

I. Изисквания за оформяне на проектите за практиките

(Проект: „Подготовка на специалисти по информационни технологии в икономиката на знанието“)

ЦЕЛТА НА ПРОЕКТИТЕ е да създаде умения на студентите да работят по в среда, близка до фирмената, за да се научат да прилагат изучаваното в университетските курсове, както и да набавят самостоятелно допълнително необходимите им знания за неговото изпълнение. По този начин студентите ще натрупат известен опит, позволяващ им по-лесен преход от университета към тяхното кариерно развитие.

Студентите получават задача за разработване на проект, проучат наличната литература и разработват продукт в зависимост от конкретното задание. Проектите са групови (5-6 студенти). Разработеният продукт (софтуерен продукт, мрежова конфигурация, математически модел, софтуерна конфигурация, осигуряваща сигурността на данни или компютърна система и други) трябва да е резултат от самостоятелно проучване и разработване на членовете на групата. Всяка група избира свой мениджър измежду членовете си. Групата има супервайзор (преподавател или фирмен специалист), който осигурява периодични консултации и направлява работата по проекта в правилната посока. Проектът е ориентиран към 30 часа работа на човек.

КОМПОНЕНТИ НА ПРОЕКТА:

- **Описание на заданието**

Целта на този проект е **да се създаде система за сравнение на банкови продукти**

- 1) Уеб сайт с каталози на банкови продукти, подобни на каталозите на www.moitepari.bg – като каталога на депозити <http://www.moitepari.bg/depoziti/katalog.aspx>
- 2) Десктоп приложение, в което да се попълват конкретните банкови продукти.

Функционалности:

Списъкът с банковите продукти трябва да отговаря на този на www.moitepari.bg, но да се реализира само бизнес логика на депозитите.

Депозитите трябва да имат следните основни параметри, по които се прави търсенето <http://www.moitepari.bg/depoziti/depoziti.aspx> (разширеното търсене)

Сайтът трябва да съдържа каталог, подобен на <http://www.moitepari.bg/depoziti/katalog.aspx>, но вместо бутон „ЗАЯВИ“ да има бутон „Изчисли“, който да води до страница с разплащателен план за конкретния продукт като този

<http://www.moitepari.bg/depoziti/razplashtatelen-plan.aspx?s=QlpoNDFBWSZTWZlXHS4AAB0%2fgAADgAJ%2f8CcgTQAvI94gIAB0liZMgyD1DQYhpo9QqeRommTRp6gGgAKWUYWXScvQlwUhorqcagSRiUa5PSaGgTxCAwnZLb70TuZDYyInUqFrQILQhq8%405CaJTIBX6LCDdAUnS0gETdLS6P4u5lpwoSEKjpc&view=Results>

Тази страница трябва да позволява изчисление с всички параметри от <http://www.moitepari.bg/depoziti/depoziti.aspx>, попълнени според особеностите на банковия продукт. Ако конкретният продукт е само в лева, то параметърът „валута“ да бъде забранен (както е на примера), както и други, които са фиксирани за продукта. Параметрите на изчисление трябва да са според зададените от продукта опции. Ако продуктът е за суми от 5000 до 100 000 например, да връща съобщение за грешка при опит за изчисление извън тях (това важи за всички параметри)

Десктоп приложението трябва да позволява попълване на данни за различните банкови продукти, като другите продукти освен депозити да са просто имена, т.е. да

позволява разширение. Депозитите да се попълват според полетата, които се показват в детайли, както тук

<http://www.moitepari.bg/depoziti/detaili.aspx?s=QlpoNDFBWSZTWbU4hlgAABu%2fgAADgAJ%2f8CxbTQA%2fI94glAB0liYQBoaA0zUNNHqFT0mj0TIYINMQAFCjJuEk1rWVhiMgUzIOdTITiKGgV5wSSc52gjb714QUPBkcWsbWbDhBkUxKI33QSQ0skQNo9baQOQCjpYQNxMM6dYfi7kinChIWpxDLAA%3d%3d>

Името на продуктите в каталога трябва да води към такава страница в детайли
Попълването на банковите продукти трябва да генерира редове в MySQL таблици, като е имплементирана само за депозити, другите да съществуват само по име. Бизнес логиката, която трябва да е в отделна от уеб сайта библиотека трябва да взима данните от тази таблица и спрямо тях да изчислява резултатите, които трябва да се показват в разплащателния план (ЕГЛ, нетно изплатени, данъци и т.н.) Изчисленията трябва да бъдат открити от разработчиците чрез сравняване с разплащателните планове и детайлите на сайта.

Системата има три компонента:

1. Десктоп приложение за попълване на банкови продукти, с имплементирани депозити, което генерира записи в база от данни MySQL
2. Бизнес логика, която изчислява лихви, плащания и др за всеки продукт според запис в базата от данни и параметрите, с които е зададено да се изчисли, ако са спазени ограниченията (зададени в базата данни)
3. Уеб сайт, който да е responsive и да съдържа каталози на продукти, с имплементирани депозити, също както на дадения линк, с бутон „Изчисли“, вместо „Заяви“, водещ към разплащателен план и със страници за детайли (името на продукта да води към детайлите)
4. Реализираният продукт да има попълнени 5-10 примерни депозита, моделирани според наличното в <http://www.moitepari.bg/depoziti/depoziti.aspx>

- **Условия за изпълнение на заданието**

Да се напише на C# и JQuery / или подходяща Javascript рамка („framework“) / в средата .NET – ASP.NET проект, създаден с Net.Core 2.2 **или** .NET Да се използват HTML5 и CSS3

Подробната документация на проекта да се приложи и като контекстуална помощ в рамките на приложението

Студентите да разпределят ролите си – архитект, дизайнер, софтуерен инженер, експерт документация, тестер – 5 души.

