

Употреба штампане литературе и комуникационих уређаја није дозвољена и строго је кажњива током трајања испита.

ПРВИ ДЕО ИСПИТА (30 минута)

1. [6] Жели се направити елементарна *CodeIgniter* апликација која приказује једну од 10 пасивних HTML страница тако што корисник из браузера задаје URL путању:

[sajt_aplikacije]index.php/pages/view/N

Где је N број странице. Навести све фајлове које треба креирати или мењати (име фајла са релативном путањом у оквиру *CodeIgniter* пројекта, као и садржај сваког фајла (за пасивне HTML странице пример једне од њих). Пасивне странице држе се на диску на одговарајућем месту у оквиру *CodeIgniter* пројекта. Нема приступа бази нити су потребни заједнички хедери ни футери на страницама.

2. [4] Програмер је пре неког времена направио локални *GIT* репозиторијум за свој пројекат, радио на пројекту и онда се после вишедневне паузе вратио на исти пројекат.

Потребно је да провери да ли је све промене убацио у репозиторијум и ако није да ажурира репозиторијум. Претпоставимо да има и фајлова које је већ користио и изменио само у радном фолдеру, затим потпуно нових фајлова које је први пут креирао у радном фолдеру, као и потпуно нових фајлова које жели експлицитно да изостави из репозиторијума (а да и даље егзистирају у радном фолдеру). Написати низ *GIT* команди којима се постижу описане акције.

3. [6] За свако од следећих тврђења навести да ли је истинито - одговорити са (ДА/НЕ):

а) У моделу водопада планирање посла за одређену активност је ограничено расположивим временом (енгл. *timeboxed*).

б) У *Scrum*-у, *product backlog* представља колекцију задатака који су били планирани за текући спринт, али нису завршени.

в) Процеси на *data flow* дијаграму (*DFD*) одвијају се у редоследу растућих редним бројевима процеса.

г) Двојни (*semidetached*) тип *COCOMO* односи се на системе као што су обрачун платних спискова, системи за попис инвентара итд.

д) Кључна разлика између шаблона *Row Data Gateway* и *Active Record* је што се инстанца *Row Data Gateway* односи на једну врсту релационе табеле у бази, док се инстанца *Active Record* односи на једну колону релационе табеле у бази.

ђ) Предмет формалне инспекције може бити техничка документација и изворни програмски код пројекта.

Напомена: сваки исправан одговор носи +1 поен, неисправан -0.5, а неодговорена ставка 0 поена. Задатак у збиру не носи негативне поене.

Употреба штампане литературе и комуникационих уређаја
није дозвољена и строго је кажњива током трајања испита.

Име и презиме:	Број индекса:
----------------	---------------

ДРУГИ ДЕО ИСПИТА (150 минута)

4. Реализовати мањи веб систем за дипломце ЕТФ-а у коме треба да се води евиденција о одрађеним стручним праксама и дипломским радовима. Корисници система су студенти, наставници и представници компанија. Студенти и наставници имају своје корисничко име, лозинку, име, презиме, датум рођења и ЈМБГ. Студент додатно има податке о броју индекса и одсеку на коме студира, а наставник – титулу (мр / др), звање (редовни проф. / ванредни проф. / доцент / асистент) и катедра којој припада. И одсек и катедра имају своје називе. Одсек може да припада једној или две катедре, односно катедра може имати један или више одсека којим руководи.

Студент пријављује дипломски рад код одређеног ментора. Ментор је наставник у звању доцента или ванр. или редовног професора). Дипломски рад има наслов и везује се за неки предмет, који је студент положио на студијама. Када дипломски рад одбрани, у евиденцији се додатно чувају оцена са дипломског рада, датум одбране и састав комисије (комисију чине ментор + један или два других наставника). Оцена може да буде између 6 и 10. Дипломски рад не може да буде пријављен док се не уради и потврди стручна пракса.

Стручну праксу расписују представници компанија. Они се на систем логују кроз засебну веб форму у односу на студенте и наставнике. Компанија има свој назив, адресу, ПИБ, жиро-рачун. Један представник је задужен по компанији за рад са праксама. Стручна пракса се расписује за одређену позицију (нпр. јуниор софтверски инжењер), временски период и има одређени број слободних места за ту позицију. Када се студент попуни пријаву за праксу, добија статус 1. Одбијена пријава има статус 0, а прихваћена 2. Када студент заврши праксу, након краја временског интервала праксе, представник компаније може мењати статус из 2 у 3. На крају, наставник који је задужен за прихватање стручне праксе, може тај статус да промени у 4, чиме признаје урађену праксу студенту и омогућава му да пријави дипломски рад. Једна компанија у једном временском периоду може имати више расписаних конкурса за стручне праксе.

- а) [14] На нивоу релационе базе података, дате у *SQL SistemZaDiplomce2018.sql*, утврдити да ли је база и да ли су подаци у бази у складу са горе описаном спецификацијом. Урадити све измене, како би база података одговарала овим корисничким захтевима. Модификовану базу података попунити подацима, неопходним за тестирање.

Излаз задатка 4а):

скрипт модификоване базе података *SistemZaDiplomce2018- GGGGBBBB.sql*

- b) [14] У алату *NetBeans IDE*, реализовати *MVC (Model-View-Controller)* апликацију коришћењем програмског језика *PHP*. Имплементацију је могуће радити коришћењем објектно-оријентисаног *PHP* кода или коришћењем *CodeIgniter* или *Laravel* оквира. Потребно је модификовати логовање корисника (за сада може да се логије само Студент) и имплементирати искључиво део система који се односи на логовање Студента на систем и пријављивање дипломског рада, као и логовање Наставника на систем и евидентирање оцене и комисије на већ пријављеном дипломском раду.

Када се студент успешно улогује, он треба да у форми за пријаву дипломског рада унесе наслов, предмет (из падајуће листе оних предмета које је слушао) и ментора (из падајуће листе свих оних са његовог одсека). Када потврди пријаву дипломског рада, таква пријава треба да се појави у листи дипломаца наставнику кога је одабрао као ментор. Одабиром одређеног дипломаца, ментор добија форму где уноси оцену и члана/чланове комисије. Датум одбране треба да буде датум када се уноси у систем (када ментор потврђује форму).

Излаз задатка 4б): пројекат *PHP* имплементације описаног дела апликације (уколико се ради у радном оквиру, прекопирати такође читав пројекат)

- c) [8] Нацртати *WAE* дијаграм овог веб система. Дијаграм треба да приказује све серверске компоненте (*M,C*) и клијентске компоненте (*V*) и *HTML* форме. Део имплементације који буде реализован у тачки б) треба да буде у складу са тачком ц).

Излаз задатка 4ц): нацртати реализован дијаграм на додатном папиру (ако дијаграм није у складу са тачком б) и са *MVC* узорком, не може се остварити макс број поена!)

- d) [8] Нацртати дијаграм секвенце којим се приказује промена статуса захтева за стручну праксу, од тренутка аплицирања на стручну праксу од стране студента (статус 1) до тренутка прихватања праксе од стране наставника (статус 4). У дијаграму обухватити и све друге сценарије који су описани спецификацијом задатка. У дијаграм укључити све актере (студент, представник компаније, наставник) и приказати која компонента/слој апликације се позива.

Излаз задатка 4д): нацртати реализован дијаграм на додатном папиру.

Име и презиме:	Број индекса:
----------------	---------------

Решење задатка 4ц и 4д: