

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



Τμήμα Πληροφορικής

Σύγχρονα θέματα Τεχνολογίας Λογισμικού - Λογισμικό για κινητές συσκευές

Αριστοτέλης Ματακιάς - Α.Μ: Π19100

Βασίλη Γκιάτα - Α.Μ. Π19036

ΕΡΓΑΣΙΑ ΣΤΟ ΜΟΝΤΕΛΟ MODEL VIEW CONTROLLER

ΤΕΧΝΙΚΟ ΕΓΧΕΙΡΙΔΙΟ



Περιεχόμενα

1	Εισαγωγή	2
2	Ζητούμενα	3
3	Επίπεδο Model	3
4	Επίπεδο View και Controller	3
4.1	Session Management	3
4.2	Students Controller	3
4.2.1	Περιγραφή των μεθόδων	3
4.2.2	Μέθοδος IActionResult Grades(bool? asc=true)	3
4.2.3	Μέθοδος IActionResult Total()	3
4.2.4	Μέθοδος IActionResult Account()	3
4.2.5	Μέθοδος IActionResult Index()	3
4.2.6	Μέθοδος IActionResult Semesters(int? page)	4
4.2.7	Σενάριο Εκτέλεσης	4
4.3	Professors Controller	4
4.3.1	Μέθοδος Task<IActionResult> Index()	4
4.3.2	Μέθοδος Task<IActionResult> Account()	4
4.3.3	Μέθοδος Task<IActionResult> ProfessorCourses()	4
4.3.4	Μέθοδος Task<IActionResult> RegisteredStudents(int? id)	4
4.3.5	Μέθοδος Task<IActionResult> UploadGrades(int CourseId, IFormFile userscsv)	4
4.3.6	Μέθοδος Task<IActionResult> EditGrades(int id)	5
4.3.7	Μέθοδος Task<IActionResult> EditGrades(int id, string addedGrades)	5
4.3.8	Σενάριο Εκτέλεσης	5
4.4	Secretaries Controller	5
4.4.1	Μέθοδος	5
4.4.2	Μέθοδος Task<IActionResult> Index()	5
4.4.3	Μέθοδος Task<IActionResult> Account()	5
4.4.4	Μέθοδος Task<IActionResult> UniversityCourses()	5
4.4.5	Μέθοδος IActionResult CreateCourse()	6
4.4.6	Σενάριο Εκτέλεσης	7
5	Πηγές	8



1 Εισαγωγή

Στο παρόν έγγραφο περιλαμβάνεται η τεχνική αναφορά για το



2 Ζητούμενα

Σε αυτή την εργασία στόχος είναι η δημιουργία εφαρμογής σε Visual C# (ASP.NET Core Web Application) χρησιμοποιώντας την αρχιτεκτονική σχεδίασης Model View Controller στο περιβάλλον ανάπτυξης Visual Studio 2022.

3 Επίπεδο Model

4 Επίπεδο View και Controller

4.1 Session Management

4.2 Students Controller

4.2.1 Περιγραφή των μεθόδων

- **Μέθοδος Student StudentGetter()**

Η μέθοδος StudentGetter αναζητά και επιστρέφει τον μαθητή που είναι συνδεδεμένος. Αρχικά αποθηκεύει το userid που έχουμε αποθηκεύσει στο httpcontext.session και αναζητάει τον μαθητή με το συγκεκριμένο userid. Τέλος, αποθηκεύει τον συνδεδεμένο μαθητή και συμπεριλαμβάνει μαζί τα μοντέλα courses και CourseHasStudent για να έχουμε στην πρόσβαση μας τους βαθμούς και τα ονόματα των μαθήματων του.

4.2.2 Μέθοδος IActionResult Grades(bool? asc=true)

Η μέθοδος αρχικά με την συνάρτηση StudentGetter αρχικοποιεί τον συνδεδεμένο φοιτητή. Έπειτα αποθηκεύει τα courses του μαθητή και τα ταξινομεί σε αύξουσα σειρά με βάση τα ονόματα των τίτλων των μαθήματων. Αν η παράμετρος asc είναι false τότε αποθηκεύει τα μαθήματα σε φθίνουσα σειρά. Τέλος επιστρέφει το view με το μοντέλο courses το οποίο είναι ταξινομημένο σε αύξουσα ή φθίνουσα σειρά.

