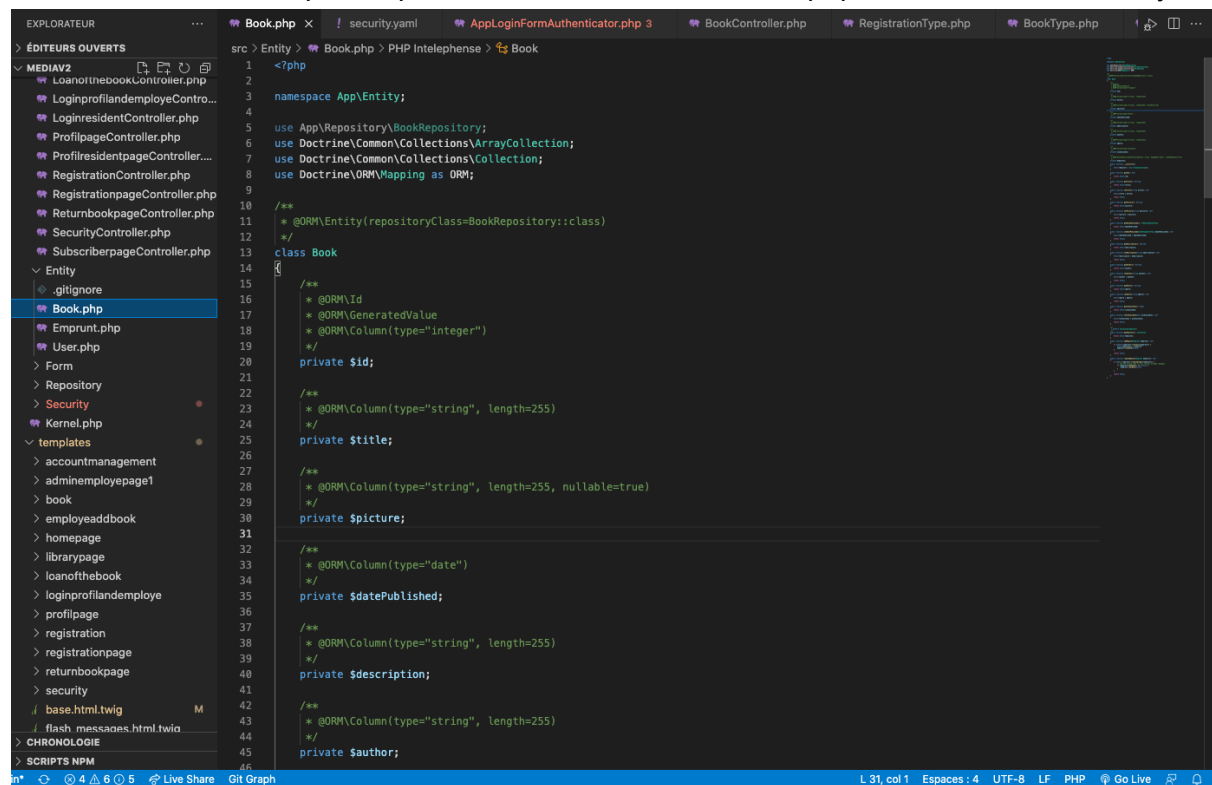


Les fichiers de création et d'alimentation de la base de données :

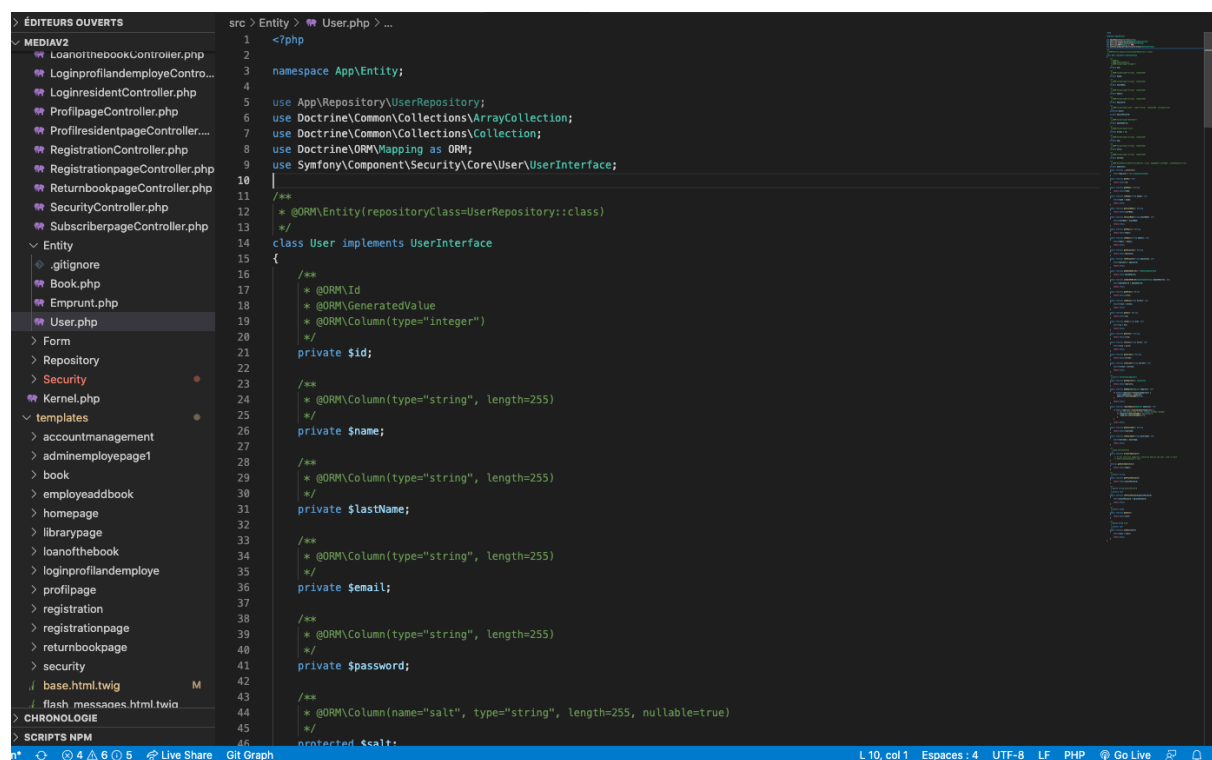
-1 création de la base de la donnée via doctrine (php bin/console doctrine : database : create)

-2 création des tables qui compose ma base de donnée avec php bin/console make:entity



