ invoke func getName() 3. ตรวจสอบว่า Name ของตัวที่ สร้าง Account คือ										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TCB01	เมื่อทดสอบแล้วพบว่า Name ไม่ถูกต้อง	Account("bob", "001", 1000)	Name = bob	Name = bob	ผ่าน					

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB007			ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์สุวรรณ ดีกรี							
ฟังก์ชันที่ทดสอบ: <code>setName(String)</code> ใน <code>Account.java</code>			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องมี Object Account ก่อนจะทดสอบต่อไป และต้องตรงตามเงื่อนไขข้างบน										
เงื่อนไขหลังทดสอบ: ชื่อของลูกค้าจะถูกเปลี่ยนเป็นชื่อที่กำหนด										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank และ Account โดย parameter (name, accountNumber, balance) ให้ string, string, double ตามลำดับ 2. ใช้ func <code>setName()</code> 3. ตรวจสอบว่า Name ที่ตั้งใหม่จะถูกตั้งใน Account ใหม่										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TSB01	เปลี่ยนชื่อ ลูกค้าจาก jcop เป็น jay	<code>Account("jay", "001", 1000)</code> <code>Name = jcop</code>	ชื่อของลูกค้าจะถูกเปลี่ยนเป็น Jay <code>Name=jay</code>	ชื่อของลูกค้าจะถูกเปลี่ยนเป็น Jay <code>Name=jay</code>	ผ่าน					

โครงการ: Bank	รหัสสถานการณ์การทดสอบ: TSB008	ผู้เขียนสถานการณ์การทดสอบ: นางสาวนิตยาลักษณ์ อุ่งรุ้ง				
ฟังก์ชันที่ทดสอบ: get Account Number () ใน Account.java	วันที่เขียน: 10/10/2024					
เงื่อนไขก่อนทดสอบ: ต้องสร้าง Object Account ก่อนจะเรียกทดสอบ						
เงื่อนไขหลังทดสอบ: ต้องมีไฟเขียว หมายความว่า บัญชีมีเงินคงเหลือ						
ขั้นตอนการดำเนินการทดสอบ:	<ol style="list-style-type: none"> 1. สร้าง Object Bank แบบ Account 里面有 (name, accountNumber, balance) เป็น string, string, double ตามลำดับ 2. ให้เรียก func get Account Number () 3. ตรวจสอบว่า ผลลัพธ์ (Account Number) ตรงกับที่ คาดหวังไว้หรือไม่ 					
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TCB01	ตรวจสอบ เลขบัญชี ลากซ้ายซ้ายขวา 9 บิต	Account("bob", "001", 1000)	Account Number = 001	Account Number = 001	ผ่าน	

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB009			ผู้เขียนสถานการณ์การทดสอบ: ឌາຍ ຕະນະພົມ ດົງງານ							
ฟังก์ชันที่ทดสอบ: set Account Number (String accountNumber) ใน Account.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ຕັ້ງສູ່ Object Account ກຳນົດຈະຕົກຕ່າງ ແລະຕັ້ງການເປັນໄລຍະບົດໃໝ່										
เงื่อนไขหลังทดสอบ: ເບີຍຸ້ນຈຳກັດ ປັບປຸງໃດວ່າການຄົກລົງ										
ขั้นตอนการดำเนินการทดสอบ: 1. ລັບ Object Bank ໃຊ້ Account constructor (name, accountNumber, balance) ຢູ່ string, string, double ທາງໆ 2. ໂທຣີ func set Account Number () 3. ແກ້ວຂອງການໂຫຼມຢູ່ (Account Number) ຕໍ່ການໃບ ດາວກົນ ບໍ່ມີກຳນົດຈະຕົກຕ່າງໃໝ່										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TCB01	ເນັບຂໍ້ມູນເຫັນຈະຕົກຕ່າງໃໝ່	Account("bob", "001", 1000) Account Number = 002	Account Number = 002	Account Number = 002	ผ่าน					

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB010			ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์สุวรรณ ดีกรี							
ฟังก์ชันที่ทดสอบ: getBalance() ใน Account.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องสร้าง Object Account กับชื่อ bob คงเหลือ และจำนวนเงินคงเหลือมากกว่า เท่ากับ 0										
เงื่อนไขหลังทดสอบ: ต้องดูแล้วคงเหลือเท่ากับ 0										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank และ Account โดย parameter (name, accountNumber, balance) ให้ stirng, string, double ตามลำดับ 2. ให้ invoke func getBalance() 3. ตรวจสอบ ว่างานดูแลคงเหลือ (Balance) ของ bob เป็น บวกตัวที่ตั้งไว้ใน object										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TSB01	ตรวจสอบจำนวนเงินคงเหลือจากบัญชี bob	Account("bob", "001", 1000) Balance = 1000.0	Balance = 1000.0	Balance = 1000.0	ผ่าน					

