Atlas

Alexander Pela & Tristan Knott

What is Atlas?

" A web-based Learning Management Software crafted to meet the program needs of small-to medium-sized organizations.

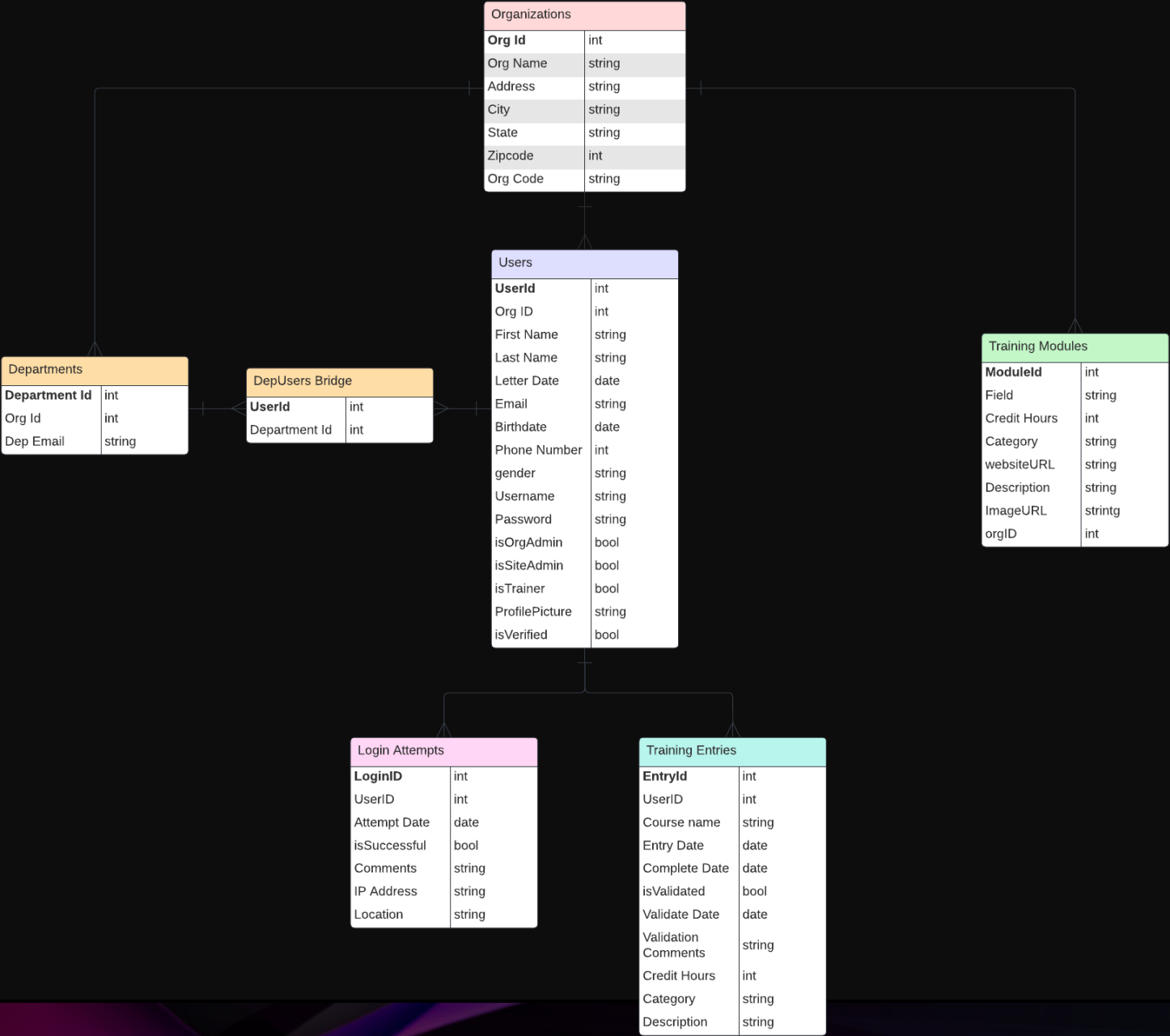| Users	
UserId	int
Org ID	int
First Name	string
Last Name	string
Letter Date	date
Email	string
Birthdate	date
Phone Number	int
gender	string
Username	string
Password	string
isOrgAdmin	bool
isSiteAdmin	bool
isTrainer	bool
ProfilePicture	string
isVerified	bool

Run Code

Examples

- Show Organization Join and Create:

- Personalize Account Settings/Enter Trainings on training manager.

