American International University- Bangladesh

ADVANCED PROGRAMMING IN WEB TECHNOLOGY

Mid Project Report (Node.js)

Spring 2019-2020

Project Title: Research Management System
Section: B

Student Name	Student Id
Islam, Md Sameull	**17-33950-1**
Adnan, Khandakar Anim Hassan	**17-33956-1**
Zidan, Mehedi Hassan	**17-33963-1**

Research Management System

User Type

- ***Admin***

    1. **Admin can login to the system with a valid username and password.** *(model: AdminUserModel.js, views: /login.ejs, controllers: /login.js)*

    2. **Admin can add a teacher account to the system.** *(model: AdminFacultyModel.js, AdminUserModel.js, views: /admin /AdminTeacherReg.ejs, controllers: /admin/AdminTeacherReg.js)*

    3. **Admin can see teachers details, update teacher's account information and delete teacher's account.** *(model: AdminFacultyModel.js, AdminUserModel.js, views: /admin /AdminTeacherDetails.ejs, controllers: /admin/AdminTeacherDetails.js)*

    4. **Admin can add a student account to the system.** *(model: AdminStudentModel.js, AdminUserModel.js, views: /admin /AdminStudentReg.ejs, controllers: /admin/AdminStudentReg.js)*

    5. **Admin can see students details, update student's account information and delete student's account.** *(model: AdminStudentModel.js, AdminUserModel.js, views: /admin /AdminStudentDetails.ejs, controllers: /admin/AdminStudentDetails.js)*

    6. **Admin can see pending students list, download student's uploaded file to verify, verify student's account.***(model: AdminStudentModel.js, AdminVerificationModel.js, AdminUserModel.js, views: /admin /AdminStudentApproval.ejs, controllers: /admin/AdminStudentApproval.js)*

    7. **Admin can temporary block student's account.** *(model: AdminStudentModel.js, views: /admin /AdminStudentBlock.ejs, controllers: /admin/AdminStudentBlock.js)*

8. **Admin can unblock student’s account.** *(model: AdminStudentModel.js, views: /admin /AdminStudentUnblock.ejs, controllers: /admin/AdminStudentUnblock.js)*

9. **Admin can temporary block teacher’s account.** *(model: AdminTeacherModel.js, views: /admin /AdminTeacherBlock.ejs, controllers: /admin/AdminTeacherBlock.js)*

10. **Admin can unblock teacher’s account.** *(model: AdminTeacherModel.js, views: /admin /AdminTeacherUnblock.ejs, controllers: /admin/AdminTeacherUnblock.js)*

11. **Admin can add a new semester and make the last semester as current semester.** *(model: AdminSemesterModel.js, views: /admin/AdminSemesterAdd.ejs, controllers: /admin/AdminSemesterAdd.js)*

12. **Admin can see semesters details, update semester and delete semester.** *(model: AdminSemesterModel.js, views: /admin /AdminSemesterDetails.ejs, controllers: /admin/AdminSemesterDetails.js)*

13. **Admin can add a new thesis type.** *(model: AdminTypeModel.js, views: /admin/AdminTypeAdd.ejs, controllers: /admin/AdminTypeAdd.js)*

14. **Admin can see thesis types details, update thesis type and delete thesis type.** *(model: AdminTypeModel.js, views: /admin /AdminTypeDetails.ejs, controllers: /admin/AdminTypeDetails.js)*

15. **Admin can add a new domain for research.** *(model: AdminDomainModel.js, views: /admin/AdminDomainAdd.ejs, controllers: /admin/AdminDomainAdd.js)*

16. **Admin can see domain details, update domain and delete domain.** *(model: AdminDomainModel.js, views: /admin /AdminDomainDetails.ejs, controllers: /admin/AdminDoaminDetails.js)*

17. **Admin can offer topic with domain, supervisor, thesis type for research.** *(model: AdminSubDomainModel.js, AdminFacultyModel.js, AdminDominModel.js, AdminTypeModel.js, AdminGroupModel.js views: /admin/AdminOfferTopic.ejs, controllers: /admin/AdminOfferTopic.js)*