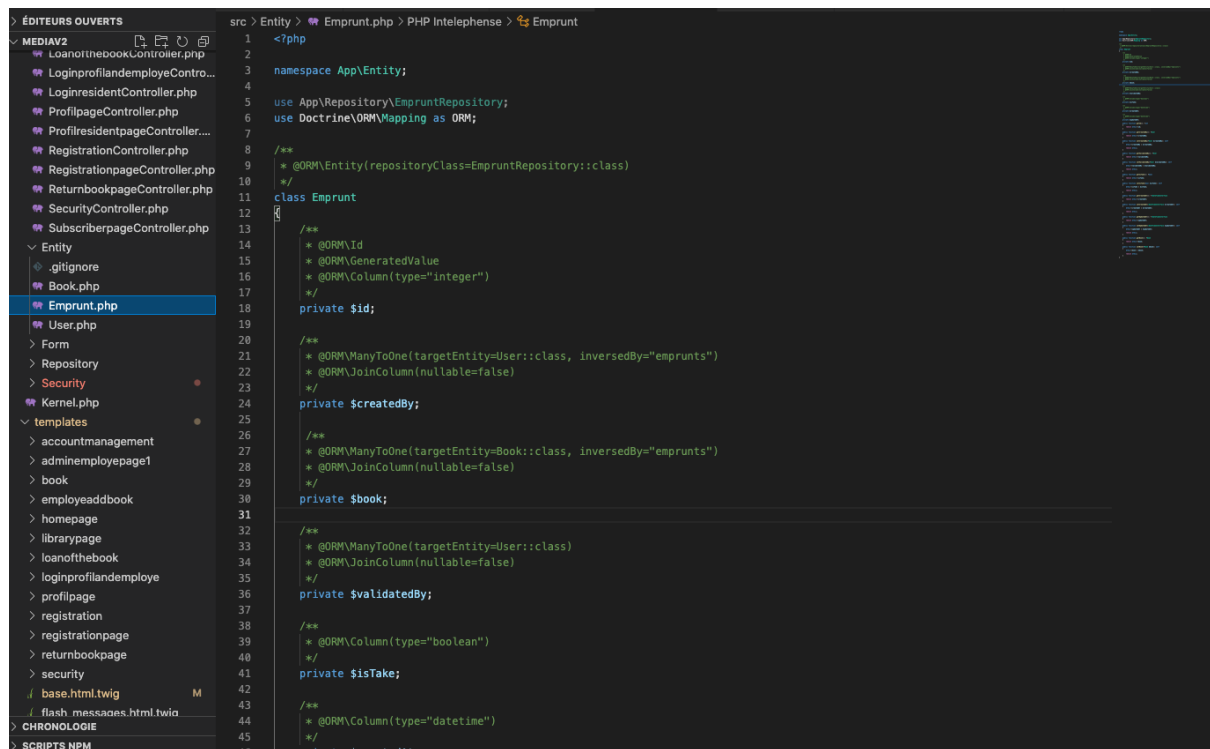
The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'Entity' and 'Form'. The code editor displays the 'Book.php' file, which is a Doctrine ORM entity. The code includes namespace declarations, use statements for Doctrine and Symfony, and a class definition for 'Book' with several attributes.