  (ATLAS_JKerry1963, TestTrainer2!)

- Site Manager show managing an organization (Alexander, AlexanderPela)

ERD

Organizations

Org Id	int
Org Name	string
Address	string
City	string
State	string
Zipcode	int
Org Code	string

Users

UserId	int
Org ID	int
First Name	string
Last Name	string
Letter Date	date
Email	string
Birthdate	date
Phone Number	int
gender	string
Username	string
Password	string
isOrgAdmin	bool
isSiteAdmin	bool
isTrainer	bool
ProfilePicture	string
isVerified	bool

Departments

Department Id	int
Org Id	int
Dep Email	string

DepUsers Bridge

UserId	int
Department Id	int

Training Modules

ModuleId	int
Field	string
Credit Hours	int
Category	string
websiteURL	string
Description	string
ImageURL	strintg
orgID	int

Login Attempts

LoginID	int
UserID	int
Attempt Date	date
isSuccessful	bool
Comments	string
IP Address	string
Location	string

Training Entries

EntryId	int
UserID	int
Course name	string
Entry Date	date
Complete Date	date
isValidated	bool
Validate Date	date
Validation Comments	string
Credit Hours	int
Category	string
Description	string

File Structure (UserDB)

File tree:
- .vscode
- assets
  - css
  - images
    - atlasPhotos
    - profilePhotos
    - Default_pfp.svg.png
  - js
- include
  - aside.php
  - checkEmail.php
  - footer.php
  - functions.php
  - header.php
  - selectDepartments....
  - selectUsers.php
- model
  - dbconfig.ini
  - DepartmentsDB.php
  - LoginAttemptDB.ph
  - OrganizationsD   C:\
  - TrainingEntryDB.php
  - TrainingModuleDB.p.
  - UserDepBridgeDB.p..
  - UsersDB.php
- private
  - departments.php
  - landingPage.php
  - loginAttempts.php
  - logout.php
  - organizations.php
  - orgControlPanel.php
  - trainingEntry.php
  - trainingModules.php
  - userAccount.php
- public
  - index.php

```php
<?php

class UserDB {
    private $userData;

    //constructor for UserDB class
    public function __construct() {
        if ($ini = parse_ini_file('dbconfig.ini')) {
            $userPDO = new PDO( "mysql:host=" . $ini['servername'] .
                                ";port=" . $ini['port'] .
                                ";dbname=" . $ini['dbname'],
                                $ini['username'],
                                $ini['password']);

            $userPDO->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
            $userPDO->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

            $this->userData = $userPDO;
        } else {
            throw new Exception("<h2>Creation of database object failed!</h2>", 0, null);
        }
    }

    public function getAllUsers() {
        $results = [];
        $userTable = $this->userData;

        $sqlString = $userTable->prepare("SELECT * FROM (users
                                          INNER JOIN organizations ON users.orgID = organizations.orgID)
                                          ORDER BY lastName");

        if($sqlString->execute() && $sqlString->rowCount() > 0) {
            $results = $sqlString->fetchAll(PDO::FETCH_ASSOC);
```

```php
<?php if($action == 'Viewer'): ?>…
<?php elseif($action == 'createUser'): ?>…
<?php elseif($action == 'personalSettings'):…
<?php elseif($action == 'updateUser'): ?>…
<?php elseif($action == 'changePassword'): ?>…
<?php elseif($action == 'Validator'): ?>…
<?php endif; ?>
```

```php
<?php if($_SESSION['isSiteAdmin'] && isset($_SESSION['orgID'])): …
<?php elseif($_SESSION['isSiteAdmin'] && !isset($_SESSION['orgID'])): …
<?php elseif($_SESSION['isOrgAdmin']): …
<?php endif; ?>
```

Alex's Successes, Failures, Impediments

Successes:

- Understanding the "flow of code" (Pseudo coding with comments).

- Creation of full-stack software.

Failures:

- Job Codes feature was cut. (Too many levels of users and views).

Impediments:

- Maintaining a healthy balance between project, other classes, work, etc.

Tristan's Successes, Failures, Impediments

Successes:

- 1

- 2

Failures:

- 3

Impediments:

- 4

What would you do differently

- Design and Documentation time.

- Prioritize getting functionality coded first and then working on visual

  code.

Future Growth

Project:

- Make trainings be assigned by departments in an organization.

- Make Power BI like dashboards for users.

- Reintroduce Job Codes.

Personal:

- Work on website projects for other people to improve as Software

  Engineers.