18. **Admin can see topic details, update topic and delete topic.** *(model: AdminSubDomainModel.js, AdminFacultyModel.js, AdminDominModel.js, AdminTypeModel.js views: /admin/AdminTopicDetails.ejs, controllers: /admin/AdminTopicDetails.js)*

19. **Admin can allocate external to a specific group.** *(model: AdminThesisModel.js, AdminFacultyModel.js, AdminGroupModel.js views: /admin /AdminAllocateFaculty.ejs, controllers: /admin/ AdminAllocateFaculty.js)*

20. **Admin can upload file (file type - .pdf, .doc, .docs and max size – 5.5MB) to a specific group.** *(model: AdminFileModel.js, AdminGroupModel.js views: /admin /AdminUploadFile.ejs, controllers: /admin/ AdminUploadFile.js)*

21. **Admin can see thesis details of a specific group and also see the group members list.** *(model: AdminThesisModel.js views: /admin /AdminThesisDetails.ejs, controllers: /admin/ AdminThesisDetails.js)*

22. **Admin can change own password.** *(model: AdminUserModel.js views: /admin /AdminChangePassword.ejs, controllers: /admin/AdminChangePassword.js)*

23. **Admin can logout from the system.** *(views: /logout.ejs, controllers: /logout.js)*

Research Management System

- ***Faculty***

  1. **Faculty can login to the system with a valid username and password.** *(model: facultyUserModel.js, views: /login.ejs, controllers: /login.js)*

  2. **Faculty can add a student account to the system.** *(model: facultyUserModel.js & facultyStudentModel ,views: /faculty/studentReg.ejs, controllers: /faculty/studentReg.js)*

  3. **Faculty can see student details.** *(model: facultyStudentModel ,views: /faculty /studentDetails.ejs, controllers: /faculty/studentDetails.js)*

  4. **Faculty can approve students.** *(model: facultyStudentModel.js ,views: /faculty / studentApproval.ejs, controllers: /faculty/ studentApproval.js)*

  5. **Faculty can change his or her password.** *(model: facultyUserModel.js, views: /faculty / changePassword.ejs, controllers: /faculty/ changePassword.js)*

  6. **Faculty can update, view his or her profile details.** *(model: facultyFacultyModel.js & facultyUserModel.js, views: /faculty / profile.ejs, controllers: /faculty/ profile.js)*

  7. **Faculty can offer thesis topic.** *(model: facultyThesisTypeModel.js , facultyDomainModel,js, facultySubDomainModel.js, facultyFacultyModel.js, facultyResearchGroupModel.js , ,views: /faculty / topicAdd.ejs, controllers: /faculty/ topicAdd.js)*

  8. **Faculty can update, view his or her profile details.** *(model: facultySubDomainModel.js & facultyStudentThesisModel.js, views: /faculty / viewTopic.ejs, controllers: /faculty/ viewTopic.js)*

  9. **Faculty can update thesis work progress.** *(model: facultyStudentThesisModel.js, views: /faculty / progressUpdate.ejs, controllers: /faculty/ progressUpdate.js)*

  10. **Faculty can upload file (file type - .pdf, .doc, .docs and max size – 5.5MB) to a specific group.** *(model: facultySubDomainModel.js & facultyFileModel.js, views: /faculty / uploadFile.ejs, controllers: /faculty* **/** *uploadFile.js)*

Research Management System

- ***Student***

1. **A student can register through Register Page with his valid information. He must provide each and every information in the required fields. After that student will need to click on Sign Up button to complete the register. If any error event presents, he will get an error message on the top of the page.** *(model: StudentStudentModel.js, StudentUserModel.js, StudentVerificationModel.js. views: /student/reg/index.ejs, controllers: /student/reg.js)*

