SOB 2024 Proposal

SQLite Database for libbitcoin

SQLite Database for libbitcoin

Libbitcoin

Shashank Shekhar Singh

Name:	Shashank Shekhar Singh
Major:	Mechanical Engineering
Degree:	Bachelor of Technology
Year:	Sophomore
University:	Indian Institute of Technology (BHU), Varanasi
GitHub:	@Shashankss1205
College Mail:	shashankshekhar.singh.mec22@itbhu.ac.in
Private Mail:	shashankshekharsingh1205@gmail.com
Phone:	(+91) 8303130207
LinkedIn:	Profile
Resume:	Link
Nationality:	India
Address:	SH 1/1A -1-12, Unchawa Lodge, Gilat Bazar, Varanasi,UP
Time zone:	Indian Standard Time (UTC +5:30)

About Me

Hi! I am Shashank Shekhar Singh, currently pursuing my sophomore year in Mechanical Engineering at the esteemed Indian Institute of Technology BHU (Varanasi), India. I have delved deeply into the realms of programming and its applications in the world of Bitcoin and blockchain technology. I have honed my skills in Bitcoin, blockchain technology, model selection, development, and deployment. My skills include programming in C, C++, Python, Rust, and SQLite as well as competitive programming. Additionally, as a Machine Learning Engineer, my passion lies in leveraging machine learning algorithms to explore the intricacies of Bitcoin and blockchain systems. I love open-source development and building scalable solutions. This has motivated me to contribute to the Libbitcoin organization.

My Personal Projects

1. **Visual Question Answering System | Inter IIT Preparation | Kaggle**
- Constructed a VQA model utilizing Keras and TensorFlow frameworks.
- Utilized NLP components such as stemming, tokenization, and Google News bin pretrained weights.
- Incorporated the VGG16 Model for ImageNet CNN for computer vision tasks.

2. **Dark Pattern Detection with Transformer-based Models | Dark Pattern Buster Hackathon' 2023 | GitHub**
- Engineered a Python script leveraging fine-tuned transformer-based models (BERT, XLNet, RoBERTa) to detect and categorize dark patterns in textual data.
- Extended functionality with a web plugin for visual highlighting of deceptive design elements, enhancing user interface analysis and empowerment tools.

3. **NLP-based Question Answering System for Tabular Data | NLP.py | GitHub**
- Developed a web interface using Vanilla JavaScript.
- Implemented a database backend using SQLite and integrated a prompting NLP tool, GPT-3 API.
- Engineered a pipeline for question input, database injection, and SQL generation for streamlined operation.

My Open-Source Contributions

1. **Gravitational Lensing Image Generation using DDPM Models | Open-Source Project | ML4SCI | GitHub**

- Developed a Denoising Diffusion Probabilistic Model in TensorFlow and Keras for high fidelity Image generation integrated with Gaussian Diffusion approach.

2. **Gravitational Lensing Image Classification using Auto Encoder-Decoder | Open-Source Project | ML4SCI | GitHub**

- Developed an **Auto Encoder-Decoder Model** in TensorFlow and Keras for high accuracy image classification using the **ResNet50** model coupled with **Gaussian and Rotational augmentation** to achieve the same.

3. **Text Detection and Recognition with CRAFT | Open-Source Project | HumanAI | GitHub**

- Developed software utilizing the Character-Region Awareness for Text detection (**CRAFT**) in PyTorch for precise text bounding box detection.
- Integrated CNN-RNN and transformers for accurate word recognition capability.

4. **Interpolation to find unknown values in space exploration | Open-Source Project | OpenAstronomy | GitHub**

- Integrated **linear** and **polynomial interpolation**, **PyTorch-based neural networks (ML/DL)**, and **support vector regression model** for accurate prediction of flux measurements at desired wavelengths.

Project Abstract

This project aims to develop libbitcoin-database-sqlite, an SQLite-based database for libbitcoin. Leveraging my C++ and SQL expertise, I will integrate SQLite while maintaining compatibility with the existing libbitcoin-database API. The project includes:

- Understanding libbitcoin-database usage within libbitcoin.
- Implementing the provided SQL schema and API-fulfilling queries.
- Creating comprehensive unit tests using the boost test suite.
- Proposing API improvements for smoother integration and potential performance gains.

This project strives to create a user-friendly alternative for resource-constrained machines.