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB011			ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์สุวรรณ ดีกรี							
ฟังก์ชันที่ทดสอบ: setBalance() ใน Account.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องมี Object Account กับ ข้อมูลตามต่อไป และ จำนวนเงินคงเหลือ มากกว่า เท่ากับ 0										
เงื่อนไขหลังทดสอบ: ต้องสามารถ แปลงยอดคงเหลือ ได้ถูกต้อง										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank และ Account โดย parameter (name, accountNumber, balance) ให้ stirng, string, double ตามลำดับ 2. ให้ invoke func setBalance() 3. ตรวจสอบ ว่า ยอดคงเหลือ (Balance) 多了เท่าไหร่ เพื่อเช็ค ให้ถูกต้อง										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TSB01	จำนวนคงเหลือ 0 多了 ไป 10000 คงเหลือ 10000 จำนวนคงเหลือ 0	Account("bob", "001", 1000) Balance = 2000.0	Balance = 2000.0	Balance = 2000.0	ผ่าน					

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB012			ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์สุวิทย์ ดีกรี							
ฟังก์ชันที่ทดสอบ: deposit(double amount) ใน Account.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องมี Object Account กับ ข้อมูลตามต่อไป และ amount ต้องมากกว่า 0 แต่ไม่เกิน 1000										
เงื่อนไขหลังทดสอบ: ยอดคงเหลือ (Balance) ต้องมากกว่าเดิม จำนวน amount Balance เดิม + amount ก็เท่ากัน										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank แบบ Account 里面有 name, accountNumber, balance ใหม่ stirng, string, double ตามลำดับ 2. ให้เรียก func deposit() 3. ตรวจสอบ ว่า ยอดคงเหลือ (Balance) ยังคงมากเท่าเดิม										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TCB01	ฝากเงิน จำนวน มากกว่า 0 แต่ไม่ถึง 1000 ก็ได้	Account("bob", "001", 1000) amount = 500.0	Balance = 1500.0	Balance = 1500.0	ผ่าน					

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB013			ผู้เขียนสถานการณ์การทดสอบ: พญ วงศ์สุวิทย์ ดีกรี							
ฟังก์ชันที่ทดสอบ: withdraw(double amount) ใน Account.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ต้องมี Object Account กับ ข้อมูลดังนี้ balance = 1000 และ amount ต้องมากกว่า 0 และไม่เกิน balance ที่มี										
เงื่อนไขหลังทดสอบ: ยอดคงเหลือ (Balance) ต้องหักไปตามจำนวนที่เดิน Balance เหลือ = amount ก็จะถูก										
ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Object Bank แบบ Account 里面有 name, accountNumber, balance ใช้ string, string, double 作為參數 2. 呼叫 func withdraw() 3. ตรวจสอบ ว่า ยอดคงเหลือ (Balance) หลังจากลดลงเท่าไร										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TCB01	ถอนเงิน จำนวน 500.0 ที่เท่ากับ 0 ไม่สามารถลดยอดคงเหลือ ให้เป็นตัวต่ำกว่า 0	Account("bob", "001", 1000) amount = 500.0	Balance = 500.0	Balance = 500.0	ผ่าน					

โครงการ: Bank										
รหัสสถานการณ์การทดสอบ: TSB014			ผู้เขียนสถานการณ์การทดสอบ: ឌາຍ ໂວງຈົນທິພອນ ອຸ້ງກະຕົງ							
ฟังก์ชันที่ทดสอบ: getAccountInfo() ใน Account.java			วันที่เขียน: 10/10/2024							
เงื่อนไขก่อนทดสอบ: ຕັ້ງສໍາ Object Account ກຳນົດຂໍ້ຕາມຕົວທີ່ ຕັ້ງຂັ້ນຫຼຸດ										
เงื่อนไขหลังทดสอบ: ຕັ້ງສໍາຄັກ (name, accountNumber, balance) ໃນຖືກຕາມ										
ขั้นตอนการดำเนินการทดสอบ: 1. ລັບ Object Bank ໃຊ້ Account ບົນລຸກຄ້າ (name, accountNumber, balance) ມີ stirng, string, double ລາຍກຳດັ່ງ 2. ໄກສອນ func getAccountInfo() 3. ເກມລວມວ່າຂໍ້ມູນປຶກຕົວ Account ກ່າວຂອງບໍ່ໄດ້										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TCB01	ສ້າງກຽມກົງລົງລົງໃຫ້ອຸ້ງກະຕົງ	Account("bob", "001", 1000)	Name: bob AccountNumber : 001 Balance : 1000.0	Name: bob AccountNumber : 001 Balance : 1000.0	ผ่าน					

โครงการ: Sorting

รหัสสถานการณ์การทดสอบ: TS001

พัฒนาที่ทดสอบ: bubble Sort และ Print Array

เงื่อนไขก่อนทดสอบ: Code ใช้ฟังก์ชัน bubbleSort, Print Array ให้เป็นแบบดูดต่อ
ส่วนร่วม $n(\text{size})$ และ บรรจุภัณฑ์ในรูปแบบ