2. **After a successful registration a student will get his login credentials with his User ID and randomly generated Password through which he can log in to the system.** *(model: StudentUserModel.js, views: /student/reg/regInfo.ejs, controllers: /student/reg.js)*

3. **A student can login to the system after a successful registration with his registered User ID and the provided password. Any inactive or invalid user is prohibited to access to the system. Only a valid and active user can login to the system. Unauthorized users will get alert messages on the top of the page.** *(model: StudentUserModel.js, views: /login.ejs, controllers: /login.js)*

4. ***After logging in to the system, a student can see his own information***. *(model: StudentStudentModel.js, views: /student/home/index.ejs, controllers: /student/home.js)*

5. **A student can update his own information (without User ID, Email and Registration Date). A student must provide all valid information for updating his profile. Any invalid request will return corresponding alert message and dismiss the update operation.** *(model: StudentUserModel.js, StudentStudentModel.js, views: /student/home/index.ejs, controllers: /student/home.js)*

6. **A student can see all the available topics propsed by the faculties. He/She can also see each topic details by clicking Details button.** *(model: StudentTopicModel.js, StudentFacultyModel.js, StudentDomainModel.js, StudentTopicModel.js views: student/topics/index.ejs, /student/topics/topicDetails.ejs, controllers: /student/topic.js)*

7. **If a student has not been enrolled in a topic, he can apply for one and at most one chosen topic to enroll in by clicking apply button to the corresponding topic sections. All he need is to wait for the enrolling confirmation given by the admin.** *(model: StudentTopicModel.js, StudentFacultyModel.js, StudentDomainModel.js, StudentTopicModel.js, StudentGroup.js, StudentThesisModel.js, StudentThesisTypeModel.js ,StudentThesisApplied.js, views: /student/topics/index.ejs, /student/topics/topicDetails.ejs, controllers: /student/group.js)*

8. **A student can see the topic in which he is enrolled in with necessary information (ex: progress etc).** *(model: StudentTopicModel.js, StudentFacultyModel.js, StudentDomainModel.js, StudentTopicModel.js, StudentGroup.js, StudentThesisModel.js, StudentThesisTypeModel.js ,StudentThesisApplied.js, views: /student/myResearch/index.ejs ,controllers: /student/myResearch.js)*

9. **There will be one group for each topic and each group can contain at most 4 members. If any topic has been chosen for a group and group becomes full of 4 members, no application request will be accepted.** *(model: StudentGroup.js ,StudentThesisApplied.js, views: /student/topics/index.ejs, /student/topics/topicDetails.ejs, controllers: /student/group.js)*

10. **A student can see other members and necessary information who are engaged in the same group**. *(model: StudentTopicModel.js, StudentFacultyModel.js, StudentDomainModel.js, StudentTopicModel.js, StudentGroup.js, StudentThesisModel.js, StudentThesisTypeModel.js ,StudentThesisApplied.ejs, views: /student/group/myGroup.ejs ,controllers: /student/group.js)*

11. **A student can upload necessary documents in the group (.pdf, .doc, .docx). Invalid file upload will not accept the upload request and return alert messages.** *(model: StudentFilesModel.js, views: /student/upload/index.ejs ,controllers: /student/upload.js)*

12. **A student can see all the uploaded documents in the group repository if uploaded any.** *(model: StudentFilesModel.js, views: /student/download/index.ejs ,controllers: /student/download.js)*

13. **A student can download necessary documents (.pdf, .doc, .docx) from the group repository if uploaded any.** *(model: StudentFilesModel.js, views: /student/download/index.ejs ,controllers: /student/download.js)*

14. **A student can change his password by using the correct Old Password and New Password with confirmation Password. Any incorrect request will return corresponding messages and will be visible to the top of the page. After a successful password change, the system will be logged out autometically.** *(model: StudentUserModel.js, views: /student/changePassword/index.ejs ,controllers: /student/changePassword.js)*

15. **Finally a student can log out of the system.** *(views: from any page except /student/upload/index.js ,controllers: /student/logout.js)*