4.2.3 Μέθοδος IActionResult Total()

Η μέθοδος αρχικά με την συνάρτηση StudentGetter αρχικοποιεί τον συνδεδεμένο φοιτητή. Έπειτα αποθηκεύει σε μεταβλητές τον αριθμό των δηλωμένων μαθήματων, των περασμένων μαθήματων, τον αριθμό των ECTS και την συνολική βαθμολογία (μέσο όρο). Στη συνέχεια χρησιμοποιούμε το ViewData για να εκτυπώσουμε τις μεταβλητές στο View Total. Τέλος επιστρέφει το View.

4.2.4 Μέθοδος IActionResult Account()

Η μέθοδος αποθηκεύει τον συνδεδεμένο μαθητή και επιστρέφει το View με τον μαθητή ως παράμετρο. Το View προβάλλει στον χρήστη τα στοιχεία του.

4.2.5 Μέθοδος IActionResult Index()

Αν ο χρήστης προσπαθήσει να μπει στο View του Students, η μέθοδος ανακατευθύνει τον χρήστη στο View του Account.



4.2.6 Μέθοδος IActionResult Semesters(int? page)

Η μέθοδος έχει ως παράμετρο την σελίδα/εξάμνηο που έχει πατήσει ο μαθητής να δει τα μαθήματα του. Αποθηκεύει τον συνδεδεμένο μαθητή και έπειτα αναζητάει τα μαθήματα που είναι στο ίδιο αριθμό εξάμνηο με την μεταβλητή page. Αν η μεταβλητή page είναι null τότε την ορίζουμε με τον αριθμό ένα ,δηλαδή τα μαθήματα του πρώτου εξάμνηο. Τέλος επιστρέφει το View με παράμετρο το μοντέλο CourseHasStudents.

4.2.7 Σενάριο Εκτέλεσης

Ο φοιτητής κάνει log in με τα credentials του και ανακατευθύνεται στην σελίδα του λογαριασμού του που περιέχει τα στοιχεία του. Έπειτα πατάει πάνω δεξιά στο όνομα του και κάνει κλικ στο dropdown menu που λέει My Grades. Εκεί προβάλλονται οι βαθμοί του φοιτητή ανά εξάμνηο. Αν θέλει να δει άλλο εξάμνηο πατάει πάνω στο ανάλογο radio button. Στην συγκεκριμένη σελίδα ο χρήστης μπορεί να πατήσει στο κουμπί settings και να πατήσει στο κουμπί «View Grades by Title» για να δει όλα τα μαθήματα ταξινομημένα με βάση το ονόματα των μαθημάτων. Επίσης μπορεί να πατήσει αν θέλει να προβληθούν τα μαθήματα σε αύξουσα ή φθίνουσα σειρά. Στη συνέχεια μπορεί να πατήσει στο dropdown menu την επιλογή total για να δει την συνολική του βαθμολογία. Τέλος ο φοιτητής μπορεί να κάνει log out.

4.3 Professors Controller

4.3.1 Μέθοδος Task<IActionResult> Index()

Αν ο χρήστης προσπαθήσει να μπει στο Index View του Professors , η μέθοδος ανακατευθύνει τον χρήστη στο View Account.

4.3.2 Μέθοδος Task<IActionResult> Account()

Η μέθοδος αποθηκεύει τον συνδεδεμένο καθηγητή και επιστρέφει το View με τον καθηγητή ως παράμετρο. Το View προβάλλει στον χρήστη τα στοιχεία του.

4.3.3 Μέθοδος Task<IActionResult> ProfessorCourses()

Η μέθοδος αρχικοποιεί τον συνδεδεμένο καθηγητή και αποθηκεύει τα μαθήματα που έχει ο συγκεκριμένος καθηγητής. Τέλος επιστρέφει το View με παράμετρο τη λίστα μοντέλων courses με το όνομα lessons.