เงื่อนไขหลังทดสอบ: ฟังก์ชัน Print Array จะแสดงผลลัพธ์

ขั้นตอนการดำเนินการทดสอบ: 1. สร้าง Array ให้ $n(\text{size})$

4. ลงภาษาทำการทดสอบที่ได้พิมพ์

2. นำเข้า func bubbleSort

3. นำเข้า func Print Array

รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TS01	สร้าง Array ที่เรียงลำดับไม่ถูกต้อง ให้ n เป็นค่าจำนวน Array	$\text{arr}[] = [1, 2, 3, 4, 5]$ $n = 5$	1 2 3 4 5	1 2 3 4 5	ผ่าน	ใช้ swap
TS02	สร้าง Array ที่เรียงลำดับจากมากไปน้อย ให้ n เป็นค่าจำนวน Array	$\text{arr}[] = [5, 4, 3, 2, 1]$ $n = 5$	1 2 3 4 5	1 2 3 4 5	ผ่าน	ใช้ swap
TS03	สร้าง Array ที่มีซ้ำเดียว ให้ n เป็นค่าจำนวน Array	$\text{arr}[] = [5, 1, 3, 2, 4]$ $n = 5$	1 2 3 4 5	1 2 3 4 5	ผ่าน	ใช้ swap
TS04	สร้าง Array ที่มีตัวเล็กลบ ให้ n เป็นค่าจำนวน Array	$\text{arr}[] = [-1, 2, 3, -4, -5]$ $n = 5$	-5 -4 -1 2 3	-5 -4 -1 2 3	ผ่าน	ใช้ swap
TS05	สร้าง Array ที่มีขนาด 1 เท่านั้น ให้ n เป็นค่าจำนวน Array	$\text{arr}[] = [1]$ $n = 1$	1	1	ผ่าน	Print ค่าเดียว
TS06	สร้าง Array ที่มีค่า 0 ให้ n เป็นค่าจำนวน Array	$\text{arr}[] = []$ $n = 0$			ผ่าน	ใช้ swap

TS07	ສ່ວນ Array ທີ່ສັບຕິດ ຖະການ ໄລຍະ ທີ່ສ່ວນກາງລໍາງວຸນ Array	<code>arr[] = [1,2,3,4,5] n = 6</code>	Code error	Code error	ຜົກມ	ໃຫ້ ArrayIndexOutOfBoundsException
TS08	ສ່ວນ Array ທີ່ສັບຕິດ ຖະການ ໄລຍະ ທີ່ມີຄວາມກາງລໍາງວຸນ Array	<code>arr[] = [1,2,3,4,5] n = 4</code>	1 2 3 4	Code error	ໄສເພີ້ມ	ໃຫ້ ArrayIndexOutOfBoundsException

โครงการ: ทดสอบ Web UI						
รหัสสถานการณ์การทดสอบ: SE0001		ผู้เขียนสถานการณ์การทดสอบ: นางนิชัชินทร์ กุลกาฬสุก				
พังก์ชันที่ทดสอบ: จัดการข้อมูลผู้ใช้งาน		วันที่เขียน: ๐๗/๑๐/๒๕๖๗				
เงื่อนไขก่อนทดสอบ: ผู้ใช้กับมีบัญชีอยู่ในระบบ						
เงื่อนไขหลังทดสอบ: ผู้ใช้เข้าสู่ระบบได้ และมีข้อมูลบันทึก						
ขั้นตอนการดำเนินการทดสอบ: 1. เปิดเบราว์เซอร์ 2. กดปุ่ม Customer Login 3. เลือกชื่อ 4. กดปุ่ม login						
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
SE01	เปิดเบราว์เซอร์, กดปุ่ม Customer Login, เลือกชื่อ login, กดปุ่ม login	ชื่อ: Hermoine Granger	ผู้ใช้ลงทะเบียนสำเร็จ	ผู้ใช้ลงทะเบียนสำเร็จ	ผ่าน	
SE02	เปิดเบราว์เซอร์, กดปุ่ม Customer Login, ไม่เลือกชื่อ login, กดปุ่ม login	ชื่อ: Null	ผู้ใช้ลงทะเบียนไม่สำเร็จ	ผู้ใช้ลงทะเบียนไม่สำเร็จ	ผ่าน	