Project Goals

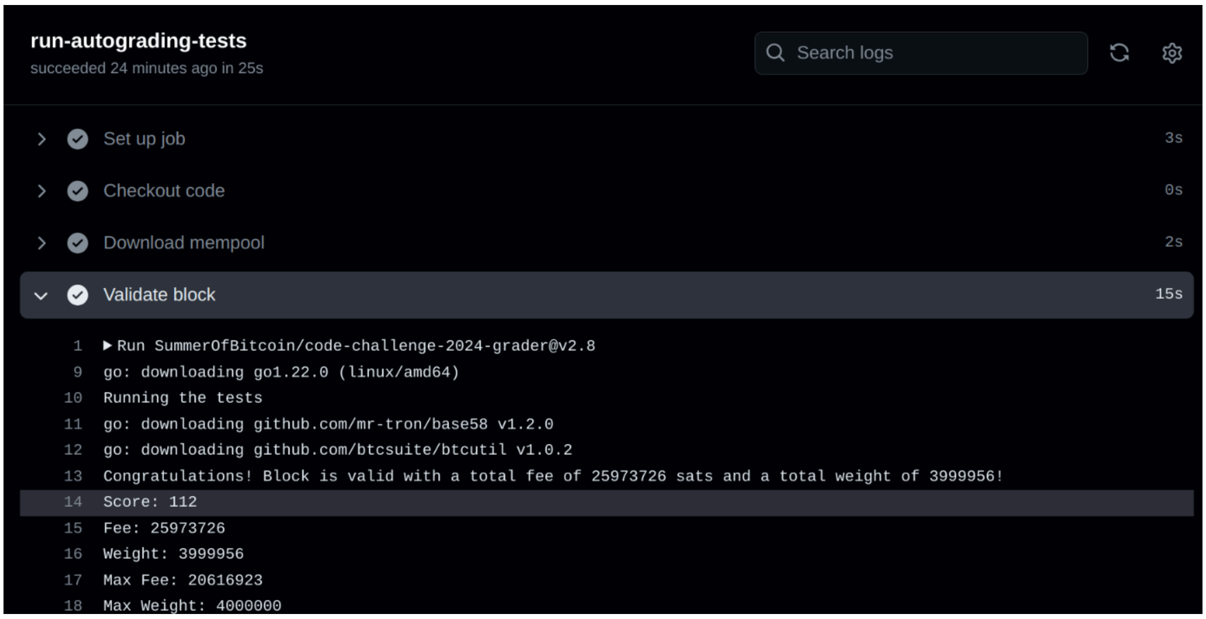
1. **API Understanding:** Thoroughly comprehending the libbitcoin-database API and its usage within other libbitcoin components. This includes analyzing function calls, data structures, and interaction patterns.

2. **SQLite Database Development:**
    - **Schema Implementation:** Creating an SQLite schema that accurately reflects the existing libbitcoin-database schema, ensuring data compatibility.
    - **API Query Implementation:** Developing SQLite queries that fulfill the functionalities exposed by the libbitcoin-database API. These queries should efficiently retrieve and manipulate data within the SQLite database.

3. **Testing and Validation:**
    - **Test Suite Creation:** Utilizing the boost test suite, familiar to the libbitcoin project, to create comprehensive unit tests for all implemented functions. These tests will ensure the correctness and reliability of `libbitcoin-database-sqlite`.

4. **API Improvement Updates :**
    - Analyzing the development process and identifying potential modifications to the libbitcoin-database API that could streamline integration with `libbitcoin-database-sqlite`. This may involve reducing complexity or introducing features specifically suited for an SQLite backend.

SOB Assignment



```
▶ Run SummerOfBitcoin/code-challenge-2024-grader@v2.8
go: downloading go1.22.0 (linux/amd64)
Running the tests
go: downloading github.com/mr-tron/base58 v1.2.0
go: downloading github.com/btcsuite/btcutil v1.0.2
Congratulations! Block is valid with a total fee of 25973726 sats and a total weight of 3999956!
Score: 112
Fee: 25973726
Weight: 3999956
Max Fee: 20616923
Max Weight: 4000000
```

Implementing the complete code by myself, I was able to achieve 112 score in the assignment test given in SOB.