4.3.4 Μέθοδος Task<IActionResult> RegisteredStudents(int? id)

Η μέθοδος έχει ως παράμετρο το id του course που έχει επιλέξει ο καθηγητής να δει. Αρχικά αποθηκεύει τον συνδεδεμένο καθηγητή και αναζητάει το μάθημα που επέλεξε. Αν δεν βρει το μάθημα σημαίνει ότι ο καθηγητής επέλεξε να δει μάθημα που δεν διδάσκει και του εμφανίζεται μήνυμα ότι δεν έχει δικαίωμα να δει το μάθημα. Έπειτα αποθηκεύει τους μαθητές που έχουν δηλώσει το συγκεκριμένο μάθημα του καθηγητή. Τέλος επιστρέφει το View με το μοντέλο Courses που περιέχει όλους τους μαθητές με το μάθημα.

4.3.5 Μέθοδος Task<IActionResult> UploadGrades(int CourseId, IFormFile userscsv)

Η μέθοδος έχει ως παραμέτρους το ID του μαθήματος και το αρχείο που περιέχει τους βαθμούς του μαθήματος. Αρχικά η μέθοδος ανοίγει το αρχείο και αποθηκεύει τα δεδομένα του σε μια μεταβλητή string. Στην μεταβλητή υπάρχουν ID μαθητών και βαθμοί. Η μέθοδος



εξετάζει για κάθε μαθητή αν υπάρχει στη βάση δεδομένων και αν υπάρχει εξετάζει αν έχει δηλώσει το μάθημα. Οι μαθητές που δεν υπάρχουν ή δεν έχουν δηλώσει το μάθημα τους αποθηκεύει σε λίστες για να εκτυπωθούν αργότερα στο View Result Report. Έπειτα αν ο φοιτητής υπάρχει και έχει δηλώσει το μάθημα τότε αποθηκεύει τον βαθμό του στην βάση δεδομένων. Τέλος αν δεν υπάρξει λάθος φοιτητής τότε η μέθοδος ανακατευθύνει τον χρήστη στο View του RegisteredStudents, ενώ αν υπάρχουν λάθος φοιτητές στο αρχείο τότε τον ανακατευθύνει στο View Result Report.

4.3.6 Μέθοδος `Task<IActionResult> EditGrades(int id)`

Η μέθοδος έχει ως παράμετρο το id του μαθήματος που έχει επιλέξει ο καθηγητής. Αρχικά αποθηκεύει τον συνδεδεμένο καθηγητή και τους φοιτητές που έχουν το μάθημα αλλά δεν έχουν βαθμό. Τέλος επιστρέφει το View με παράμετρο το μοντέλο CourseHasStudents.

4.3.7 Μέθοδος `Task<IActionResult> EditGrades(int id, string addedGrades)`

Η μέθοδος έχει ως παραμέτρους το id του μαθήματος που έχει επιλέξει ο καθηγητής και τους βαθμούς των μαθητών που έχει βάλει. Αν η μεταβλητή με τους βαθμούς είναι κενό σημαίνει ότι ο καθηγητής δεν άλλαξε τους βαθμούς και ανακατευθύνει τον καθηγητή στην σελίδα EditGrades. Έπειτα επεξεργάζεται το string με τους βαθμούς και αποθηκεύει τους νέους βαθμούς στην βάση δεδομένων. Τέλος επιστρέφει το View του EditGrades με το id του επιλεγμένου μαθήματος.

4.3.8 Σενάριο Εκτέλεσης

Ο καθηγητής κάνει log in με τα credentials του και ανακατευθύνεται στην σελίδα του λογαριασμού του που περιέχει τα στοιχεία του. Έπειτα πατάει από το dropdown menu την επιλογή My courses και ανακατευθύνεται στα μαθήματα που διδάσκει ο συγκεκριμένος καθηγητής. Σε κάθε μάθημα υπάρχει ένα κουμπί Show Students με το οποίο όταν το πατήσει δείχνει όλους τους μαθητές που έχουν δηλώσει το μάθημα και εκεί μπορεί να επεξεργαστεί τους βαθμούς των μαθητών. Επίσης μπορεί αντί να επεξεργαστεί τους βαθμούς στην ιστοσελίδα μπορεί να ανεβάσει ένα .csv με τους βαθμούς των φοιτητών. Τέλος ο καθηγητής μπορεί να κάνει log out.