โครงการ: ห้อง Web UI						
รหัสสถานการณ์การทดสอบ: SE0002	ผู้เขียนสถานการณ์การทดสอบ: นางสาวนิษฐา กมลเดชานุกา					
พิจารณาที่ทดสอบ: Transaction	วันที่เขียน: 07/10/2569					
เงื่อนไขก่อนทดสอบ: บัญชีเดือนตุลาคม ปี 2569						
เงื่อนไขหลังทดสอบ: บัญชีเดือนตุลาคม ปี 2569 รายการบันทึกได้						
ขั้นตอนการดำเนินการทดสอบ: 1. ลงทะเบียน transaction 2. กดปุ่ม Back 3. หักบัญชี 4. เรียก transaction ใหม่ 5. กดปุ่ม Reset						
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
SE01	หักบัญชี transaction ตัวแรก	-	บัญชีปรับตัวรุกราน	บัญชีปรับตัวรุกราน	ผ่าน	
SE02	หักบัญชี transaction	-	บัญชีปรับตัวรุกราน	บัญชีปรับตัวรุกราน	ผ่าน	
SE03	หักบัญชี, กดปุ่ม transaction	บัญชี = ลบหนี้ 500	หักบัญชี ลบหนี้ 500 และหักบัญชีใน col	หักบัญชี ลบหนี้ 500	ผ่าน	
SE04	หักบัญชี, กดปุ่ม transaction no reset	บัญชี = ฝากเงิน 1000	หักบัญชี ลบหนี้ 1000 กดปุ่ม reset หักบัญชีหาย	หักบัญชี ลบหนี้ 1000 หักบัญชี กดปุ่ม reset หักบัญชีหาย	ผ่าน	
SE05	หักบัญชี, หักบัญชี transaction กด back	บัญชี = ฝากเงิน 1	หักบัญชี หักบัญชี กดปุ่ม back แล้วหักบัญชียังคงอยู่	หักบัญชี หักบัญชี กดปุ่ม back แล้วหักบัญชีหาย	ไม่ผ่าน	

โครงการ: กบว๐๘ Web VI						
รหัสสถานการณ์การทดสอบ: SE 0003	ผู้เขียนสถานการณ์การทดสอบ: แหนมหูนิพัทธ์ กะตองกุณฑ์					
ฟังก์ชันที่ทดสอบ: Deposit	วันที่เขียน: 07/10/2560					
เงื่อนไขก่อนทดสอบ: ผู้ใช้รับรหัส 0003 และยังไม่ได้ทำการ login						
เงื่อนไขหลังทดสอบ: ผู้ใช้รับรหัส 0003 และยังไม่ได้ทำการ login เนื่องจากยังไม่ได้ทำการ deposit						
ขั้นตอนการดำเนินการทดสอบ: 1. กดปุ่ม Deposit 2. ล็อกอินและ 3. กดปุ่ม Deposit						
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
SE 01	กด Deposit , ล็อกอินแล้ว, กดปุ่ม Deposit	เงิน = 1,111	เงินคงเหลือ 1,111	เงินจำนวน 1,111 เงินที่เหลือ 1,111	ผ่าน	
SE 02	กด Deposit , ล็อกอินแล้ว, กดปุ่ม Deposit	เงิน = -1	เงินคงเหลือ -1	เงินคงเหลือ -1	ผ่าน	
SE 03	กด Deposit , ล็อกอินแล้ว, กดปุ่ม Deposit	เงิน = 0	เงินคงเหลือ 0	เงินคงเหลือ 0	ผ่าน	
SE 04	กด Deposit , ล็อกอินแล้ว, กดปุ่ม Deposit	เงิน = null	เงินคงเหลือ null	เงินคงเหลือ null	ผ่าน	
SE 05	กด Deposit , ล็อกอินแล้ว, กดปุ่ม Deposit	เงิน = ๙๙๙	เงินคงเหลือ ๙๙๙	ผ่าน	ผ่าน	
SE 06	กดปุ่ม Deposit , ล็อกอินแล้ว, กดปุ่ม Deposit	เงิน = ๔๗๓๔๔	เงินคงเหลือ ๔๗๓๔๔	ผ่าน	ผ่าน	

โครงการ: ทดสอบ Web UI										
รหัสสถานการณ์การทดสอบ: SE 0004			ผู้เขียนสถานการณ์การทดสอบ: นางสาวนิรันดร์ ภานุเดชกุล							
พิจารณาที่ทดสอบ: withdraw			วันที่เขียน: 07/10/2567							
เงื่อนไขก่อนทดสอบ: รู้จักสังคมออนไลน์ มีบัญชีในมือถือ										
เงื่อนไขหลังทดสอบ: รู้จักสังคมออนไลน์ เรียนรู้เกี่ยวกับ withdraw										
ขั้นตอนการดำเนินการทดสอบ: 1. กด withdraw 2. รอดูหน้าเงิน 3. กดปุ่ม withdraw										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
SE01	ไม่ withdraw, รู้จักสังคมออนไลน์ ไม่ withdraw	$balance = 2000$	คงเหลือเงิน 2000	คงเหลือเงิน 2000	ผ่าน					
SE02	ไม่ withdraw, รู้จักสังคมออนไลน์ ไม่ withdraw	$balance = 0$	คงเหลือเงิน 0	คงเหลือเงิน 0	ผ่าน					
SE03	ไม่ withdraw, รู้จักสังคมออนไลน์ ไม่ withdraw	$balance = -10$	คงเหลือเงิน 0	คงเหลือเงิน 0	ผ่าน					
SE04	กด withdraw, รู้จักสังคมออนไลน์ ไม่ withdraw	$balance = null$	แสดงหน้าจอแจ้งเตือน	แสดงหน้าจอแจ้งเตือน	ผ่าน					
SE05	ไม่ withdraw, รู้จักสังคมออนไลน์ ไม่ withdraw	$balance = 200$	แสดงหน้าจอแจ้งเตือน	แสดงหน้าจอแจ้งเตือน	ผ่าน					
SE06	กด withdraw, รู้จักสังคมออนไลน์ ไม่ withdraw เงินเพียงพอ, กดปุ่ม withdraw	$balance = 300$	คงเหลือเงิน 0	คงเหลือเงิน 0	ผ่าน					

