1. Register

* POST /api/auth/register ile aşağıdakileri alacağız. Sonra Email Verification Page e redirect edeceğiz.
* firstName: The first name of the user.
* lastName: The last name of the user.
* username: The username of the user.
* email: The email address of the user (must be a valid Sabancı University email).
* password: The password for the user account.

1. Email Verification

* POST /api/auth/verify-email ile aşağıdakileri alacağız, email kısmı otomatik gelirse güzel olur register sayfasından. HTTP 200 OK gelirse Login Page e redirect edeceğiz.
* email: The email address of the user.
* verificationCode: The verification code sent to the user's email.

1. Login

* POST /api/auth/login ile aşağıdakileri alacağız, role = “USER” ve HTTP 200 ise Main Page e redirect, role = “ADMIN” ise Admin Main Page e redirect; HTTP 202 ise 2-FA Page e redirect edeceğiz.
* usernameOrEmail: The username or email address of the user.
* password: The password for the user account.

1. 2-FA

* POST /api/auth/verify-2fa ile aşağıdakileri alacağız, email kısmı otomatik gelirse login sayfasından güzel olur. HTTP 200 ve role = “USER” ise Main Page e redirect, role = “ADMIN” ise Admin Main Page e redirect
* email: The email address of the user.
* verificationCode: The 2FA verification code sent to the user's email.

1. Forgot Password

* POST /api/auth/forgot-password ile email alacağız, HTTP 200 ya da HTTP 400 Bad Request: "A password reset code has already been sent. Please wait until the current code expires." dönerse, Reset Password Page e yönlendireceğiz.
* email: The email address of the user.

1. Reset Password

* POST /api/auth/reset-password ile aşağıdakileri alacağız, email otomatik gelirse iyi olur. HTTP 200 gelirse Login Page e redirect ediyoruz.
* email: The email address of the user.
* code: The password reset code sent to the user's email.
* newPassword: The new password for the user account.

1. Main Page

* GET /api/course-view-log/top-visited ile son 30 günde en çok ziyaret edilen 10 kursu alacağız, main page de göstereceğiz. Çıktıların üzerine tıklanabilir olacak, o sayfanın page ine yönlendireceğiz.
* GET /api/instructor-view-log/top-visited ile son 30 günde en çok ziyaret edilen 10 hocayı alacağız, main page de göstereceğiz. Çıktıların üzerine tıklanabilir olacak, o sayfanın page ine yönlendireceğiz.
* Courses butonu ile Course Select Page e redirect edeceğiz.
* Instructors butonu ile Instructor Select Page e redirect edeceğiz.
* Academic Program Planner butonu ile Program Select Page e redirect edeceğiz.
* Profile butonu ile Profile Page e redirect edeceğiz.
* Profile butonu altından çıkan Logout butonu ile POST /api/auth/logout ile logout olacağız, HTTP 200 ise tabi, Login Page e redirect edeceğiz.
* Bunlar dışında bizim hakkımızda açıklamalar fln olacak.

1. Admin Main Page

* MAIN PAGE E İLAVE ADMIN CONSOLE OLACAK, SEÇENEKLER ARASINDA:
* Add Course butonu ile Admin Add Course Page e redirect edeceğiz.
* Change Course butonu ile Admin Change Course Page e redirect edeceğiz.
* Add Instructor butonu ile Admin Add Instructor Page e redirect edeceğiz.

1. Admin Add Course

* POST /api/courses/add ile aşağıdakileri alacağız, yeni kurs ekleyeceğiz.
* {
* "subject": "CS",
* "courseCode": "103",
* "courseNameEn": "Data Structures",
* "courseNameTr": "Veri Yapıları",
* "suCredit": 3,
* "ectsCredit": 6,
* "engineeringEcts": 3,
* "basicScienceEcts": 0,
* "contentEn": "This course covers data structures.",
* "contentTr": "Bu ders veri yapılarını kapsar.",
* "linkEn": "http://example.com/cs103",
* "linkTr": "http://example.com/cs103-tr",
* "faculty": "FENS",
* "courseStatus": true
* }
* NOT: FACULTY KISMI FASS, FMAN, FENS, SL OLARAK SEÇİLMELİ SADECE.

1. Admin Change Course

* Ekranda ikişer tane subject ve course-code seçici olacak, ikisi de course\_id elde etmek için.
* GET /api/courses/subjects ile bütün unique subjectleri alacağız, admine seçtireceğiz.
* GET /api/courses/{subject}/codes ile seçilen subjecte ait bütün code ları alacağız, admine seçtireceğiz.
* GET /api/courses/{subject}-{courseCode} ile kurs detaylarını alacağız, id bize yetecek.
* POST /api/admin/change-course ile aşağıdakileri alacağız, eski kursu deaktif edeceğiz ve eski kursun yorumlarını yeni kursun yorumlarına, başlarına “From {eski kurs}: ” koyarak taşımış olacağız.
* oldCourseId: The ID of the course to be changed (as a query parameter).
* newCourseId: The ID of the new course (as a query parameter).