Competency Test

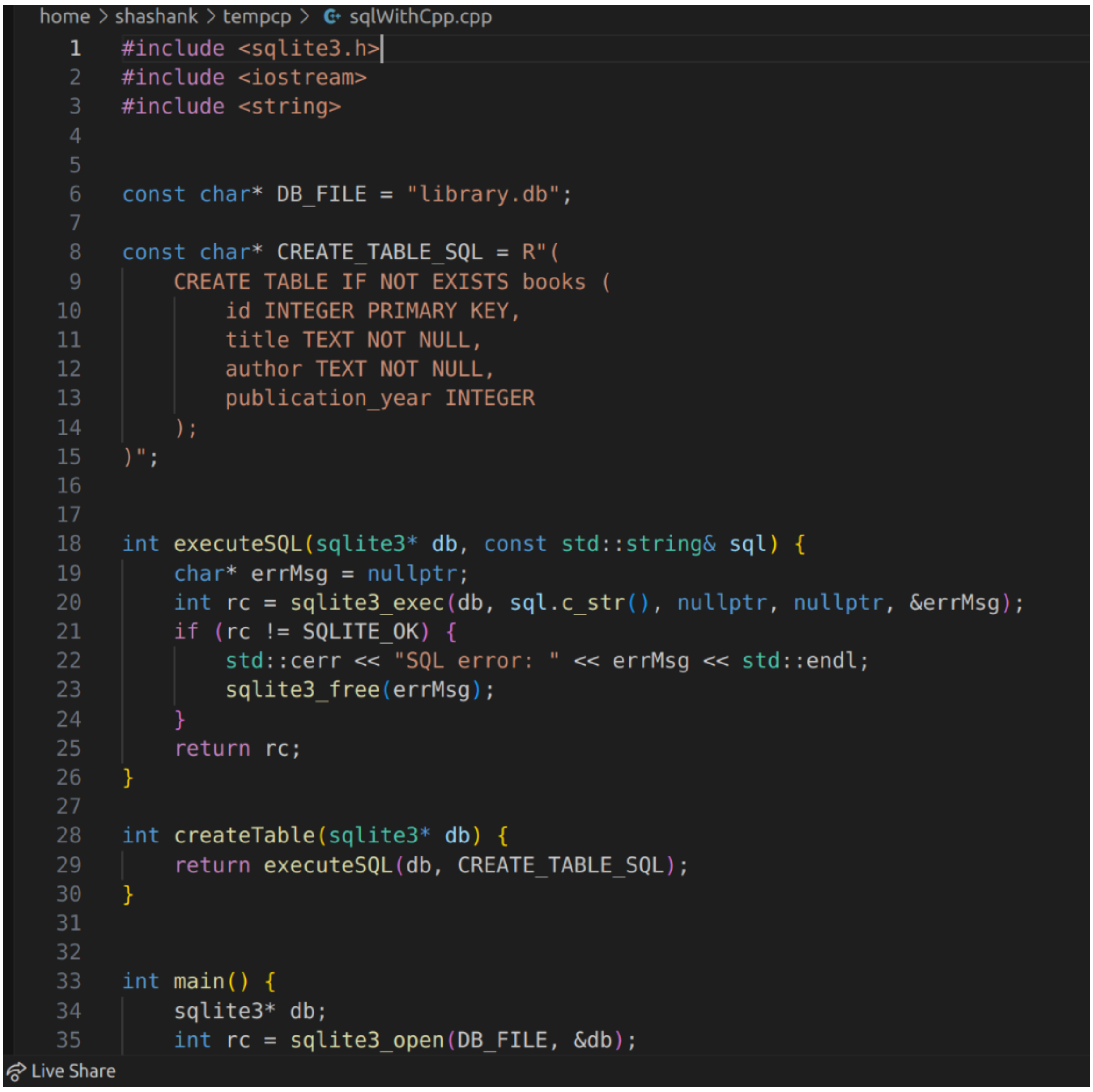
Libbitcoin-database Setup (Version 3):

1. Following the instructions provided in the README file for version 3 of libbitcoin-system, I successfully set up libbitcoin-database and ran a node. The process was relatively smooth, and I encountered no major issues during installation or configuration. However, I didn’t come across any minor or major obstacles.

2. Overall, the setup process was well-documented, and any issues encountered were easily resolved. I have documented these in more detail in the provided single-page report.

Stretch Goal Achievement - SQLite Program in C++: Repo Link

3. As per the stretch goal, I have developed a simple SQLite program in C++ to create and query a single table. The program creates a database with a single table, and users can perform basic CRUD (Create, Read, Update, Delete) operations on it. The schema for the table was designed to be straightforward yet functional, allowing for easy querying and manipulation of data.



home > shashank > tempcp > sqlWithCpp.cpp

```cpp
#include <sqlite3.h>
#include <iostream>
#include <string>


const char* DB_FILE = "library.db";

const char* CREATE_TABLE_SQL = R"(
    CREATE TABLE IF NOT EXISTS books (
        id INTEGER PRIMARY KEY,
        title TEXT NOT NULL,
        author TEXT NOT NULL,
        publication_year INTEGER
    );
)";


int executeSQL(sqlite3* db, const std::string& sql) {
    char* errMsg = nullptr;
    int rc = sqlite3_exec(db, sql.c_str(), nullptr, nullptr, &errMsg);
    if (rc != SQLITE_OK) {
        std::cerr << "SQL error: " << errMsg << std::endl;
        sqlite3_free(errMsg);
    }
    return rc;
}

int createTable(sqlite3* db) {
    return executeSQL(db, CREATE_TABLE_SQL);
}


int main() {
    sqlite3* db;
    int rc = sqlite3_open(DB_FILE, &db);
```

Live Share

Why Libbitcoin?

Choosing Libbitcoin for the Summer of Bitcoin 2024 aligns with my passion for Bitcoin, blockchain, and programming. Some factors that contribute to my decision are as follows:

1. **Interest in Machine Learning:** During my programming journey for the 2 years in my college I have developed an immense interest in Machine Learning and have undertaken various projects, including **Visual Question Answering systems** and **GPT**, **Transformer based models**, honing skills in **Natural Language Processing**, **Feature Engineering**, **outlier detection**, and **Deep Learning**. These experiences have equipped me with the technical proficiency required for the project, allowing me to contribute meaningfully to the organization's goals. The anomaly detection project presents an exciting challenge that resonates with my passion for data analysis and problem-solving.
2. **Alignment with My Skills:** My expertise in C++ aligns perfectly with libbitcoin's codebase. This strong foundation will allow me to delve deeper into the intricacies of Bitcoin development and contribute meaningfully to the project.
3. **Interactive and Supportive Environment:** Interacting with the members has provided valuable insights into the organization's culture and values. I have found the community to be welcoming, supportive, and eager to foster growth among its members. The opportunity to engage in meaningful discussions, seek guidance from experienced mentors, and collaborate with like-minded individuals greatly appeals to me.
4. **Open Source Collaboration:** Libbitcoin's open-source ethos fosters collaboration and innovation, providing an ideal environment for learning and growth. By participating in the Summer of Bitcoin program with Libbitcoin, I aim to not only contribute meaningfully to the project but also to immerse myself in a community dedicated to advancing the decentralized future of finance.

Why am I an ideal contributor?

1. **Technical Skills and Bitcoin Knowledge:** I possess a strong foundation in programming languages like C++ and SQLite, Python, cryptography, and distributed systems which are essential for Bitcoin development. My understanding of core Bitcoin concepts like proof-of-work, cryptography, and blockchain technology is solid. I also have a **score of 112** on the assignment provided by the Summer Of Bitcoins.
2. **Dedicated Time and Commitment:** I am fully committed to dedicating my time and effort to this project for the next 12 weeks (about 3 months). I have **no other major commitments**, ensuring my full focus on meeting and exceeding project expectations within the given time limit. Participating in the Summer of Bitcoin would be an incredible opportunity to contribute to the future of Bitcoin. I'm incredibly enthusiastic about collaborating with the Libbitcoin developers and making a lasting impact on the project.
3. **Passion for Bitcoin and open source:** I'm genuinely passionate about Bitcoin and its potential to revolutionize finance and technology. My enthusiasm drives my desire to contribute meaningfully to the Bitcoin ecosystem. Also, I am deeply involved in open-source communities, showcasing my teamwork and tech skills. I have made 7 pull requests of which 3 have been merged and have raised 6 issues. Beyond code, I actively discuss, offer feedback, and solve problems collaboratively.

Project Timeline

Project Period		
Week 1-Week 4	**API Understanding:** Thoroughly comprehending the libbitcoin-database API and its usage within other libbitcoin components. This includes analyzing function calls, data structures, and interaction patterns.	🟥
Week 5-Week 8	**SQLite Database Development:** **Schema Implementation:** Creating an SQLite schema that accurately reflects the existing libbitcoin-database schema, ensuring data compatibility. **API Query Implementation:** Developing SQLite queries that fulfill the functionalities exposed by the libbitcoin-database API. These queries should efficiently retrieve and manipulate data within the SQLite database.	🟥
Week 9-Week 12	**Testing and Validation:** **Test Suite Creation:** Utilizing the boost test suite, familiar to the libbitcoin project, to create comprehensive unit tests for all implemented functions. These tests will ensure the correctness and reliability of `libbitcoin-database-sqlite`.	🟥

Final Week	**API Improvement Updates:** Analyzing the development process and identifying potential modifications to the libbitcoin-database API that could streamline integration with `libbitcoin-database-sqlite`. This may involve reducing complexity or introducing features specifically suited for an SQLite backend.	🟥
Thorough testing and deployment		

I will devote 4 hours per day for 12 weeks (about 3 months) during my summer holidays and around 2-3 hours each day during the final week. This sums up to **nearly 300 hours** for the entire project. The work distribution has been written in a tabular form for better understanding.