โครงการ: หตส๐๖ Web UI											
รหัสสถานการณ์การทดสอบ: SE0005		ผู้เขียนสถานการณ์การทดสอบ: นางสาวปัจฉิม์ ภูมิธรรมกุล									
พิมพ์ชื่อที่ทดสอบ: logout		วันที่เขียน: ๐๙/๑๐/๒๕๖๗									
เงื่อนไขก่อนทดสอบ: ผู้ใช้งานต้องเข้าระบบ											
เงื่อนไขหลังทดสอบ: ผู้ใช้งานต้องออกจากระบบ ไปยังหน้าจอจัดการ											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
SE01	ทดสอบ logout	-	ระบบ รีเซ็ต กลับไปยังหน้าจอจัดการ	ระบบ รีเซ็ต กลับไปยังหน้าจอจัดการ	ผ่าน						

โครงการ: ทดสอบ Web UI											
รหัสสถานการณ์การทดสอบ: SE0006		ผู้เขียนสถานการณ์การทดสอบ: นางนิรุ๊บดี กุลจตุกรา									
พิมพ์ชื่อที่ทดสอบ: Add Customer		วันที่เขียน: ๐๗ / ๑๐ / ๒๕๖๗									
เงื่อนไขก่อนทดสอบ: 1. ต้อง login Bank Manager											
เงื่อนไขหลังทดสอบ: มีรายชื่อผู้ใช้งานเพิ่มขึ้น											
ขั้นตอนการดำเนินการทดสอบ: 1. ล็อกอิน - หน้าจอ login 2. ผู้ดูแลระบบเข้าสู่ระบบ 3. คลิก Add Customer											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
SE01	9 ล็อกอิน - หน้าจอ login , รหัสไปรษณีย์ ๘๘๐ Add Customer	ชื่อ = Nattanun นามสกุล = Umoldejitta รหัสไปรษณีย์ = ๒๑๑๓๐	ระบุชื่อที่ ๙๘๐ ไปรษณีย์ ได้ลง id	ระบุชื่อถูกเพิ่มแล้ว ได้ลง id	ผ่าน						
SE02	9 ล็อกอิน - หน้าจอ login , รหัสไปรษณีย์ ๘๘๐ Add Customer	ชื่อ = ๑๒ นามสกุล = ๓๔ รหัสไปรษณีย์ = ๘๘๘๘๘๘	เข้าสู่ระบบ	ระบุชื่อถูกเพิ่มแล้ว ได้ลง id	ผ่าน						
SE03	๙ ล็อกอิน - หน้าจอ login , รหัสไปรษณีย์ ๘๘๐ Add Customer	ชื่อ = null นามสกุล = null รหัสไปรษณีย์ = null	แจ้งเตือนว่า ๙ ล็อกอิน	แจ้งเตือนว่า ๙ ล็อกอิน	ผ่าน						

โครงการ: ทดสอบ Web UI						
รหัสสถานการณ์การทดสอบ: SE0007			ผู้เขียนสถานการณ์การทดสอบ: นางสาวนิษฐ์ กมลเดชานา			
พิจารณาที่ทดสอบ: Open Account			วันที่เขียน: 07/10/2569			
เงื่อนไขก่อนทดสอบ: login Bank Manager						
เงื่อนไขหลังทดสอบ: 1. เลือกบัญชี 2. เลือกสกุลเงิน 3. กด Process						
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
SE01	เลือกบัญชี, เลือกสกุลเงิน กด Process	บัญชี = Hermoine Granger สกุลเงิน = Dollar	เข้าสู่หน้าจอชี้ไปที่	เข้าสู่หน้าจอชี้ไปที่	ผ่าน	
SE02	ไม่เลือกบัญชี, เลือกสกุลเงิน กด Process	บัญชี = null สกุลเงิน = null	เข้าสู่หน้าจอชี้ไปที่	เข้าสู่หน้าจอชี้ไปที่	ผ่าน	
SE03	เลือกบัญชี, ไม่เลือกสกุลเงิน กด Process	บัญชี = Harry Potter สกุลเงิน = null	เข้าสู่หน้าจอชี้ไปที่	เข้าสู่หน้าจอชี้ไปที่	ผ่าน	
SE04	ไม่เลือกบัญชี, เลือกสกุลเงิน กด Process	บัญชี = null สกุลเงิน = Dollar	เข้าสู่หน้าจอชี้ไปที่	เข้าสู่หน้าจอชี้ไปที่	ผ่าน	