4.4 Secretaries Controller

4.4.1 Μέθοδος

4.4.2 Μέθοδος `Task<IActionResult> Index()`

Αν ο χρήστης προσπαθήσει να μπει στο View του Secretaries, η μέθοδος ανακατευθύνει τον χρήστη στο View Account.

4.4.3 Μέθοδος `Task<IActionResult> Account()`

Η μέθοδος αποθηκεύει τον συνδεδεμένο γραμματέα και επιστρέφει το View με τον γραμματέα ως παράμετρο.

4.4.4 Μέθοδος `Task<IActionResult> UniversityCourses()`

Η μέθοδος αποθηκεύει τα μαθήματα ταξινομημένα σε αύξουσα σειρά ως προς το εξάμηνο και επιστρέφει το View με το μοντέλο Courses σε λίστα ως παράμετρο.



4.4.5 Μέθοδος IActionResult CreateCourse()

Η συγκεκριμένη συνάρτηση χρησιμοποιείται για να προβάλει στον χρήστη μια φόρμα δημιουργίας μαθήματος.

Μέθοδος Task<IActionResult> CreateCourse([Bind("CourseId,Title,Semester,ProfessorId")] Course course)

Η μέθοδος χρησιμοποιείται για εισαχθεί το νέο μάθημα στη βάση δεδομένων. Αν δεν έχουν εισαχθεί ακόμα μαθήματα το courseId παίρνει την τιμή 1, ενώ αντίθετα παίρνει την τιμή courseId του τελευταίου μαθήματος στην λίστα συν ένα. Αν τα δεδομένα που έχει γράψει ο γραμματέας είναι έγκυρα τότε εισάγονται στην βάση δεδομένων και η μέθοδος ανακατευθύνει τον χρήστη στο View του UniversityCourses. Αν ο χρήστης δεν έχει αναθέσει καθηγητή στο μάθημα, μπορεί να αναθέσει καθηγητή αργότερα.

Μέθοδος Task<IActionResult> AssignProfessor(int id)

Η μέθοδος χρησιμοποιείται για να προβάλει στον χρήστη μια φόρμα με την οποία μπορεί να αναθέσει καθηγητή σε μάθημα που δεν έχει δηλωμένο καθηγητή.

Μέθοδος Task<IActionResult> AssignProfessor(int id, [Bind("CourseId,Title,Semester,ProfessorId")] Course course)

Αν το Id που δόθηκε δεν είναι το ίδιο με το id μαθήματος που δεν έχει υπεύθυνο καθηγητή εμφανίζει στον χρήστη ότι δεν βρέθηκε το μάθημα. Έπειτα προσθέτει τον καθηγητή στο μάθημα που δεν έχει, όμως αν το id που δόθηκε δεν υπάρχει θα πεταχτεί exception και θα εμφανιστεί στον χρήστη ανάλογο μήνυμα.

Μέθοδος Task<IActionResult> UniversityStudents(int? page, int? pageSize, string? search, string? searchParam, string? sortOrder)

Η μέθοδος χρησιμοποιείται για να προβάλει στον χρήστη όλους τους φοιτητές του τμήματος. Ο κώδικας περιλαμβάνει pagination για να προβάλει στον χρήστη τους φοιτητές σε ξεχωριστές σελίδες και όχι όλους μαζί.

Μέθοδος IActionResult CreateStudent() Η μέθοδος χρησιμοποιείται για να προβάλει στο χρήστη φόρμα δημιουργίας μαθητή.

Μέθοδος Task<IActionResult> CreateStudent([Bind("RegistrationNumber,Name,Surname,Department,U Student student)

Η μέθοδος χρησιμοποιείται για εισαχθεί ένας νέος φοιτητής στη βάση δεδομένων. Αν δεν έχουν εισαχθεί ακόμα φοιτητές το student ID παίρνει την τιμή 1, ενώ αντίθετα παίρνει την τιμή Student ID του τελευταίου φοιτητή στην λίστα συν ένα(το ίδιο ισχύει για το userId του φοιτητή). Αν τα δεδομένα που έχει γράψει ο γραμματέας είναι έγκυρα τότε δημιουργείται ένα νέο model User με τα στοιχεία που έχει πληκτρολογήσει ο γραμματέας ,ενώ εισάγονται στο model Student τα ίδια στοιχεία και αποθηκεύονται στην βάση δεδομένων.

