Student pre-qualification System -Screens design

# Public screens

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | **Angular** | **Express** | **Mongo** |
| Home Screen  Login    Enroll  Enrollment info Text content | Check if logged in auto redirect  Routes:   * Login * Enroll * Admin/staff |  |  |
| Login Screen  Login  password  email | Routes: (role)   * Admin * Staff   Token-> localstorage | /authenticate | Users {email, password} |
| Enroll form screen  User details form  Enroll | Checkemail exist  Submit | /checkemail  /students (post) | Students {email}  Students -> new |

# Staff screens (staff role/logged in only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | **Angular** | **Express** | **Mongo** |
| Staff Home Screen  enrolled  new | Routes:   * New students * Enrolled |  |  |
| New Students Screen  Newly enrolled students screens  invite | * Get students/ status=new * Invite-submit | Students/:status  /invite/:email | Students {status: new}  Students {invite\_sent:false}  Create exam Compose and send email |
| Enrolled Students screen  Re-invite  Invited, answered  Passed/ failed | * Get students/ status=!new * Re-nvite-submit | Students/:status  /reinvite/:email | Students {status: $not:new}  Clear exam records, create exam, compose and send email |

# Admin screens (Admin role/logged in only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | **Angular** | **Express** | **Mongo** |
| Admin Home Screen  Grade Students  Manage staff  Manage Questions | Routes:   * Manage staff * Manage Questions * Grade Students |  |  |
| Manage Staff screen  List current staff users  CRUD | * Get staff list * New staff * Update staff * Delete staff * Toggle active | users/ (get)  users/ (post)  users/:id (put)  users/:id/(del)  users/:id/(patch) | users {\*}  users => new  users{id} update  remove users{id}  update users{id} |
| Manage Questions screen  CRUD  List current questions | * Get questions list * New question * Update question * Delete question * Toggle active | questions/ (get)  questions/ (post)  questions/:id (put)  questions/:id/(del)  questions/:id/(patch) | questions {\*}  questions => new  questions {id} update  remove questions {id}  update questions {id} |
| Answered students screen  Grade  Answered students list | * Gest students list * Route: Grade | Students/:status | Students{status:  Answered} |
| Grade Student Screen  Results  Score  Play  Q3  Score  Play  Q2  score  Play  Q1 | * Get questions * Results submit | Students/exam/:id  Students/grade/:id | Students{snapshots, monitoring  Students{status:  Pass/fail} |

# Take the exam (passwordless login/email link)

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | **Angular** | **Express** | **Mongo** |
| Exam home page  Exam instructions  Start | Check token  Check chrome  Routes:   * examscreen | Passwordless logic | Students{token} |
| Take Exam Screen  End exam  Submit  Submit  Submit  Q2  Q2  Q1 | Exam start  Timer   * opentab * question duration * snapshot   exam-submit | students/start/:id  students/snap/:id  students/submit/:id | Students{exam\_start}  Settimeout(120 min)  Students{update}  Check time, update students{id} |

## frontend routes

* Login
* enrollform
* adminhome
* staffhome
* newstudents
* enrolledstudents
* managestaff
* managequestions
* answeredstudents
* gradeexam
* examhome
* takeexam

# backend routes

* authenticate
* checkemail
* students/:status
* students/invite/:email
* students/reinvite/:email
* students/start/:email
* students/end/:email
* studetns/stats/:email
* users
* questions