โครงการ: กบกบ Web VI						
รหัสสถานการณ์การทดสอบ: SE 0008		ผู้เขียนสถานการณ์การทดสอบ: แหนมณีนันท์ กบกบ				
ฟังก์ชันที่ทดสอบ: Customers		วันที่เขียน: 07/10/2560				
เงื่อนไขก่อนทดสอบ: login Bank Manager						
เงื่อนไขหลังทดสอบ: ลบบัญชี ลับหน้าจูดีไล						
ขั้นตอนการดำเนินการทดสอบ: 1. ตรวจสอบบัญชี 2. ลบ Delete						
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
SE01	ตรวจสอบบัญชี, ลบ Delete	ชื่อ = Hermoine	บัญชี Hermoine ถูกพบ	บัญชี Hermoine ถูกพบ	ผ่าน	
SE02	ลบบัญชี, ลบ Delete	ชื่อ=null	ไม่พบบัญชีชื่อ ปัญชี	ลบบัญชีที่ ลบ Delete 75	ผิดพลาด	
SE03	ตรวจสอบบัญชี, ลบ Delete	ชื่อ = Potter	บัญชี Harry ถูกพบ	บัญชี Harry ถูกพบ	ผ่าน	
SE04	ลบบัญชี, ลบ Delete	ลบบัญชี = 1002	ลบบัญชี 1002 ถูกพบ	ลบบัญชี 1002 ถูกพบ	ผ่าน	

โครงการ: Popdung Selenium Test						
รหัสสถานการณ์การทดสอบ: PST 001		ผู้เขียนสถานการณ์การทดสอบ: นางสาวนุสบา ลังกรีบ				
พึงชั้นที่ทดสอบ: กอกเพิ่มค่าท่าน		วันที่เขียน: ๑๖/๑๐/๒๕๖๗				
เงื่อนไขทดสอบ: ต้องไฟฟ้ามีเรื่อง ผ่อนไฟฟ้าคงเดิม: ดูหน้าป้องกัน						
ข้อตอนภาษาอังกฤษ: 1. เนื้อร่อง 2. ทดสอบซึ่ง						
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
PST01	เนื้อร่อง, กอกลักษณะ	ค่าปกติ	ดูหน้าป้องกัน	ดูหน้าป้องกันที่กล่าวไว้	ผ่าน	
PST02	เนื้อร่อง, กอกลักษณะ	คลิก ปาน	กະหนาแน่น	ดูหน้าป้องกันที่กล่าวไว้ ผ่าน command	ผ่าน	

โครงการ: ทดสอบ API ของเว็บไซต์ https://fakestoreapi.com												
รหัสสถานการณ์การทดสอบ: Test_api_get	ผู้เขียนสถานการณ์การทดสอบ: นายวัชรากร รัศมีดิษฐ์											
ฟังก์ชันที่ทดสอบ: get a single product	วันที่เขียน: 15/10/2024											
เงื่อนไขก่อนทดสอบ : เว็บไซค์สามารถใช้งานได้ และ คุณมีการใช้งานถูกต้อง												
เงื่อนไขหลังการทดสอบ : api ควร return product ตาม id ที่กำหนด												
ขั้นตอนการทดสอบ :												
<ol style="list-style-type: none"> 1. เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น GET 2. ใส่ URL: https://fakestoreapi.com/products/{productID} 3. คลิก send 												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TA01	Request ใหม่ โดยเลือก Method เป็น GET และใส่ url และจึงกด send	url = https://fakestoreapi.com/products/1	แสดงรายละเอียดของ product ที่ 1	{ "id": 1, "title": "Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops", "price": 109.95, "description": "Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday", "category": "men's clothing", "image": " https://fakestoreapi.com/img/81fPKd-2AYL.AC_SL1500.jpg ", "rating": { "count": 120, "rate": 4.6 }	ผ่าน	แสดง product ที่ 1						

				<pre> "rate": 3.9, "count": 120 } } </pre>		
TA02	Request ใหม่ โดยเลือก Method เป็น GET และใส่ url และวิจัยกด send	url = https://fakestoreapi.com/products/0	ไม่แสดงรายละเอียดของ product ที่ 0	-	ผ่าน	ไม่มี product ที่ 0
TA03	Request ใหม่ โดยเลือก Method เป็น GET และใส่ url และวิจัยกด send	url = https://fakestoreapi.com/products/-1	ไม่แสดงรายละเอียดของ product ที่ -1	-	ผ่าน	ไม่มี product ที่ -1
TA04	Request ใหม่ โดยเลือก Method เป็น GET และใส่ url และวิจัยกด send	url = https://fakestoreapi.com/products/a	ไม่แสดงรายละเอียดของ product ที่ a	-	ผ่าน	ไม่มี product ที่ a
TA05	Request ใหม่ โดยเลือก Method เป็น GET และใส่ url และวิจัยกด send	url = https://fakestoreapi.com/products/21	ไม่แสดงรายละเอียดของ product ที่ 21	-	ผ่าน	ไม่มี product ที่ 21

