Project Report

Aston Events

Name: Bradley Willies

Student ID: 180212217

Link to website entry page:

Organiser credentials:

Username:

Password:

Link to GitHub source code:

### Key Technologies and Structure of System

I have created my Aston Events website system using the technologies:

* Composer – to manage dependencies within the PHP project.
* Laravel – the PHP framework which creates the structure of the project and includes other technologies with it.
* Eloquent – to handle the object relational mapping of the database with the PHP models.

See Appendix A which shows the system design for reference. I followed the “model view controller” architecture, which is standard when using Laravel. For the views, I split them into the authentication pages and the event viewing pages which a student would view (without being logged in), and the pages an event organiser could access. I used three controllers; one to handle authentication, one to handle CRUD activities on the events and another to handle any of the student page activities. The models just reflected the database tables I created.

### Implemented Functions

|  |  |
| --- | --- |
| Functions | Source Files |
| S1 |  |
| S2 |  |
| S3 |  |
| S4 |  |
| S5 |  |
| E1 |  |
| E2 |  |
| E3 |  |
| E4 |  |

### Implemented Security Features

|  |  |
| --- | --- |
| Security Features | Source Files |
| Authentication/Authorisation |  |
| Form validation |  |
| Handle injections (SQL/HTML) |  |
| Hash password |  |
| Restrict file upload to only images |  |
| Cross-Site Request Forgery |  |

### Implemented Stretchers

|  |  |
| --- | --- |
| Stretchers | Source Files |
| Tables sorted based on headings |  |
| Events have multiple pictures |  |
| Student sends email to event organiser |  |
| Organiser links one event to another event and/or web sources |  |

### Database Schema

See Appendix B for the ERD describing the database schema.

The users table is setup automatically with Laravel authentication, so it makes sense to use this to store new users, such as registered organisers. All authentication details are kept in the users table, while all details specific to an organiser are kept in the organisers table.

The related\_event\_id field in the events table optionally references another event’s id.

### Points to Note When Running the System

* In order to create an event you are required to provide at least one image, along with other event details.
* The Contact form to send a message to an event organiser is live and uses MailJet with Laravel. It will send an email to the event organiser's email address. To test this; register as an organiser using your own email -> create an event -> go to the contact page for that event. When you send a message you should receive an email.

### Appendix A: System Design

### Appendix B: Entity Relationship Diagram