1. Admin Add Instructor

* POST /api/instructors/add ile aşağıdakileri alacağız, yeni kurs ekleyeceğiz.
* {
* "name": "John",
* "surname": "Smith",
* "department": "Computer Science",
* "imageUrl": "https://example.com/profile.jpg",
* "aboutTr": "Bu öğretim üyesi hakkında Türkçe bilgiler.",
* "aboutEn": "Information about this instructor in English.",
* "linkTr": "https://example.com/tr/instructor/1",
* "linkEn": "https://example.com/en/instructor/1"
* }

1. Course Select

* GET /api/courses/subjects ile bütün unique subjectleri alacağız, kullanıcıya seçtireceğiz.
* GET /api/courses/{subject}/codes ile seçilen subjecte ait bütün code ları alacağız, kullanıcıya seçtireceğiz. Sonra Course Review Page e redirect edeceğiz.

1. Course Review

* GET /api/courses/{subject}-{courseCode} ile kurs detaylarını alacağız.
* POST /api/course-view-log/log ile ziyaret loglanacak.
* GET /api/course-comments/{courseId} ile (id bir önceki api den geliyor) kursa ait bütün kurs yorumlarını döneceğiz.
* GET /api/course-comment-ratings/{commentId}/stats ile bütün yorumlar için tek tek like dislike sayısını öğreneceğiz.
* GET /api/course-comment-ratings/{commentId}/user-rating ile user yorumları önceden beğenmiş mi ya da dislikelamış mı göreceğiz, bütün yorumlar için tek tek.
* Notes of this course butonu ile bu dersin Note Page e redirect.
* YORUM YAZMA, DÜZENLEME, SİLME
* POST /api/course-comments/{courseId} ile yorum ekleme aşağıdakileri alıp:
* {
* "content": "This course is amazing!",
* "anonymous": false,
* "termTaken": 202401,
* "gradeReceived": "A",
* "parentCommentId": null
* }
* PUT /api/course-comments/{commentId} ile yorum editleme aşağıdakileri alıp:
* {
* "content": "This course is fantastic!",
* "anonymous": true,
* "termTaken": 202401,
* "gradeReceived": "A-"
* }
* DELETE /api/course-comments/{commentId} ile yorum silme comment id ile.
* YORUM LİKE / DİSLİKE ATMA VE BUNLARI KALDIRMA
* POST /api/course-comment-ratings/{{commentId}/rate ile yorum like atma ya da dislike atma. Aşağıdaki ile.
* {
* "isLike": true
* }
* DELETE /api/course-comment-ratings/{{commentId}/rate ile yoruma atılmış like ya da dislike silme.
* NOT: kullanıcı like attıysa ve sonra dislike atacaksa, DOĞRUDAN POST APISINI KULLANABİLİRİZ ÇÜNKÜ ESKİSİNİ SİLİYOR OTOMATİK AMA DİREKT LİKE ATTIĞI YORUMA TIKLAYIP GERI ALACAKSA, O ZAMAN DELETE KULLANILACAK.

1. Admin Course Review

* NORMAL COURSE REVIEW E İLAVE AŞAĞIDAKİLER:
* Bütün yorumlar silinebilir olacak.
* POST /api/admin/ban-user ile yorumları yapan userlar banlanabilecek aşağıdakiler ile:
* userId: The ID of the user to be banned (as a query parameter).
* reason: The reason for the ban (as a query parameter).
* Update this course butonu ile Admin Update Course Page e redirect.

1. Note

* GET /api/notes/{courseId} ile kursa ait bütün notlar gelecek.
* POST /api/notes/{courseId} ile kursa not eklenebilecek aşağıdakilerle:
* {
* "termTaken": 202401,
* "instructorTaken": "Dr. John Smith",
* "description": "This is a note about the course material.",
* "cloudLink": "https://drive.google.com/file/d/12345abcde"
* }
* PUT /api/notes/{courseId} ile not düzenlenebilecek aşağıdakilerle:
* {
* "termTaken": 202402,
* "instructorTaken": "Dr. Jane Doe",
* "description": "Updated note description.",
* "cloudLink": "https://drive.google.com/file/d/67890fghij"
* }
* DELETE /api/notes/{courseId} ile not silinebilecek.

1. Admin Note

* NORMAL NOTE A İLAVE:
* Bütün yorumlar silinebilecek.

1. Admin Update Course

* PUT /api/courses/{courseId} ile bir kursu güncelleme aşağıdakilerle, id bir önceki sayfadan gelecek otomatik:
* {
* "subject": "CS",
* "courseCode": "103",
* "courseNameEn": "Advanced Data Structures",
* "courseNameTr": "İleri Veri Yapıları",
* "suCredit": 4,
* "ectsCredit": 7,
* "engineeringEcts": 4,
* "basicScienceEcts": 0,
* "contentEn": "This course covers advanced data structures.",
* "contentTr": "Bu ders ileri veri yapılarını kapsar.",
* "linkEn": "http://example.com/cs103-advanced",
* "linkTr": "http://example.com/cs103-advanced-tr",
* "faculty": "Engineering",
* "courseStatus": true
* }

1. Instructor Select

* GET /api/instructors/all ile bütün instructorlar gelecek, user a seçtirilecek. Sonra Instructor Review Page e redirect.