TA06	<p>Request ใหม่ โดยเลือก Method เป็น GET และใส่ url และจึงกด send</p>	<pre>url = https://fakestoreapi.com/products/20</pre>	<p>แสดงรายละเอียดของ product ที่ 20</p>	<pre>{ "id": 20, "title": "Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops", "price": 109.95, "description": "Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday", "category": "men's clothing", "image": "https://fakestoreapi.com/img/81fPKd-2AYL._AC_SL1500_.jpg", "rating": { "rate": 3.9, "count": 120 } }</pre>	<p>ผ่าน</p>	<p>แสดง product ที่ 20</p>
------	---	---	---	--	-------------	----------------------------

--	--	--	--	--	--	--

โครงการ: ทดสอบ API ของเว็บไซต์ https://fakestoreapi.com										
รหัสสถานการณ์การทดสอบ: Test_api_pose	ผู้เขียนสถานการณ์การทดสอบ: นายวัชรากร รัคมีดิษฐ์									
ฟังก์ชันที่ทดสอบ: add a new product	วันที่เขียน: 15/10/2024									
เงื่อนไขก่อนทดสอบ : เว็บไซต์สามารถใช้งานได้ และ คุณมีการใช้งานถูกต้อง										
เงื่อนไขหลังการทดสอบ : สามารถเพิ่ม product ได้										
ขั้นตอนการทดสอบ :										
<ol style="list-style-type: none"> 1. เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น POSE 2. ใส่ URL: https://fakestoreapi.com/products/ 3. ไปที่แท็บ Body และเลือก raw และ JSON (จาก dropdown menu) 4. ใส่ข้อมูล JSON ที่ต้องการเพิ่มในส่วน Body ของ Request: 5. คลิกที่ Send 										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				

TB01	เปิด Postman และสร้าง Request ใหม่โดยเลือก Method เป็น POST ใส่ url และ ใส่ข้อมูล JSON ที่ต้องการเพิ่มในส่วน body	url = https://fakestoreapi.com/products/ "id" : 21, "title": "test product", "price": 13.5, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic"	สามารถเพิ่ม product ได้สำเร็จ	{ "id": 21, "title": "test product", "price": 13.5, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic" }	ผ่าน	เพิ่ม product ที่ 21
TB02	เปิด Postman และสร้าง Request ใหม่โดยเลือก Method เป็น POST ใส่ url และ ใส่ข้อมูล JSON ที่ต้องการเพิ่มในส่วน body	url = https://fakestoreapi.com/products/ "id" : 22, "title": "test product", "price": 13.5, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic"	สามารถเพิ่ม product ได้สำเร็จ	{ "id": 21, "title": "test product2", "price": 5.5, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic" }	ไม่ผ่าน	ต้องการเพิ่ม product 22 แต่เป็นการเพิ่ม product 21 แทน

โครงการ: ทดสอบ API ของเว็บไซต์ https://fakestoreapi.com										
รหัสสถานการณ์การทดสอบ: Test_api_put					ผู้เขียนสถานการณ์การทดสอบ: นายวัชรากร รัศมีดิษฐ์					
พึงชันที่ทดสอบ: update a product					วันที่เขียน: 15/10/2024					
เงื่อนไขก่อนทดสอบ : เว็บไซค์สามารถใช้งานได้ และ คุณมีการใช้งานถูกต้อง										
เงื่อนไขหลังการทดสอบ : สามารถแก้ไข product ได้										
ขั้นตอนการทดสอบ :										
<ol style="list-style-type: none"> 1. เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น PUT 2. ใส่ URL: https://fakestoreapi.com/products/{productId} 3. ไปที่แท็บ Body และเลือก raw และ JSON (จาก dropdown menu) 4. ใส่ข้อมูล JSON ที่ต้องการอัปเดตลงในส่วน Body ของ Request: 5. คลิกที่ Send 										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				

TC01	เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น PUT ใส่ url และ ใส่ข้อมูล JSON ที่ต้องการ เพิ่มในส่วน body และจึงกด send	url = https://fakestoreapi.com/products/1 "title": "testupdateproduct1", "price": 7, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic"	แก้ไข product ที่ 1 ได้สำเร็จ	{ "title": "testupdateproduct1", "price": 7, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic" }	ผ่าน	แก้ไข product 1
TC02	เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น PUT ใส่ url และ ใส่ข้อมูล JSON ที่ต้องการ เพิ่มในส่วน body และจึงกด send	url = https://fakestoreapi.com/products/-1 "title": "testupdateproduct1", "price": 7, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic"	แก้ไข product ที่ -1 ได้สำเร็จ	{ "title": "testupdateproduct2", "price": 13.5, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic" }	ผ่าน	แก้ไข product ที่ -1
TC03	เปิด Postman และสร้าง Request ใหม่	url = https://fakestoreapi.com/products/a "title": "testupdateproduct1",	แก้ไข product ที่ a ได้สำเร็จ	{ "status": "error",	ไม่ผ่าน	ไม่สามารถกรอก ตัวอักษรใน productid ได้

โดยเลือก Method เป็น PUT ใส่ url และ ใส่ข้อมูล JSON ที่ต้องการ เพิ่มในส่วน body แล้วจึงกด send	"price": 7, "description": "lorem ipsum set", "image": "https://i.pravatar.cc", "category": "electronic"		"message": "something went wrong! check your sent data" }		
--	---	--	---	--	--