Μέθοδος Task<IActionResult> StudentDetails(int? id, int? semester, string? sortOrder)

Η μέθοδος χρησιμοποιείται για να προβάλει τις πληροφορίες του φοιτητή που επέλεξε. Η μέθοδος αρχικά αναζητάει και αποθηκεύει με βάση το id που δόθηκε στη παράμετρο τον φοιτητή. Αν το μοντέλο είναι null σημαίνει ότι δεν βρήκε φοιτητή με το συγκεκριμένο id και εμφανίζει στον χρήστη ανάλογο μήνυμα. Έπειτα αποθηκεύει στο ViewData τα δηλωμένα μαθήματα και τα ETCS του φοιτητή για να τα προβάλει .Τέλος επιστρέφει το View με παράμετρο το μοντέλο Student.

Μέθοδος Task<IActionResult> StudentAssignCourses(int? id)

Η μέθοδος αναζητάει και αποθηκεύει τον φοιτητή που έχει το ίδιο id με την παράμετρο. Αν δεν βρεθεί φοιτητής με το ίδιο id θα εμφανιστεί ανάλογο μήνυμα. Έπειτα αναζητάει τα μαθήματα του φοιτητή και αποθηκεύει τα μαθήματα που δεν έχουν δηλωμένο βαθμό για να προβληθούν στο ανάλογο View. Τέλος επιστρέφει το View με τον φοιτητή ως παράμετρο.

Μέθοδος Task<IActionResult> StudentAssignCourses(int id, string selectedCourses)

Η μέθοδος αρχικά αναζητάει και αποθηκεύει τον φοιτητή που έχει το ίδιο id με την παράμετρο. Αν δεν βρεθεί φοιτητής με το ίδιο id θα εμφανιστεί ανάλογο μήνυμα. Αν δεν



έχουν εισαχθεί ακόμα βαθμοί στον φοιτητή το grade ID παίρνει την τιμή 1, ενώ αν υπάρχουν μαθήματα παίρνει την τιμή Grade ID του τελευταίου μαθήματος στην λίστα συν ένα. Έπειτα επεξεργάζεται την παράμετρο selected Courses και αποθηκεύει τα μαθήματα και βαθμούς στην βάση δεδομένων. Τέλος επιστρέφει το View StudentDetails που προβάλλει τις λεπτομέρειες του φοιτητή.

Μέθοδος Task<IActionResult> UniversityProfessors()

Η μέθοδος χρησιμοποιείται για να προβάλλει στον χρήστη όλους τους καθηγητές του τμήματος

Μέθοδος IActionResult CreateProfessor()

Η μέθοδος χρησιμοποιείται για να προβάλλει στο χρήστη φόρμα δημιουργίας καθηγητή.

Μέθοδος Task<IActionResult> CreateProfessor([Bind("ProfessorId,Afm,Name,Surname,Department,UserId")] Professor professor)

Η μέθοδος χρησιμοποιείται για εισαχθεί ένας νέος καθηγητής στη βάση δεδομένων. Αν δεν έχουν εισαχθεί ακόμα καθηγητές το professor ID παίρνει την τιμή 1, ενώ αντίθετα παίρνει την τιμή Professor ID του τελευταίου καθηγητή στην λίστα συν ένα(το ίδιο ισχύει για το userId του καθηγητή). Αν τα δεδομένα που έχει γράψει ο γραμματέας είναι έγκυρα τότε δημιουργείται ένα νέο model User με τα στοιχεία που έχει πληκτρολογήσει ο γραμματέας ,ενώ εισάγονται στο model Student τα ίδια στοιχεία και αποθηκεύονται στην βάση δεδομένων.

Μέθοδος Task<IActionResult> ProfessorDetails(int? id, string? sortOrder)

Η μέθοδος χρησιμοποιείται για να προβάλλει τις πληροφορίες του καθηγητή που επέλεξε ο γραμματέας. Η μέθοδος αρχικά αναζητάει και αποθηκεύει με βάση το id που δόθηκε στη παράμετρο τον καθηγητή. Αν το μοντέλο είναι null σημαίνει ότι δεν βρήκε καθηγητή με το συγκεκριμένο id και εμφανίζει στον χρήστη ανάλογο μήνυμα. Έπειτα αποθηκεύει στο ViewData τα μαθήματα που δεν έχουν καθηγητή δηλωμένο και τα μαθήματα που διδάσκει ο καθηγητής για να τα προβάλλει .Τέλος επιστρέφει το View με παράμετρο το μοντέλο Professor.

Μέθοδος Task<IActionResult> AssignProfessorCourse(int courseid, int professorid)

Η μέθοδος αναζητάει και αποθηκεύει τον καθηγητή που έχει το ίδιο id με την παράμετρο. Αν δεν βρεθεί καθηγητής με το ίδιο id θα εμφανιστεί ανάλογο μήνυμα. Έπειτα αναζητάει το μάθημα που δόθηκε στην παράμετρο και προσθέτει στο μάθημα ως professor id το id του καθηγητή. Τέλος επιστρέφει το View με τον φοιτητή ως παράμετρο. Τέλος ανακατευθύνει τον χρήστη στο View ProfessorDetails.

4.4.6 Σενάριο Εκτέλεσης

Ο γραμματέας κάνει log in με τα credentials του και ανακατευθύνεται στην σελίδα του λογαριασμού του που περιέχει τα στοιχεία του. Έπειτα πατάει στο dropdown menu την επιλογή University Students και ανακατευθύνεται σε νέα σελίδα που περιέχει όλους φοιτητές του τμήματος. Εκεί μπορεί να αναζητήσει τον μαθητή που επιθυμεί και να ταξινομήσει τους μαθητές σε αύξουσα ή φθίνουσα σειρά με βάση το Registration Number, όνομα, επίθετο ή και τμήμα. Σε κάθε μαθητή μπορεί να πατήσει το κουμπί more details που θα εμφανίσει περισσότερες πληροφορίες για τον μαθητή και επίσης μπορεί να δηλώσει νέα μαθήματα στον μαθητή. Στη συνέχεια πατάει στο dropdown menu την επιλογή University Professors και ανακατευθύνεται σε νέα σελίδα που περιέχει όλους καθηγητές του τμήματος. Εκεί μπορεί να αναζητήσει τον καθηγητή που επιθυμεί και να ταξινομήσει τους καθηγητές σε αύξουσα ή φθίνουσα σειρά με βάση το όνομα ή επίθετο. Σε κάθε μαθητή μπορεί να πατήσει το κουμπί more details που θα εμφανίσει περισσότερες πληροφορίες για τον καθηγητή και επίσης στην νέα σελίδα μπορεί να δηλώσει νέα μαθήματα στον καθηγητή. Τέλος πατάει την επιλογή University Courses και ανακατευθύνεται σε νέα σελίδα με όλα τα μαθήματα του πανεπιστημίου. Στην σελίδα αυτήν μπορεί να αναζητήσει μάθημα και να τα ταξινομήσει με βάση το όνομα του μαθήματος. Επίσης μπορεί να πατήσει το κουμπί Add Course για να



εισάγει νέο μάθημα ή να πατήσει σε κάποιο από τα μαθήματα το κουμπί more details για να δει τα στοιχεία του μαθήματος αλλά και να δηλώσει καθηγητή στα μαθήματα που δεν έχουν καθηγητή.

5 Πηγές

- Βίντεο για την δημιουργία ενός RecyclerView.
<https://youtu.be/Mc0XT58A1Z4>
- Βίντεο για την υλοποίηση του OnClickEvent σε items του RecyclerView.
<https://youtu.be/7GPUpvcU1FE>
- Android Documentation για την δημιουργία μενού στην μπάρα της εφαρμογής.
<https://developer.android.com/develop/ui/views/components/appbar/actions>
- Βίντεο για την δημιουργία κουμπιού με progress bar.
<https://youtu.be/zv9R5EcRKHM>
- Android Documentation για την δημιουργία του File Provider.
<https://developer.android.com/reference/androidx/core/content/FileProvider>