1. Instructor Review

* GET /api/instructors/{id} ile instructor detayı alınacak.
* POST /api/instructor-view-log/log ile ziyaret loglanacak.
* GET /api/instructor-comments/{instructorId} ile bütün yorumlar alınacak.
* GET /api/instructor-comment-ratings/{commentId}/stats ile bütün yorumlar için tek tek like dislike sayısını öğreneceğiz.
* GET /api/instructor-comment-ratings/{commentId}/user-rating ile user yorumları önceden beğenmiş mi ya da dislikelamış mı göreceğiz, bütün yorumlar için tek tek.
* YORUM YAZMA, DÜZENLEME, SİLME
* POST /api/instructor-comments/{instructorId} ile yorum ekleme aşağıdakileri alıp:
* {
* "content": "This instructor is amazing!",
* "anonymous": false,
* "parentCommentId": null
* }
* PUT /api/instructor-comments/{commentId} ile yorum editleme aşağıdakileri alıp:
* {
* "content": "This instructor is really great!",
* "anonymous": true
* }
* DELETE /api/instructor-comments/{commentId} ile yorum silme comment id ile.
* YORUM LİKE / DİSLİKE ATMA VE BUNLARI KALDIRMA
* POST /api/instructor-comment-ratings/{commentId}/rate ile yorum like atma ya da dislike atma. Aşağıdaki ile.
* {
* "isLike": true
* }
* DELETE /api/instructor-comment-ratings/{commentId}/rate ile yoruma atılmış like ya da dislike silme.
* NOT: kullanıcı like attıysa ve sonra dislike atacaksa, DOĞRUDAN POST APISINI KULLANABİLİRİZ ÇÜNKÜ ESKİSİNİ SİLİYOR OTOMATİK AMA DİREKT LİKE ATTIĞI YORUMA TIKLAYIP GERI ALACAKSA, O ZAMAN DELETE KULLANILACAK.

1. Admin Instructor Review

* NORMAL INSTRUCTOR REVIEW E İLAVE AŞAĞIDAKİLER:
* Bütün yorumlar silinebilir olacak.
* POST /api/admin/ban-user ile yorumları yapan userlar banlanabilecek aşağıdakiler ile:
* userId: The ID of the user to be banned (as a query parameter).
* reason: The reason for the ban (as a query parameter).
* Update this instructor butonu ile Admin Update Instructor Page e redirect.

1. Admin Update Instructor

* PUT /api/instructors/{id} ile bir kursu güncelleme aşağıdakilerle, id bir önceki sayfadan gelecek otomatik:
* {
* "name": "John",
* "surname": "Smith",
* "department": "Mathematics",
* "imageUrl": "https://example.com/new\_profile.jpg",
* "aboutTr": "Güncellenmiş Türkçe bilgiler.",
* "aboutEn": "Updated information in English.",
* "linkTr": "https://example.com/tr/instructor/updated",
* "linkEn": "https://example.com/en/instructor/updated"
* }

1. Profile

* GET /api/profile ile profil bilgileri alınacak.
* PUT /api/profile/2fa ile 2FA opsiyonu açılıp kapatılabilecek.
* PUT /api/profile ile aşağıdaki bilgiler güncellenebilecek:
* {
* "name": "John",
* "surname": "Smith",
* "major": "Computer Science and Engineering",
* "minor": "Business",
* "second\_major": "Mathematics",
* "start\_term": 202101,
* "graduation\_term": 202501
* }

1. Program Select

* GET /api/programs/unique-names ile bütün unique program isimleri alınacak
* Admission Term girilecek sonra ve Planner Page e redirect.

1. Planner

* GET /api/programs/details ile programın detayları alınacak aşağıdakilerle:
* nameEn: The English name of the program (as a query parameter).
* - admissionTerm: The admission term of the program (as a query parameter, format: YYYYMM).
* GET /api/programs/{programId}/courses ile kurslar alınacak, grupları buradan elde edilmiş olacak.
* GET /api/prerequisites/{courseId} ile tek tek bütün kursların prerequisite bilgisi alınacak.
* GET /api/courses/{subject}-{courseCode} ile tek tek bütün kursların kredi bilgileri alınacak.
* GET /api/student-plan/all ile başta öğrencinin geçmişte kaydettiği bütün dersler gelecek.
* DELETE /api/student-plan/delete-all ile, user programını kaydedecekse eski kaydedilmiş bütün planlar silinecek, sonra alttaki api ile
* POST /api/student-plan/save ile aşağıdakileri alarak tek tek user ın seçtiği programlar kaydedilecek.
* courseId: The ID of the course to add to the student's plan (as a query parameter).
* term: The term in which the student plans to take the course (as a query parameter, format: X).
* NOT: YANİ EN BAŞTA VARSA ESKİ PLAN GELECEK, SONRA EĞER USER TEKRARDAN KAYDET TUŞUNA BASARSA, ÖNCE GEÇMİŞTEKİ HER ŞEY SİLİNECEK, SONRA EKRANDAKİ PLAN TEK TEK KAYDEDİLECEK.