โครงการ: ทดสอบ API ของเว็บไซต์ https://fakestoreapi.com									
รหัสสถานการณ์ทดสอบ: Test_api_delete				ผู้เขียนสถานการณ์ทดสอบ: นายวัชรากร รัศมีดิษฐ์					
พิจารณาที่ทดสอบ: delete a product				วันที่เขียน: 15/10/2024					
เงื่อนไขก่อนทดสอบ : เว็บไซค์สามารถใช้งานได้ และ คุณมีการใช้งานถูกต้อง									
เงื่อนไขหลังการทดสอบ : สามารถลบ product ได้									
ขั้นตอนการทดสอบ :									
<ol style="list-style-type: none"> 1. เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น Delete 2. ใส่ URL: https://fakestoreapi.com/products/{productId} 3. คลิกที่ Send 									
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ			
TD01	เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น DELETE ใส่ url และวิ่งกด send	url = https://fakestoreapi.com/products/1	สามารถลบ product ที่ 1 ได้สำเร็จ	{ "id": 1, "title": "Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops", "price": 109.95, "description": "Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday", "category": "men's clothing", "image": "https://fakestoreapi.com/img/81fP Kd-2AYL._AC_SL1500_.jpg",	ผ่าน	ลบ product ที่ 1			

				<pre>"rating": { "rate": 3.9, "count": 120 }</pre>		
TD02	เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น DELETE ใส่ url และวิ่งกด send	url = https://fakestoreapi.com/products/21	แสดงช่องว่าง	NULL	ผ่าน	ไม่สามารถลบ product ที่ไม่มีจริงได้
TB03	เปิด Postman และสร้าง Request ใหม่ โดยเลือก Method เป็น DELETE ใส่ url และวิ่งกด send	url = https://fakestoreapi.com/products/a	แสดงช่องว่าง	<pre>{ "status": "error", "message": "cart id should be provided" }</pre>	ไม่ผ่าน	ไม่สามารถกรอก ตัวอักษรใน productid ได้

โครงการ: Pantip Performance testing						
รหัสสถานการณ์ทดสอบ: PT	ผู้เขียนสถานการณ์การทดสอบ: ชุต นุรักษ์					
ฟังก์ชันที่ทดสอบ: New result, Summary, Aggregate	วันที่เขียน: ๑๖/๑๐/๒๐๒๔					
ข้อมูลการทดสอบ: 10 User Sec Load	ผู้ทดสอบ: ชุต นุรักษ์					
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
PT01	10 User 300 นาที Web ทดสอบ	User: 300 Sec: 5 loop: 2	ยกเว้น 300 นาที สามารถเข้าถึงเว็บได้ ยกเว้น 300 นาที	ยกเว้น 300 นาที ไม่สามารถเข้าถึงเว็บได้	ยกเว้น	
PT02	10 User 300 นาที Web ทดสอบ	User: 300 Sec: 5 loop: 1	ยกเว้น 300 นาที สามารถเข้าถึงเว็บได้ ยกเว้น 300 นาที	ยกเว้น 300 นาที ไม่สามารถเข้าถึงเว็บได้	ยกเว้น	
PT03	10 User 10 นาที Web ทดสอบ	User: 10 Sec: 5 loop: 1	ยกเว้น 10 นาที สามารถเข้าถึงเว็บได้ ยกเว้น 10 นาที	ยกเว้น 10 นาที ไม่สามารถเข้าถึงเว็บได้	ยกเว้น	

โครงการ: PG Performance testing							
รหัสสถานการณ์การทดสอบ: PG				ผู้เขียนสถานการณ์การทดสอบ: ชัชวาล พัฒนา			
ฟังก์ชันที่ทดสอบ: View result, Summary, Aggregate				วันที่เขียน: 17/10/20			
หมายเหตุ: กำหนดการทดสอบ ; 9 ชั่วโมง User , Sec , Loop							
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ	
PG01	9 ชั่วโมง User 300 ผู้ใช้งาน Web	User: 300 Sec: 1 Loop: 1	ผู้ใช้งาน 300 คน สามารถเข้า Web ได้ โภต ไม่ล่าช้า	ผู้ใช้งาน 300 คน สามารถเข้า Web ได้ โภต ไม่ล่าช้า	ผู้ใช้งาน		
PG02	9 ชั่วโมง User 20 ผู้ใช้งาน Web	User: 20 Sec: 1 Loop: 1	ผู้ใช้งาน 20 คน สามารถเข้า Web ได้ โภต	ผู้ใช้งาน 20 คน สามารถเข้า Web ได้ โภต	ผู้ใช้งาน		
PG03	9 ชั่วโมง User 10 ผู้ใช้งาน Web	User: 10 Sec: 10 Loop: 1	ผู้ใช้งาน 10 คน สามารถเข้า Web ได้ โภต	ผู้ใช้งาน 10 คน สามารถเข้า Web ได้ โภต	ผู้ใช้งาน		