**Project Title:**

|  |
| --- |
| BUS TICKET RESERVATION SYSTEM |

**Short Description:**

|  |  |
| --- | --- |
| Role Name | Description |
| Admin | Admin can login to the system and verify Bus Manager account after registration a Bus manager. Can view/add/delete/modify/search Supports Staff, Bus Manager, Bus Counter, Counter Staff , Manage Bus, Bus Schedule time and locations information. In addition, he/she can view Bus wise Summary of any operator. |
| Bus Manager | Bus manager can login/logout to the system and registration as a bus manager by giving proper information. As a Bus manager he can view/add/delete/modify/search Bus counter, Counter Staff, Buses, Bus Schedule information. Can update own profile. |

**Modules:**

|  |  |  |
| --- | --- | --- |
| Login Module (M1) | |  |
| Function | Description | Database |
| M1F1 | In the view of of **Login page** in the route of **“/system/supportstaff/add”** and there’ll be a html form having fields **email, password,**and a login button to fire form submission to the same route with post request. | N/A |
| M1F2 | After receiving post request following form validation will be happen,  Validates using Laravel Illuminate\Supports\Validator   * Email validation. * Empty field validation. * Email existence validation (From database users table). | Match data from db table **“users”** and columns **email .** |
| M1F3 | PHP Manual validation for password cross checking (From database users table). | Match data from db table **“users”** and columns **email** and **password.** |
| M1F4 | If everything is fine **name, email, password** will be saved in the database. After then user will redirect to the system. | N/A |

|  |  |  |
| --- | --- | --- |
| Bus Manager Delete (M7) | |  |
| Function | Description | Database |
| M7F1 | In the List of Module **M4F1** a **Delete** Button will be added to every row referring the route **“/system/busmanager/{id}/delete”.**  Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the db table **“users”** |
| M7F2 | In the view there’ll be a javascript confirmation**.** After confirm a virtual form will be submitted to the route of **M4F1** on **Delete** Request | N/A |
| M7F3 | After receiving delete request support staff will be retrieved from the database and will be removed. After then user will redirect to **M4F1** | Find User from db table **“users”** by **“id”**  Delete from db table **“users”** by **“id”** |

|  |  |  |
| --- | --- | --- |
| Support Staff Edit (M11) | |  |
| Function | Description | Database |
| M11F1 | In the List of Module **M8F1** a **Edit** Button will be added to every row referring the route **“/system/supportstaff/{id}/edit”.**  Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the db table **“users”** |
| M11F2 | In the view there’ll be a html form having fields **name, email, password** and a submit button to fire form submission to the same route with **Patch** request. | N/A |
| M11F3 | After receiving patch request support staff will be retrieved from the database and following form validation will be happen,   * Name : Required * Email: Required | Valid Email | Unique except Current * Password: Nullable | Length:4   If everything is fine **name, email, password** will be saved in the database. After then user will redirect to **M8F1** | Find User from db table **“users”** by **“id”**  Updates **name=name, email=email, password=password** to the db table **“users”** |

|  |  |  |
| --- | --- | --- |
| Support Staff Delete (M12) | |  |
| Function | Description | Database |
| M12F1 | In the List of Module **M8F1** a **Delete** Button will be added to every row referring the route **“/system/supportstaff/{id}/delete”.**  Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the db table **“users”** |
| M12F2 | In the view there’ll be a javascript confirmation**.** After confirm a virtual form will be submitted to the route of **M8F1** on **Delete** Request | N/A |
| M12F3 | After receiving delete request support staff will be retrieved from the database and will be removed. After then user will redirect to **M8F1** | Find User from db table **“users”** by **“id”**  Delete from db table **“users”** by **“id”** |

|  |  |  |
| --- | --- | --- |
| Bus Counter Listing Module - Admin (M13) | |  |
| Function | Description | Database |
| M13F1 | **Bus Counter’s** listing view can be access by route **“/system/buscounter”** . In the view there will be a table having column **operator, manager, name, location**. All the respective data will be shown from database. Action column will be use to show action buttons such as **Edit, Delete** etc. | List will use db table **“buscounters”** and **“users”** and columns **name, location user. company, user.name as manager,** |

|  |  |  |
| --- | --- | --- |
| Bus counter Searching Module (M15) | |  |
| Function | Description | Database |
| M15F1 | It will use the view of **M13F1** to list the search result and can be access by same route. There’ll be a search field named **“search”** and a search submit button to fire the searching event**.** | List will use db table **“users”** and columns **id,** **name, email, company, registered** |
| M15F2 | Search event will call for an **Ajax (post) request** to **‘system/busmanager/ajax/search’** with the parameter **search** having value of search field. Once the search result is received then it will list on **M13F1 Listing** | Matches the search parameter with the db table **“users”** and columns **name, email, company.** Matching relation will be **OR** and pattern will be **LIKE %%** |

|  |  |  |
| --- | --- | --- |
| Buses Listing Module - Admin (M19) | |  |
| Function | Description | Database |
| M19F1 | **Buses** listing view can be access by route **“/system/ buses”** . In the view there will be a table having column **operator, manager, name, location, seat\_row,seat\_column**. All the respective data will be shown from database. Action column will be use to show action buttons such as **Edit, Delete** etc. | List will use db table **“buses”** and **“users”** and columns **name, location user.company, user.name as manager, seat\_row,seat\_column** |

|  |  |  |
| --- | --- | --- |
| Buses Listing Module - Bus Manager (M20) | |  |
| Function | Description | Database |
| M20F1 | **Buses** listing view can be access by route **“/system/ buses”** . In the view there will be a table having column **name, location, seat\_row, seat\_column, company**. All the respective data of **the bus manager** will be shown from database. Action column will be use to show action buttons such as **Edit, Delete** etc. | List will use db table **“buses”** and columns **name, location, seat\_row,seat\_column, company** |

|  |  |  |
| --- | --- | --- |
| Buses Searching Module (M21) | |  |
| Function | Description | Database |
| M21F1 | It will use the view of **M19F1** to list the search result and can be access by same route. There’ll be a search field named **“search”** and a search submit button to fire the searching event**.** | List will use db table **“users”** and columns **id name, location, seat\_row, seat\_column, company** |
| M21F2 | Search event will call for an **Ajax (post) request** to **‘system/ buses /ajax/search’** with the parameter **search** having value of search field. Once the search result is received then it will list on **M19F1 Listing** | Matches the search parameter with the db table **“buses”** and columns **name, location, seat\_row, seat\_column, company.** Matching relation will be **OR** and pattern will be **LIKE %%** |

|  |  |  |
| --- | --- | --- |
| Buses Add (M22) | |  |
| Function | Description | Database |
| M22F1 | In the view of Module **M19F1** a Add New Button will be added referring the route **“/system/buses/add”** | N/A |
| M22F2 | In the view there’ll be a html form having fields **name, location, seat\_row, seat\_column, company** and a submit button to fire form submission to the same route with post request. | N/A |
| M22F3 | After receiving post request following form validation will be happen,   * Name: Required * operator: Required * location: Required * seat\_row: Required * seat\_column: Required   If everything is fine **name, location, seat\_row, seat\_column, company** will be stored in the database. After then user will redirect to **M19F1** | Stores  **name=name, operator=operator, seat\_row, seat\_column, location=location**, to the db table **“buses”** |

**Access control:**

|  |  |
| --- | --- |
| Role name | Module(M), Functions(F) |
| Admin | * M1F1,M1F2,M1F3,M1F4, M11F1,M11F2,M11F3, M12F1,M12F2,M12F3, M13F1, M15F2,M15F2 * M16F1,M16F2,M16F3, M17F1,M17F2,M17F3, M18F1,M18F2,M18F3 |
| Bus Manager | * M1F1,M1F2,M1F3,M1F4, M17F1,M17F2,M17F3, M18F1,M18F2,M18F3 * M19F1, M20F1, M21F1, M21F2, M22F1, M22F2, M22F3 |

**M – 1, 7, 11,12,13,15, 18,19,20,21,22, =11**

**Submission Instruction:**

1. Implement only the features that is listed in the access control table.
2. Create a git repository named “Individual\_Assignment”**.**
3. Do the commit after implementing every feature.
4. Commit history is very important for the evaluation so do it properly. All the commit name should be the feature name or identifier code (Ex: M1F1).
5. Upload the GitHub link in VUES under the assignment section.