Бизнес логиката на калкулаторите трябва да бъде изведена експериментално, чрез тестване на трите калкулатора, посочени в заданието.

Тестерът трябва да състави план за тестовете и да предостави като резултат диаграми и формули, които софтуерният инженер и архитектът да имплементират. Той трябва да напише скриптове за автоматизирани тестове и да обобщи статистическите резултати от тяхната работа – какви грешки са били открити, на кой етап от разработката, статистика на представяне на калкулатора при тестване на натоварване със заявки (то варира с всеки тест) и др.

- **Изисквания към решението (представени резултати)**

Решението, което се представя от групата, се състои от две основни части:

1. Работещ продукт, например функциониращ софтуер, написан на съответния компютърен език и платформа, или мрежова конфигурация, която или е

реализирана физически, или е доказана нейната функционалност чрез симулационен софтуер.

2. Документация, която подробно и разбираемо описва представеното решение. Ако някои от двата компонента не са представени, на студента не се позволява да премине успешно проекта.

- **Изисквания към продукта**

(за конкретния случай на компютърна програма)

1. Компютърната програма трябва да е реализирана на съответните компютърен език и платформа, посочени от преподавателя.
2. Компютърната програма трябва да се компилира или интерпретира от съответния компилатор или интерпретатор без грешки. Програми, които не се компилират или интерпретират не се приемат.
3. Програмният код трябва да е написан в съответствие със стандартите за оформяне на програмен код на съответния език. Код, който не е форматиран (например, ако всички редове са подравнени от лявата страна на екрана) намалява крайната оценка.
4. Програмният код трябва да е подробно документиран в съответствие със стандартите на конкретната технология. Например, ако продуктът е реализиран на Java, кодът трябва да е документиран, използвайки Javadoc стандартите.

- **Изисквания към документацията**

1. Представената документация трябва да съдържа разбираемо и подробно описание на решението на конкретната задача. Програмният код не трябва да е част от документацията, а отделно приложение.
2. Документацията трябва да бъде представена в системата за електронно обучение Moodle в PDF формат. Всеки друг текстов файлов формат, особено затворените като DOC и DOCX не се приемат.
3. Документацията започва със сигнатурата на проекта, заглавието на проекта, името на студента, факултетния номер и семестъра.
4. Документацията съдържа номера и текста на поставената задача.
5. Текстът трябва да е оформен в съответствие със стандартите за оформление на курсови и дипломни работи. Неформатиран текст, текст, съдържащ графики с лошо качество, няма да бъде приет.

- **Междинни и крайни срокове за изпълнение**

До края на пролетния семестър

- **Дати за консултации**

По заявка (в приемно време)

- **Дати за защита на проектите**

Допълнително указани в системата Moodle

- **Система за оценяване на резултатите**

Всеки студент се оценява в съответствие с общото качество на продукта и качеството на неговия компонент, за което отговаря той в избраната от него/нея роля. Критерият за оценка е:

Процент на изпълнение, 2/3 се формират от личната роля на студента и 1/3 от процента на изпълнение на цялостния продукт	Оценка 2-6
61-70%	3
71-80%	4

81-90%	5
91-100%	6

- **Указания за управление на проекта**

Реализирането на продукта да се направи по метода Agile, като се реализира цялостната система първо с ограничена функционалност (само най-прости изчисления), но да позволява разширение при последващи цикли. За успешно справяне със задачата се приемат частични реализации на функционалностите на депозитите, но при напълно завършена система, при която да се попълват продукти и да се показват, заедно с изчисленията.

Бизнес логиката трябва да се третира като black box и чрез тестване да се разпознае и възпроизведе, което е едно от основните предизвикателство пред разработчиците.

I. Изисквания към курсовите работи (проекти) към даден курс

Груповите проекти към курсовете, изискващи разработване на продукт, софтуер, мрежова конфигурация, математически модел, софтуерна конфигурация, осигуряваща сигурността на данни или компютърна система и др., се оформят както в I.

Проектите към курсове, касаещи обзор на определена тема, без разработването на конкретен продукт се оформят по схемата, дадена по-долу:

КОМПОНЕНТИ НА ПРОЕКТА:

- **Описание на заданието**
 - **Условия за изпълнение на заданието**
 - **Изисквания към обзора/анализа**
 - **Изискване за представените резултати**
1. Курсовата работа трябва да бъде оформена като текстов документ, съгласно стандартите за курсова и дипломна работа.
 2. Да съдържа конкретна и строга формулировка на задачата, която ще бъде разгледана.
 3. Да представя разглеждането на задачата в контекста на съответната предметна област.
 4. Да представи други решения на разглежданата задача, ако има известни такива, заедно с подходящ техен анализ.
 5. Да съдържа строго изложение на конкретното решение на проблема.
 6. Да съдържа изчерпателен и съдържателен списък от литературни източници с цитирания в текста.