```
src > Entity > Book.php > PHP Intelephense > Book
1 </php>
2
3 namespace App\Entity;
4
5 use App\Repository\BookRepository;
6 use Doctrine\Common\Collections\ArrayCollection;
7 use Doctrine\Common\Collections\Collection;
8 use Doctrine\ORM\Mapping as ORM;
9
10 /**
11  * @ORM\Entity(repositoryClass=BookRepository::class)
12  */
13 class Book
14 {
15     /**
16      * @ORM\Id
17      * @ORM\GeneratedValue
18      * @ORM\Column(type="integer")
19      */
20     private $id;
21
22     /**
23      * @ORM\Column(type="string", length=255)
24      */
25     private $title;
26
27     /**
28      * @ORM\Column(type="string", length=255, nullable=true)
29      */
30     private $picture;
31
32     /**
33      * @ORM\Column(type="date")
34      */
35     private $datePublished;
36
37     /**
38      * @ORM\Column(type="string", length=255)
39      */
40     private $description;
41
42     /**
43      * @ORM\Column(type="string", length=255)
44      */
45     private $author;
46 }
```



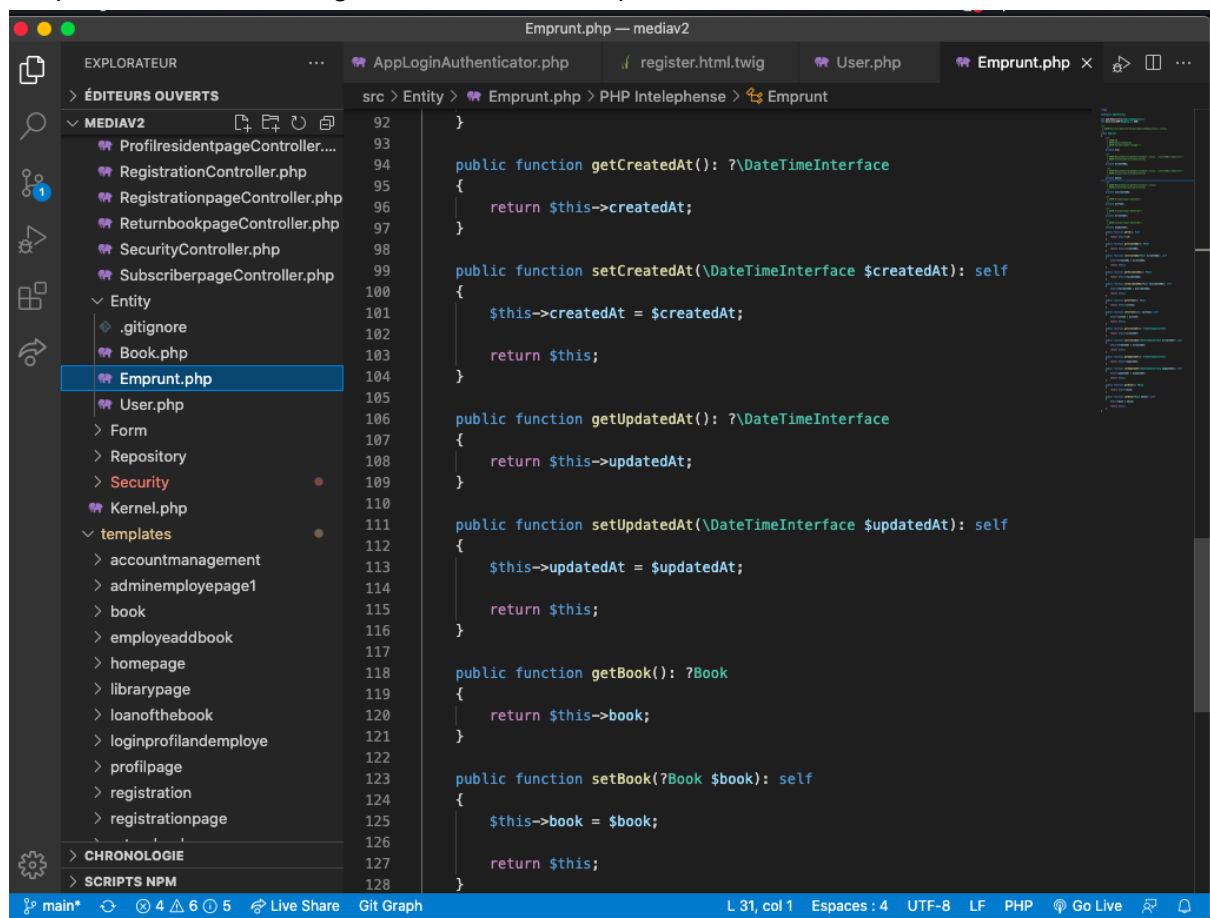
The screenshot shows the same IDE with the 'User.php' file open in the code editor. The code is similar to the 'Book.php' file, but for a 'User' entity. It includes namespace declarations, use statements for Doctrine and Symfony, and a class definition for 'User' that implements the 'UserInterface'.

```
src > Entity > User.php > ...
1 </php>
2
3 namespace App\Entity;
4
5 use App\Repository\UserRepository;
6 use Doctrine\Common\Collections\ArrayCollection;
7 use Doctrine\Common\Collections\Collection;
8 use Doctrine\ORM\Mapping as ORM;
9 use Symfony\Component\Security\Core\User\UserInterface;
10
11 /**
12  * @ORM\Entity(repositoryClass=UserRepository::class)
13  */
14 class User implements UserInterface
15 {
16     /**
17      * @ORM\Id
18      * @ORM\GeneratedValue
19      * @ORM\Column(type="integer")
20      */
21     private $id;
22
23     /**
24      * @ORM\Column(type="string", length=255)
25      */
26     private $name;
27
28     /**
29      * @ORM\Column(type="string", length=255)
30      */
31     private $lastName;
32
33     /**
34      * @ORM\Column(type="string", length=255)
35      */
36     private $email;
37
38     /**
39      * @ORM\Column(type="string", length=255)
40      */
41     private $password;
42
43     /**
44      * @ORM\Column(name="salt", type="string", length=255, nullable=true)
45      */
46     private $salt;
47 }
```



```
1 <?php
2
3 namespace App\Entity;
4
5 use App\Repository\EmpruntRepository;
6 use Doctrine\ORM\Mapping as ORM;
7
8 /**
9  * @ORM\Entity(repositoryClass=EmpruntRepository::class)
10  */
11 class Emprunt
12 {
13     /**
14      * @ORM\Id
15      * @ORM\GeneratedValue
16      * @ORM\Column(type="integer")
17      */
18     private $id;
19
20     /**
21      * @ORM\ManyToOne(targetEntity=User::class, inversedBy="emprunts")
22      * @ORM\JoinColumn(nullable=false)
23      */
24     private $createdBy;
25
26     /**
27      * @ORM\ManyToOne(targetEntity=Book::class, inversedBy="emprunts")
28      * @ORM\JoinColumn(nullable=false)
29      */
30     private $book;
31
32     /**
33      * @ORM\ManyToOne(targetEntity=User::class)
34      * @ORM\JoinColumn(nullable=false)
35      */
36     private $validatedBy;
37
38     /**
39      * @ORM\Column(type="boolean")
40      */
41     private $isTake;
42
43     /**
44      * @ORM\Column(type="datetime")
45      */
46     private $createdAt;
```

Scripts établit à la main : génération du mot de passe,



```
92 }
93
94 public function getCreatedAt(): ?\DateTimeInterface
95 {
96     return $this->createdAt;
97 }
98
99 public function setCreatedAt(\DateTimeInterface $createdAt): self
100 {
101     $this->createdAt = $createdAt;
102
103     return $this;
104 }
105
106 public function getUpdatedAt(): ?\DateTimeInterface
107 {
108     return $this->updatedAt;
109 }
110
111 public function setUpdatedAt(\DateTimeInterface $updatedAt): self
112 {
113     $this->updatedAt = $updatedAt;
114
115     return $this;
116 }
117
118 public function getBook(): ?Book
119 {
120     return $this->book;
121 }
122
123 public function setBook(?Book $book): self
124 {
125     $this->book = $book;
126
127     return $this;
128 }
```

