1. جلب جميع المراجعات من القاعدة وحفظها في المتغير matrix

$query = *mysqli\_query*($conn, "SELECT \* FROM `reviews` ");  
$matrix = [];  
while($row = *mysqli\_fetch\_assoc*($query)){  
 $matrix[$row["user\_id"]][$row["book\_id"]] = $row["rating"];  
}

1. جلب جميع الكتب من القاعدة وحفظها في المتغير books

$books = [];  
$queryBook = *mysqli\_query*($conn, "SELECT `id` FROM `books` ");  
while($row = *mysqli\_fetch\_assoc*($queryBook)){  
 $books[$row['id']] = $row["id"];  
}

1. التعويض في مصفوفة الماتريكس عن أي كتاب غير مقيم بالقيمة صفر

foreach ($matrix as $k=>$v){  
 foreach ($books as $book){  
 if (!*array\_key\_exists*($book,$v)){  
 $matrix[$k][$book]=0;  
 }  
 }  
}

1. جلب مسافة الكوساين بين المستخدمين

$numeatore = 0;  
$denItem = 0;  
$denOtherItem = 0;  
foreach ($matrix[$item] as $book\_id=>$rate){  
 $numeatore += $rate \* $matrix[$otherItem][$book\_id];  
 $denItem += *pow*($rate,2); *// u1* $denOtherItem += *pow*($matrix[$otherItem][$book\_id],2); *// other item*}  
return $numeatore / ( *sqrt*($denItem) \* *sqrt*($denOtherItem) );

1. جلب التقييم المتوقع للمستخدم على الكتب التي لم يقيمها
2. إيجاد مجاميع البسط والمقام

$numratore = []; *// ["b1" => 0, "b2" =>0 ,.....bn =>0]*$denomiratore = [];  
foreach ($matrix as $otherItem=>$itemValue){  
 if ($otherItem != $item){  
 $sim = cosinSim($matrix, $item, $otherItem);  
 foreach($matrix[$otherItem] as $book=>$rating){  
 if ($matrix[$item][$book] == 0){  
 if (!*array\_key\_exists*($book, $numratore)){  
 $numratore[$book] = 0;  
 }  
 $numratore[$book] += $sim \* $rating;  
 if(!*array\_key\_exists*($book, $denomiratore)){  
 $denomiratore[$book] = 0;  
 }  
 $denomiratore[$book] += $sim;  
 }  
 }  
 }  
}

1. تنفيذ عملية قسمة البسط على المقام

foreach($numratore as $key=>$value){  
 $result[$key] = $value/$denomiratore[$key];  
}

1. إعادة ترتيب المصفوفة result من الأعلى تقييم الى الأقل تقييم, ثم إعادة النتائج

*uasort*($result, function($a,$b){  
 if($a==$b) return 0;  
 return ($a>$b) ? -1:1;  
}  
);  
return $result;

1. بعد جلب ترتيب نسب التشابه للكتب التي لم يقرأها المستخدم ووضعها في مصفوفة نعمل حلقة تكرار (for loop) على المصفوفة ونجلب بيانات الكتاب بحسب الـ id لأول 6 عناصر في المصفوفة

$i = 1;  
$reR = [];  
global $conn;   
foreach($rec as $book\_id=>$rating) {  
 if ($i > 6) {  
 break;  
 }  
}

1. نجلب بيانات الكتاب من القاعدة حسب id الكتاب

.  
.  
.  
if($i > 6) {break;}  
$result = *mysqli\_query*($conn, "SELECT `id`, `publisher`, `num\_pages`, `image\_url`, `title`  
 FROM `books` WHERE `id` = $book\_id");

1. نعمل حلقة تكرار while loop على النتائج

while ($row = *mysqli\_fetch\_assoc*($result)){

1. نعمل استعلام وجلب لبيانات المؤلفين والتقييم المتوسط للكتاب والتصنيف الخاصة بالكتاب حسب id الكتاب

*///// authors ///////* $authors = [];  
 $query = *mysqli\_query*($conn,"SELECT `name` FROM `author\_book`  
JOIN `authors` ON `author\_book`.`author\_id` = `authors`.`id`  
WHERE `author\_book`.`book\_id` = ".$row["id"]);  
 while ($rowQuery = *mysqli\_fetch\_assoc*($query)){  
 *array\_push*($authors,$rowQuery["name"]);  
 }  
 $row['authors'] = $authors;  
 */// rating ////////////* $query = *mysqli\_query*($conn,"SELECT AVG(`rating`) AS `rating` FROM `reviews` WHERE `book\_id` = ".$row["id"]);  
 while ($rowQuery = *mysqli\_fetch\_assoc*($query)){  
 $row['rate'] = *round*($rowQuery['rating'],1);  
 }  
 *// --------------------- genres* $query = *mysqli\_query*($conn,"SELECT \* FROM `book\_geners`  
JOIN `geners` ON `book\_geners`.`genres\_id` = `geners`.`id`  
HAVING `book\_id` = ".$row["id"]);  
 $genres = [];  
 while ($rowQuery = *mysqli\_fetch\_assoc*($query)){  
 *array\_push*($genres,$rowQuery["name"]);  
 }  
 $row['genres'] = $genres;

1. قبل انهاء حلقة تكرار foreach نقوم بإضافة الصف للمصفوفة الأب ونقوم بإضافة 1 الى العداد

*array\_push*($reR,$row);  
 } *// end while ($row = mysqli\_fetch\_assoc($result)){* $i++;  
} *// end foreach($rec as $book\_id=>$rating){*

1. إعادة ترتيب النتائج حسب متوسط التقييم

*uasort*($reR, function($a,$b){  
 return $b['rate']<=>$a['rate'];  
}  
);

1. إعادة الكتب

return $reR;